CPSC 340:
Machine Learning and Data Mining

More CNNs

Fall 2016

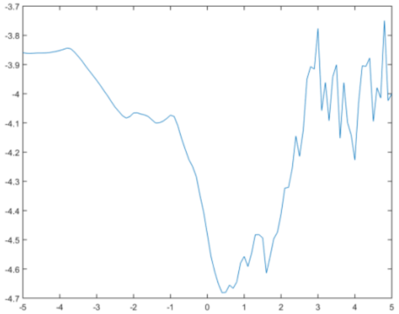
Admin

- Assignment 5 now due tomorrow at 2pm.
  - Extra offices hours today at 4:30-5:30 in ICICS 104.
  - 1 late day to hand in Monday, 2
- Assignment 6:
  - Due next Friday (usual late day policy, assuming phantom “classes”).
  - Neural network code updated to be easier to understand/modify.
- Final:
  - December 12 (8:30am – HEBB 100)
  - Covers Assignments 1-6.
  - List of topics will be posted this weekend.
  - Final from last year will be posted next weekend.
  - Closed-book, cheat sheet: 4-pages each double-sided.

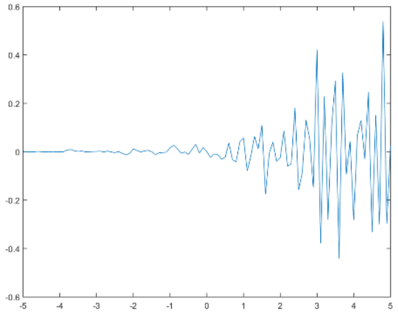
Last Time: Convolutions

- 1D convolution:
  - Takes signal ‘x’ and filter ‘w’ to produces vector ‘z’:

$$x \quad * \quad w \quad = \quad z$$



$$[-1 \quad 2 \quad -1]$$



  - Can be written as a matrix multiplication:

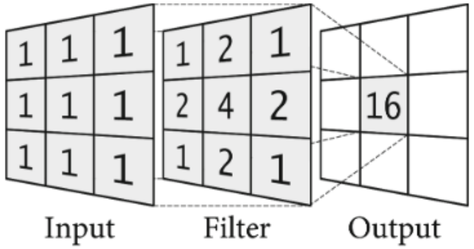
$$Wx = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ & & & \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix} = z$$

Last Time: Convolutions

- 2D convolution:
  - Signal ‘x’, filter ‘w’, and output ‘z’ are now all images/matrices:

$$x \quad * \quad w \quad = \quad z$$

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

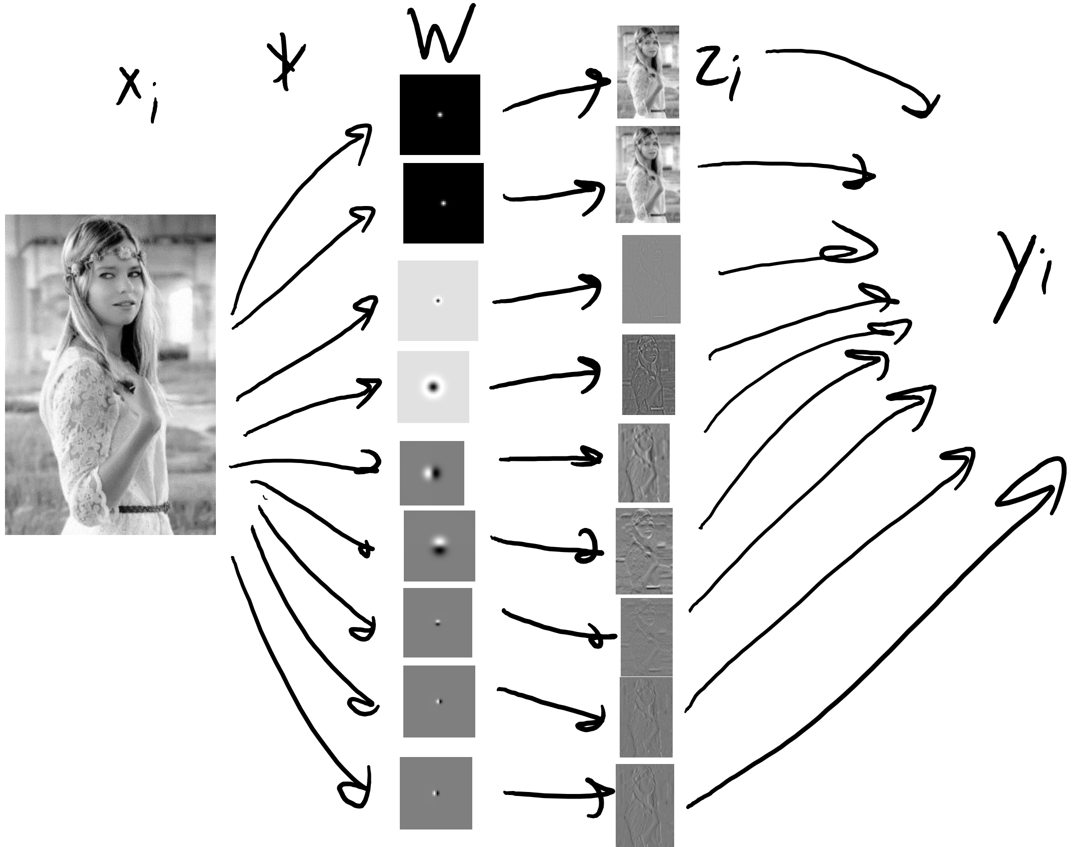


  - Vectorized ‘z’ can be written as a matrix multiplication with vectorized ‘x’:

$$W = \begin{bmatrix}
-2 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \\
0 & -2 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 2 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \\
 & & \vdots & & & \cdots & 2 & -1\ 0 & 0 & 0 & 0 & \cdots & & & -1 & 0\ 1 & 0 & 0 & 0 & \cdots & & & 0 & 1\ 2 & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & & & & & & & & & & & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 2 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 2 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\
 & & \vdots & & & & & & & & & \vdots & & & & & & & & \vdots & & & & & & &
\end{bmatrix}$$

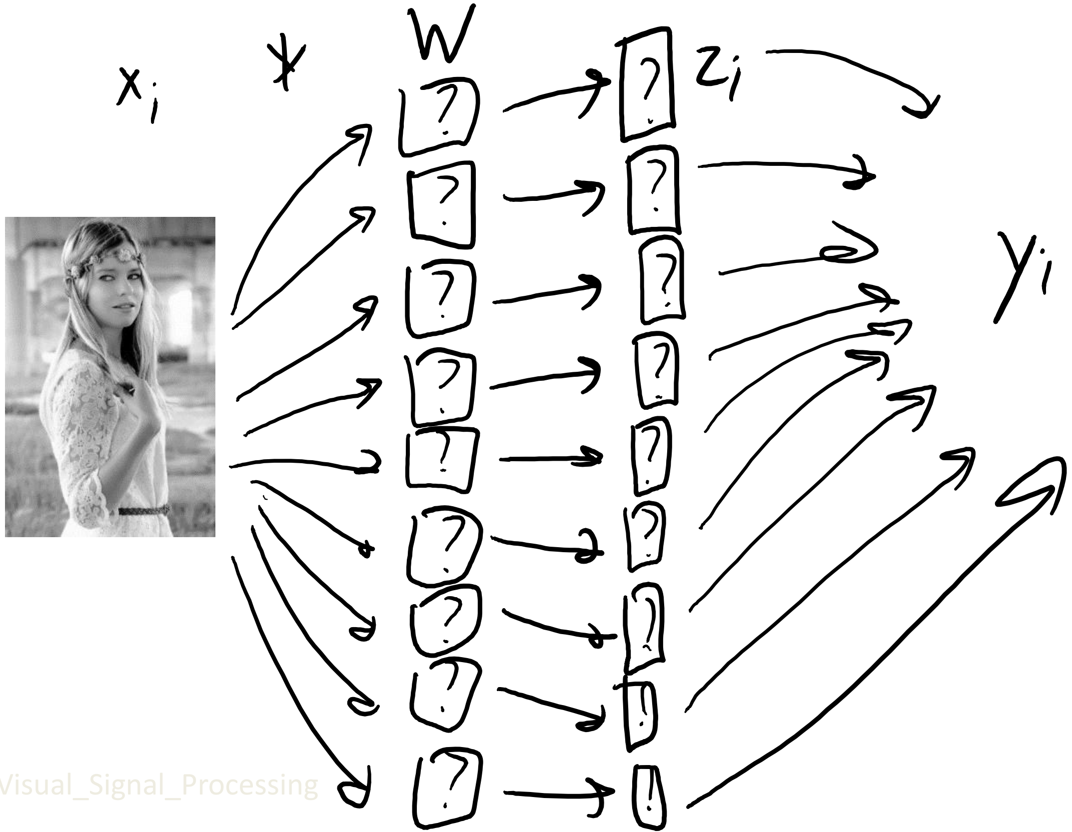
Last Time: Convolutional Neural Networks

- Classic approach uses fixed convolutions as features:
  - Usually have different types/variances/orientations.
  - Can do subsampling or taking maxes across locations/orientations/scales.

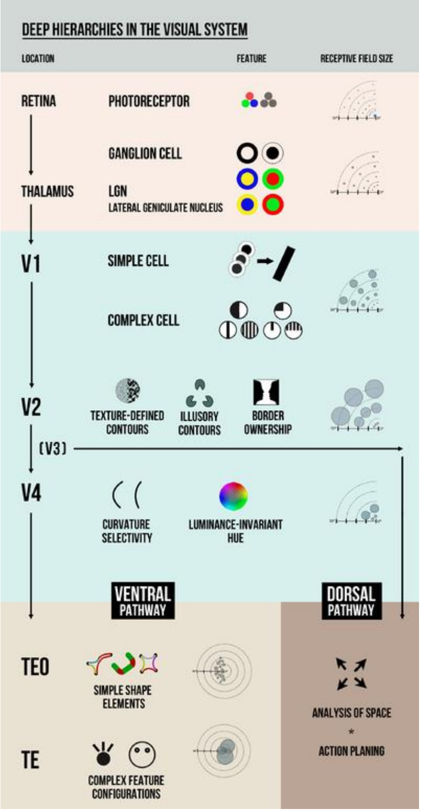


Last Time: Convolutional Neural Networks

- Convolutional neural networks learn the features:
  - Learning ‘W’ and ‘w’ automatically chooses types/variances/orientations.
  - Can do multiple layers of convolution to get deep hierarchical features.

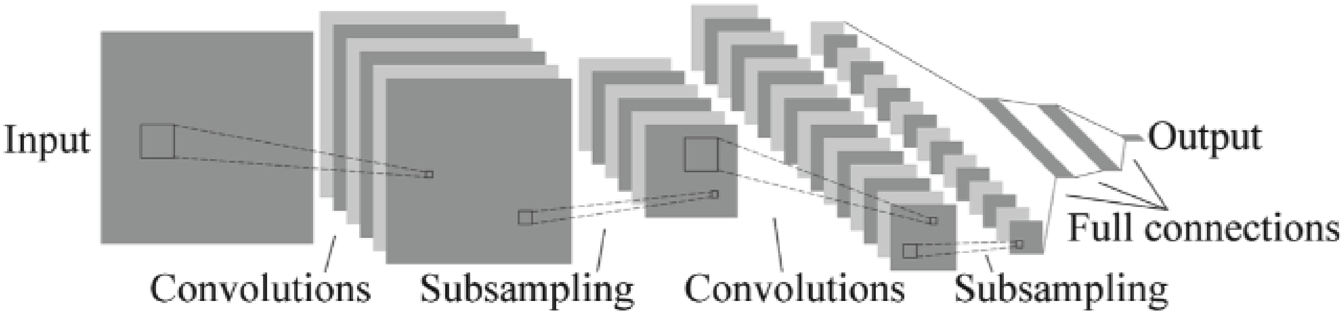


https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing

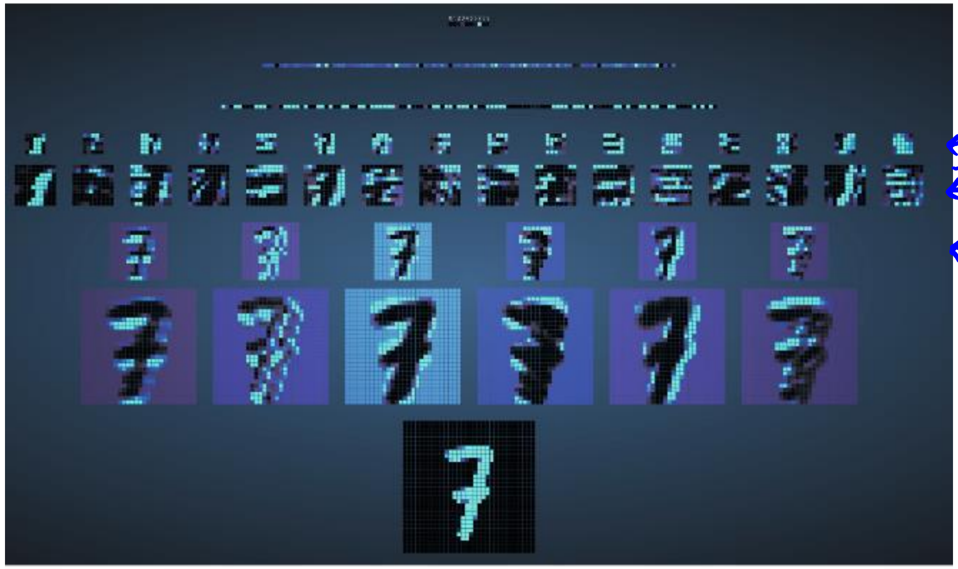


Last Time: Convolutional Neural Networks

- Classic convolutional neural network (LeNet):



- Visualizing the "activations" of the layers:
  - http://scs.ryerson.ca/~aharley/vis/conv



softmax
2 "fully-connected"
max pooling
3D convolutions
max pooling
2D convolutions

http://scs.ryerson.ca/~aharley/vis/harley_vis_isvc15.pdf

AlexNet Convolutional Neural Network

- ImageNet 2012 won by AlexNet:
  - 15.4% error vs. 26.2% for closest competitor.
  - 5 convolutional layers.
  - 3 fully-connected layers.
  - SG with momentum.
  - ReLU non-linear functions.
  - Data translation/reflection/ cropping.
  - L2-regularization + Dropout.
  - 5-6 days on two GPUs.

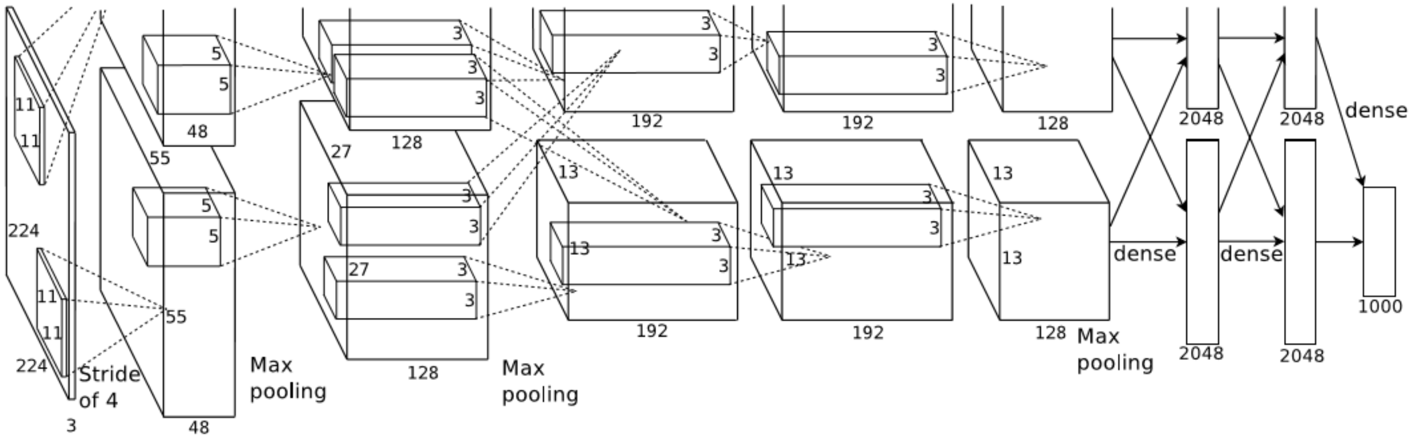


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

AlexNet Convolutional Neural Network

- ImageNet 2012 won by AlexNet:
  - 15.4% error vs. 26.2% for closest competitor.

Gaussian times sine/cosine: "Gabor" filters

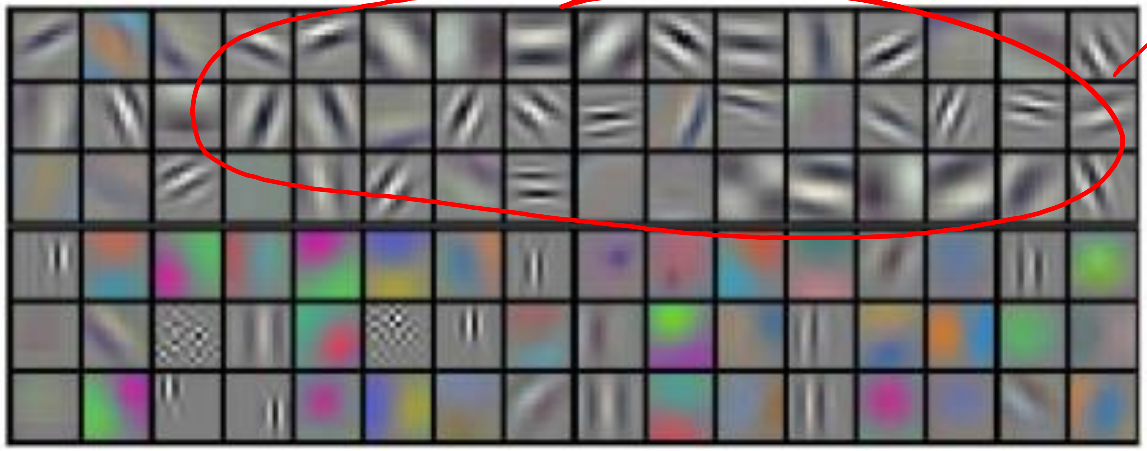
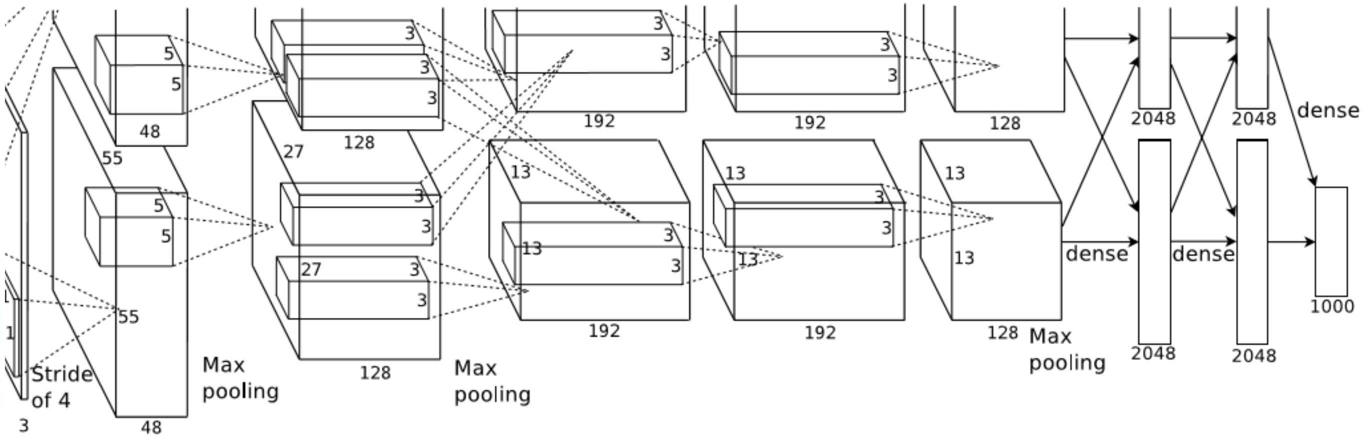


Figure 3: 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images. The



ure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities veen the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts ie bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–6–4096–1000.

http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

ZFNet Convolutional Neural Network

- ImageNet 2013 won by variation of AlexNet called ZF Net:
  - 11.2% error (now using 11x11 instead of 7x7).
  - Introduced deconvolutional networks to visualize what CNNs learn.

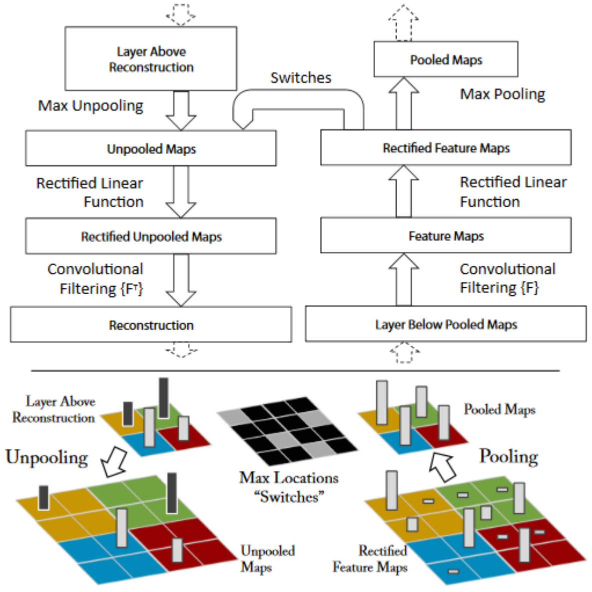
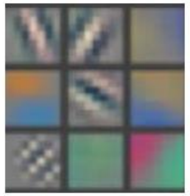
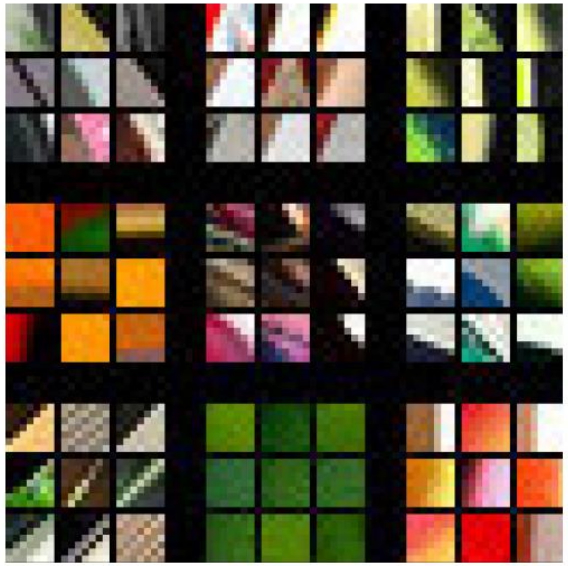


Figure 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local max in each pooling region (colored zones) during pooling in the convnet.

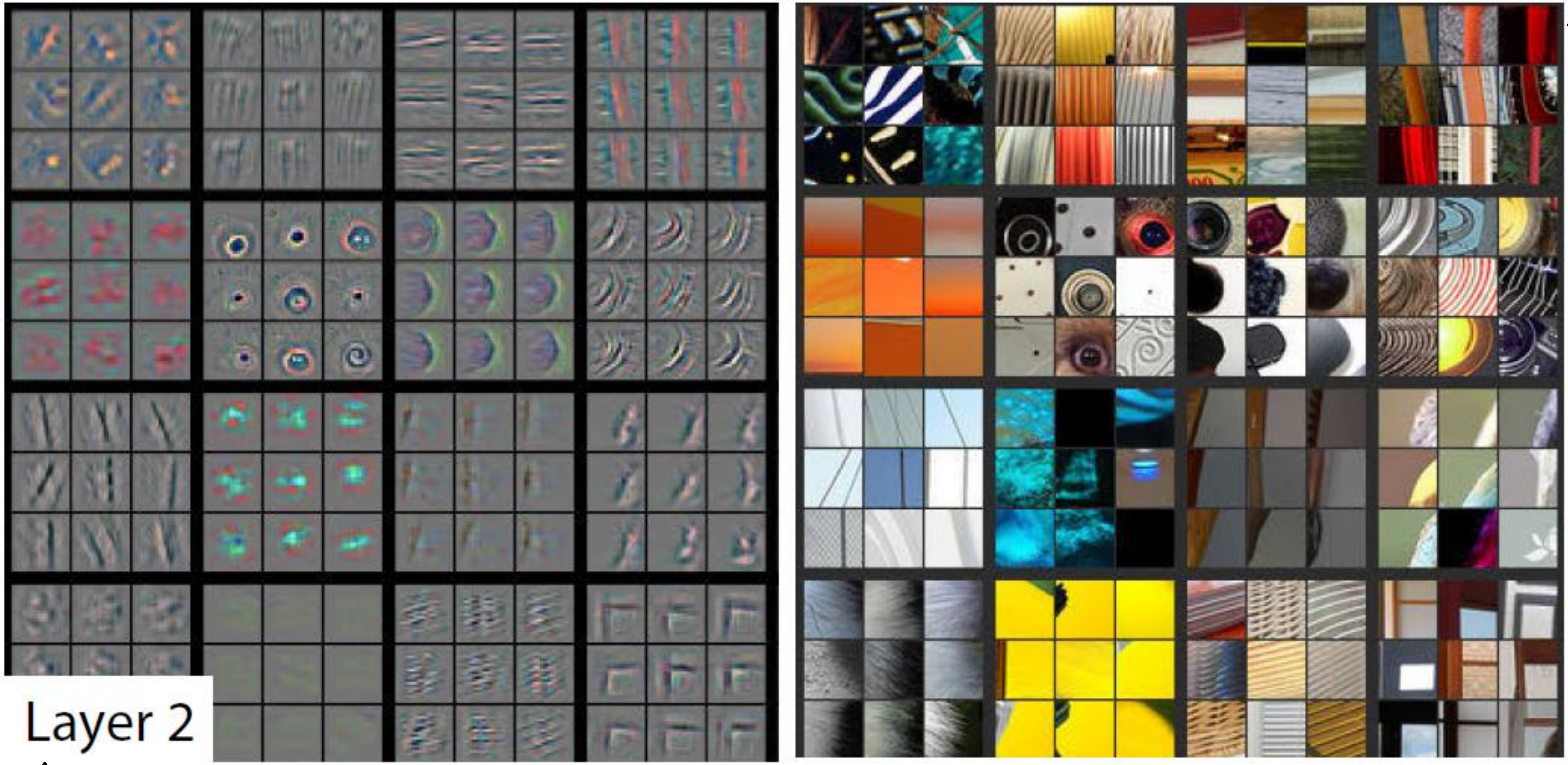


Layer 1



https://arxiv.org/pdf/1311.2901v3.pdf

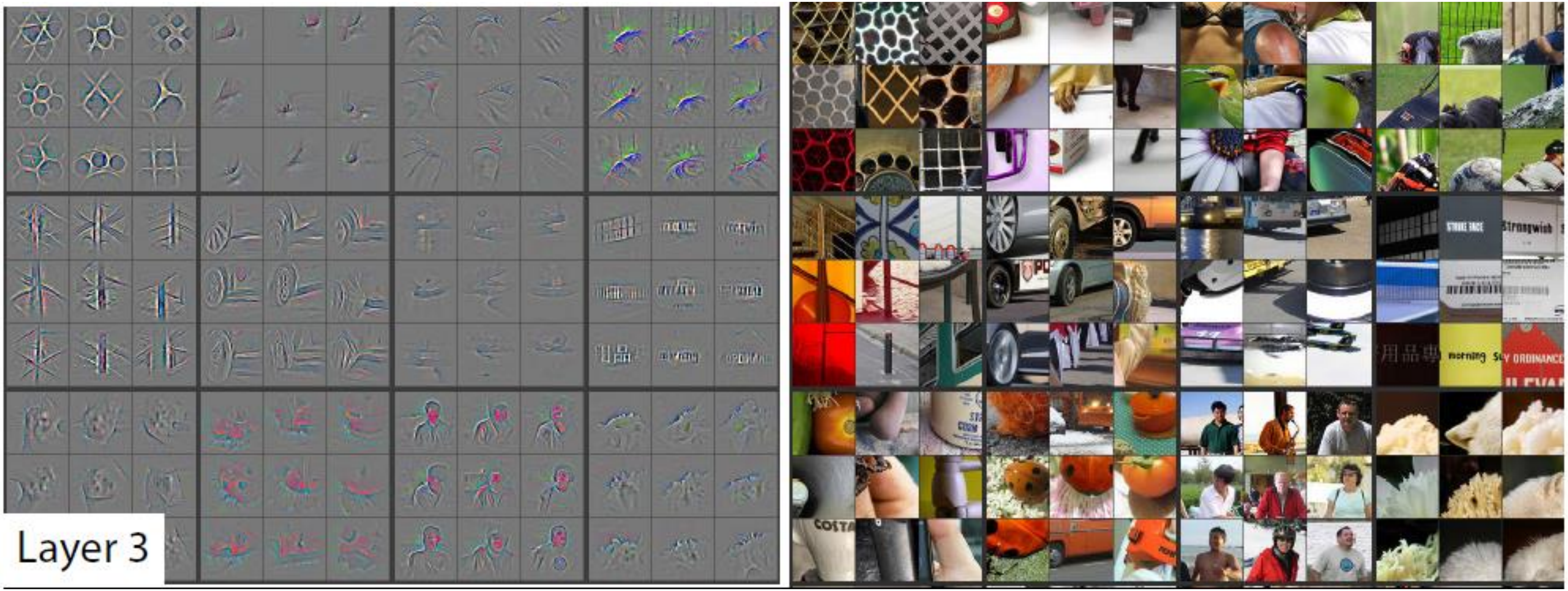
ZFNet Convolutional Neural Network



Patch from data giving largest response

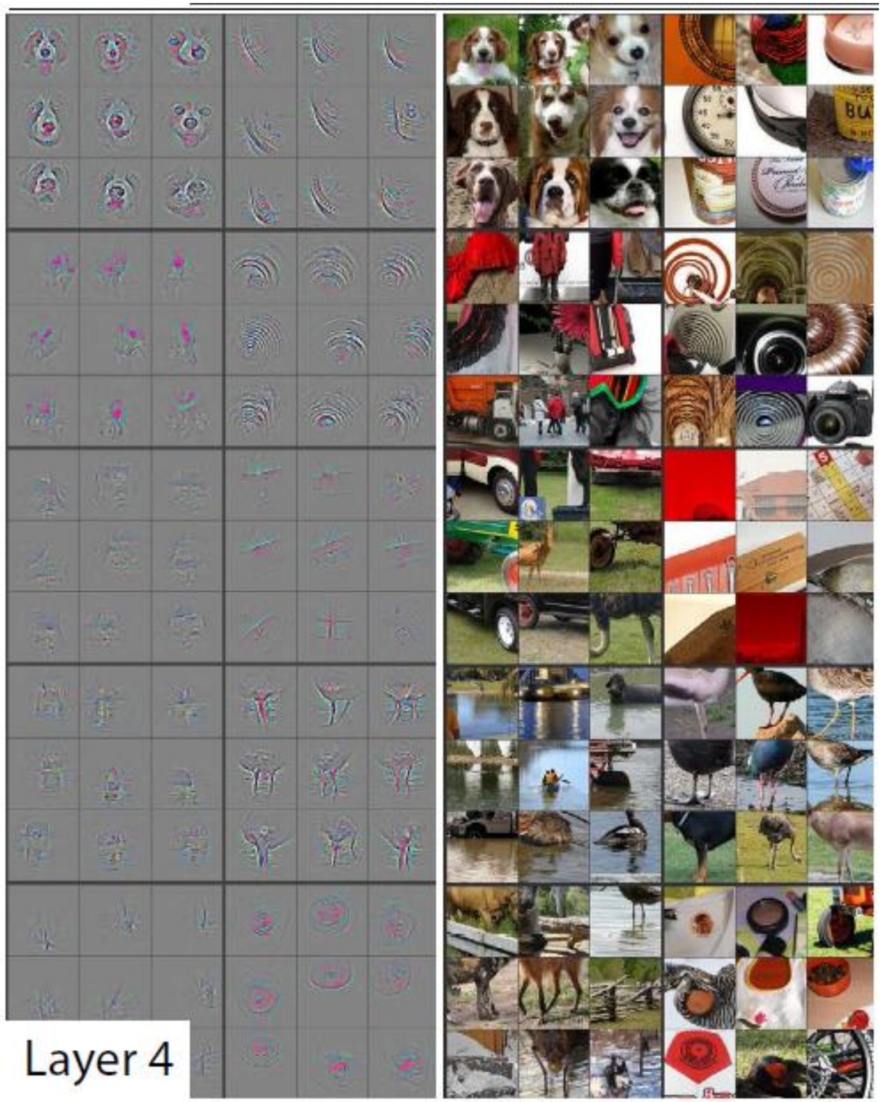
Deconvolution network giving patch that leads to largest response

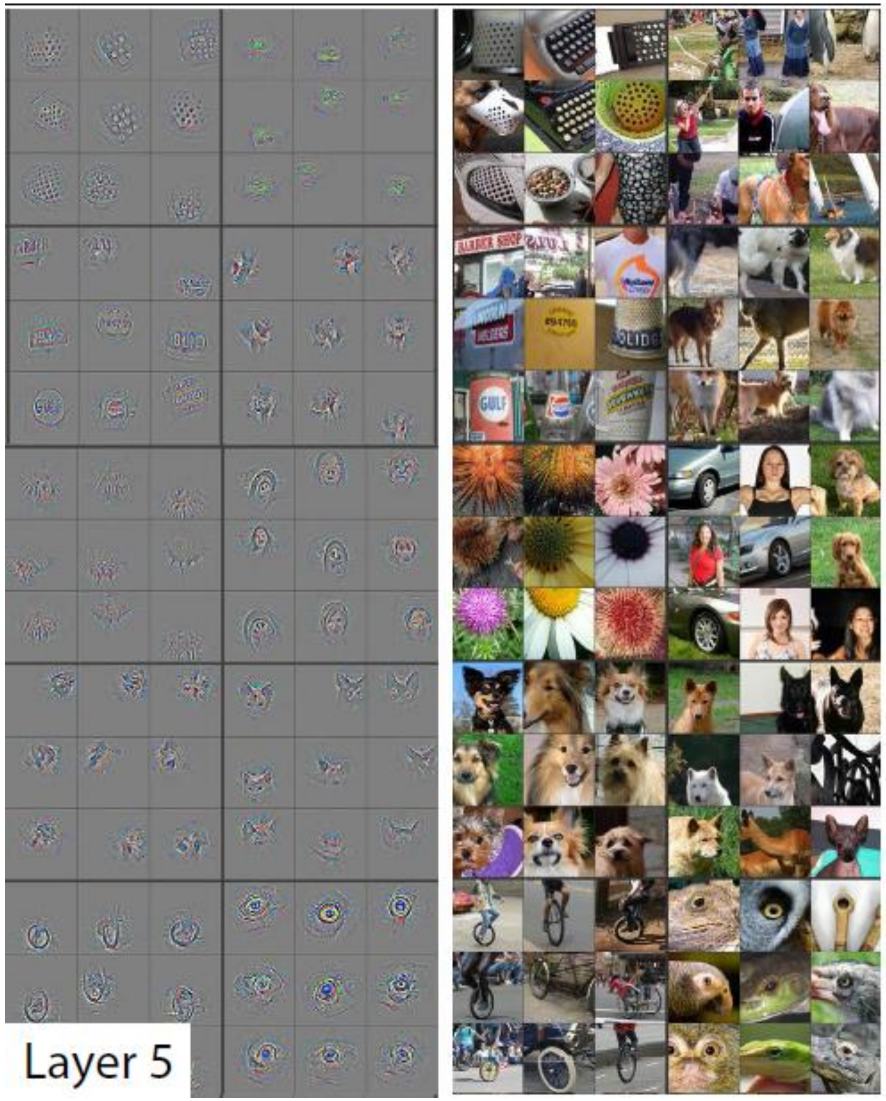
ZFNet Convolutional Neural Network



https://arxiv.org/pdf/1311.2901v3.pdf

ZFNet Convolutional Neural Network

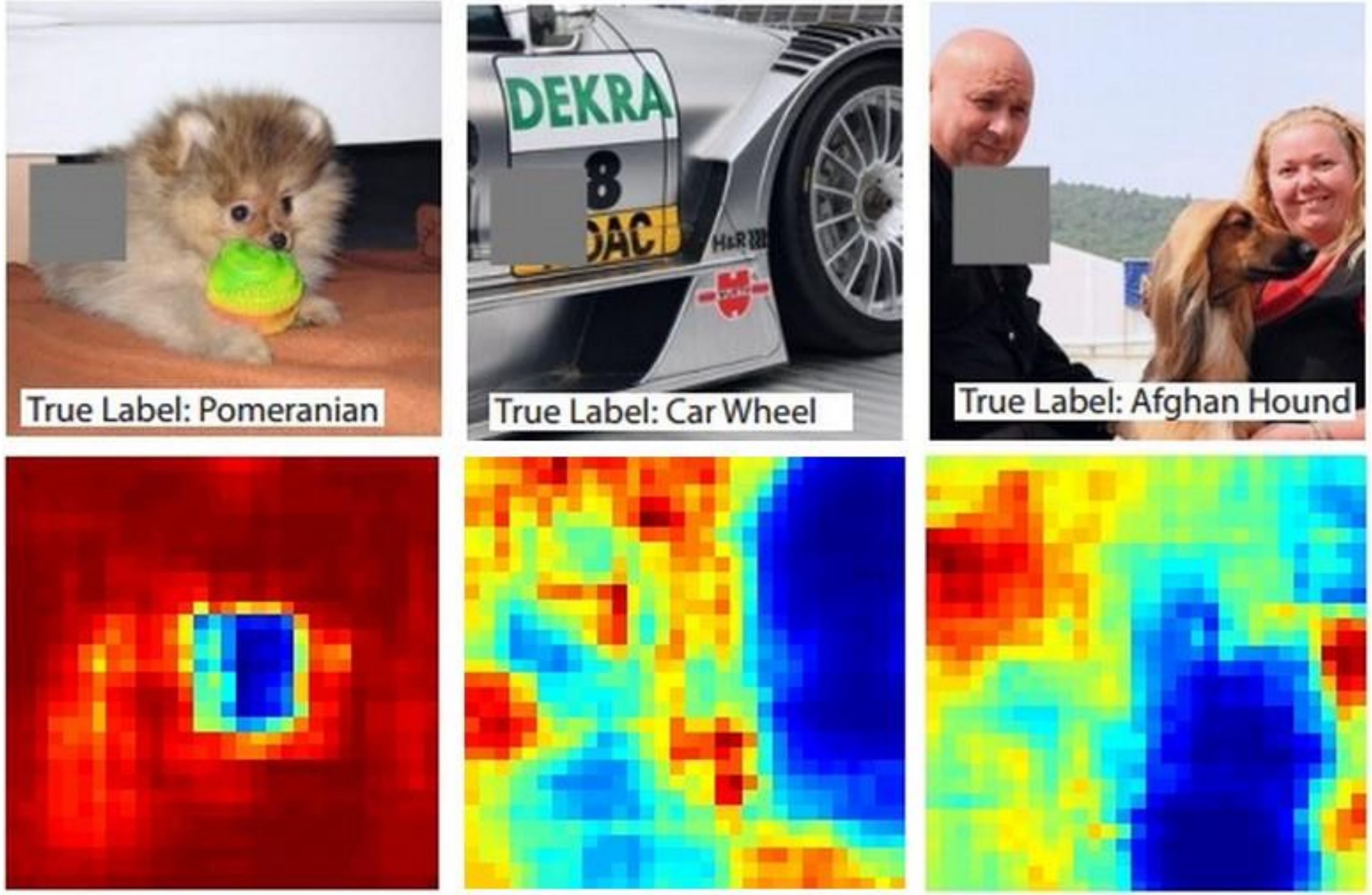




https://arxiv.org/pdf/1311.2901v3.pdf

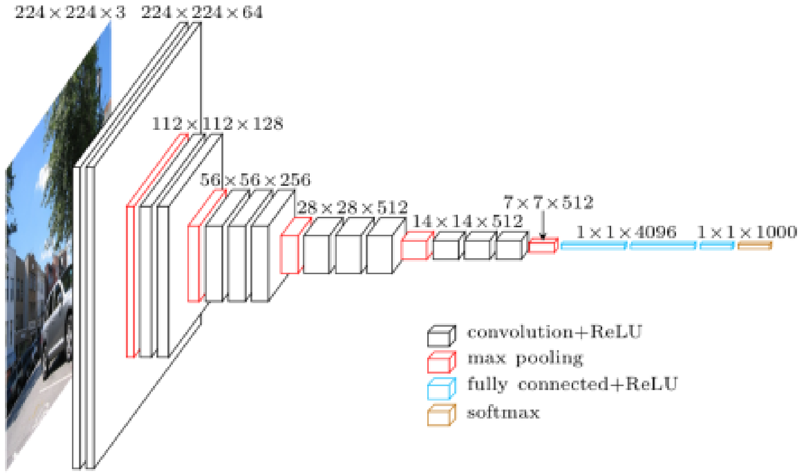
ZFNet Convolutional Neural Network

- Looked at how prediction changes if we hide part of the image:



VGG Convolutional Neural Network

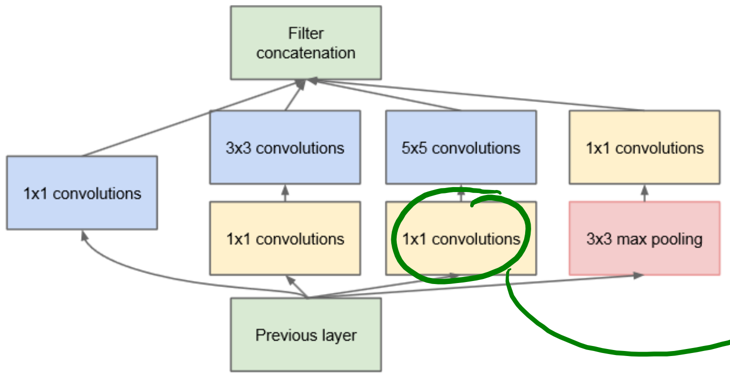
- Image 2014 "Localization" Task won by a 19-layer VGG network:
  - 7.3% error for classification (2^{nd} place).
  - Uses 3x3 convolution layers with stride of 1:
    - 3x3 followed by 3x3 simulates a 5x5, and another 3x3 simulates a 7x7, and so on.
    - Speeds things up and reduces number of parameters.
    - Increases number of non-linear ReLU operations.
  - "Deep and simple": variants of VGG are the most popular CNNs.



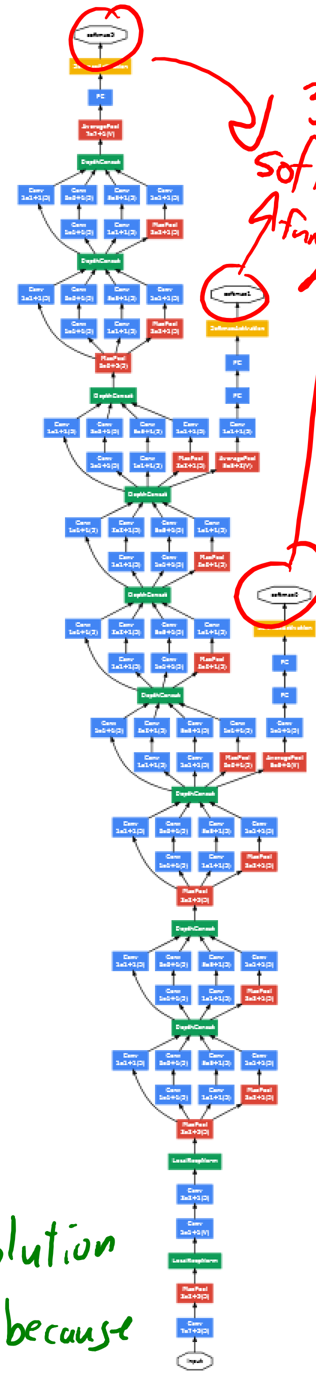
https://www.cs.toronto.edu/~frossard/post/vgg16/

GoogLeNet

- Image 2014 classification task won by GoogLeNet:
  - 6.7% errors.
  - 22 layers
    - No fully connected layers.
    - During training, try to predict label at multiple locations.
      - During testing, just take the deepest predictions.
    - “Inception” modules: combine convolutions of different sizes.



(b) Inception module with dimensionality reduction



3 softmax function

“1x1” convolution makes sense because these are first 2 dimensions of 3D conv.

https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf
http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

ResNet

- Image 2015 won by Resnet (all 5 tasks):
  - 3.6% error (below estimate 5% human error).
  - 152 layers (2-3 weeks on 8 GPUs to train).
  - "Residual learning" allows better performance with deep networks:
    - Include input to layer in addition to non-linear transform.

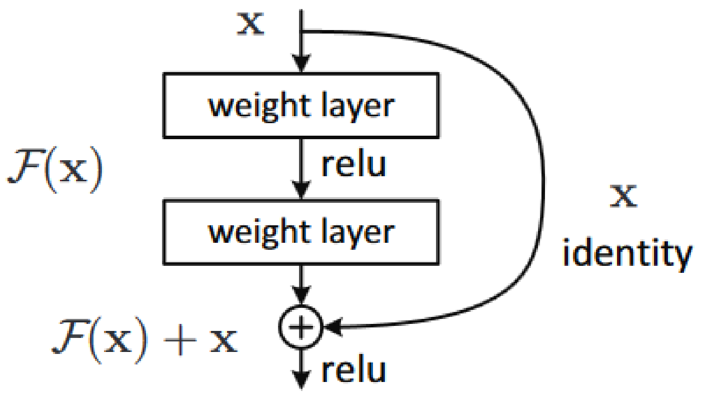
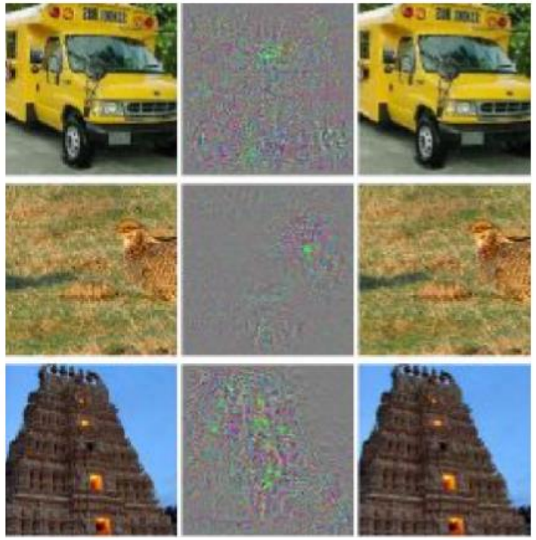


Figure 2. Residual learning: a building block.

    - Network just focuses on "residual": what is not captured in original signal.

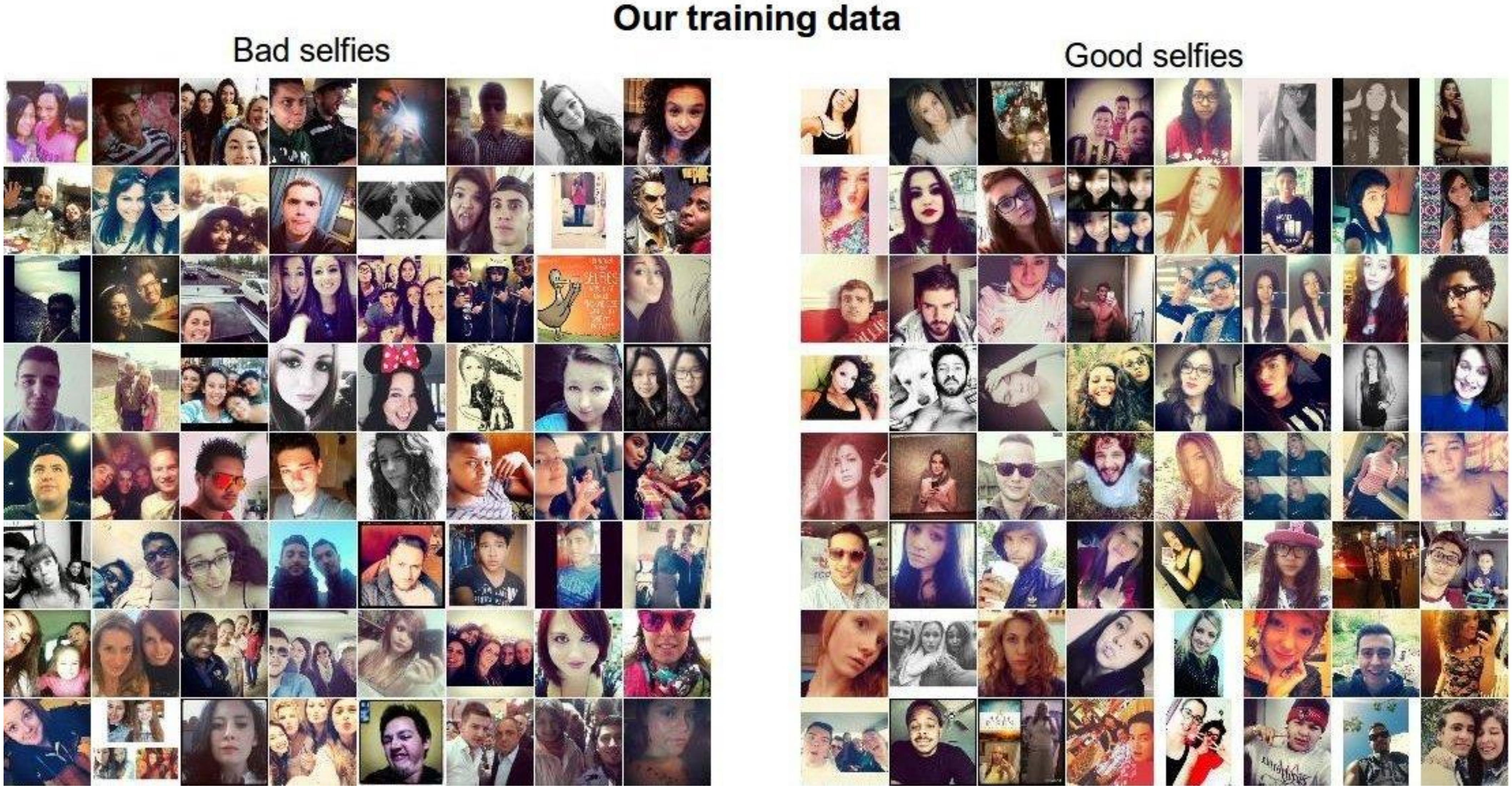
Mission Accomplished?

- For speech recognition and object detection:
  - No other methods have ever given the current level of performance.
  - But, we also don't know how to scale up other universal approximators.
  - There is likely some overfitting to these particular tasks.
- Despite high-level of abstraction, deep CNNs are easily fooled:
  - But progress on fixing 'blind spots'.



http://cs.nyu.edu/~zaremba/docs/understanding.pdf

CNNs for Rating Selfies

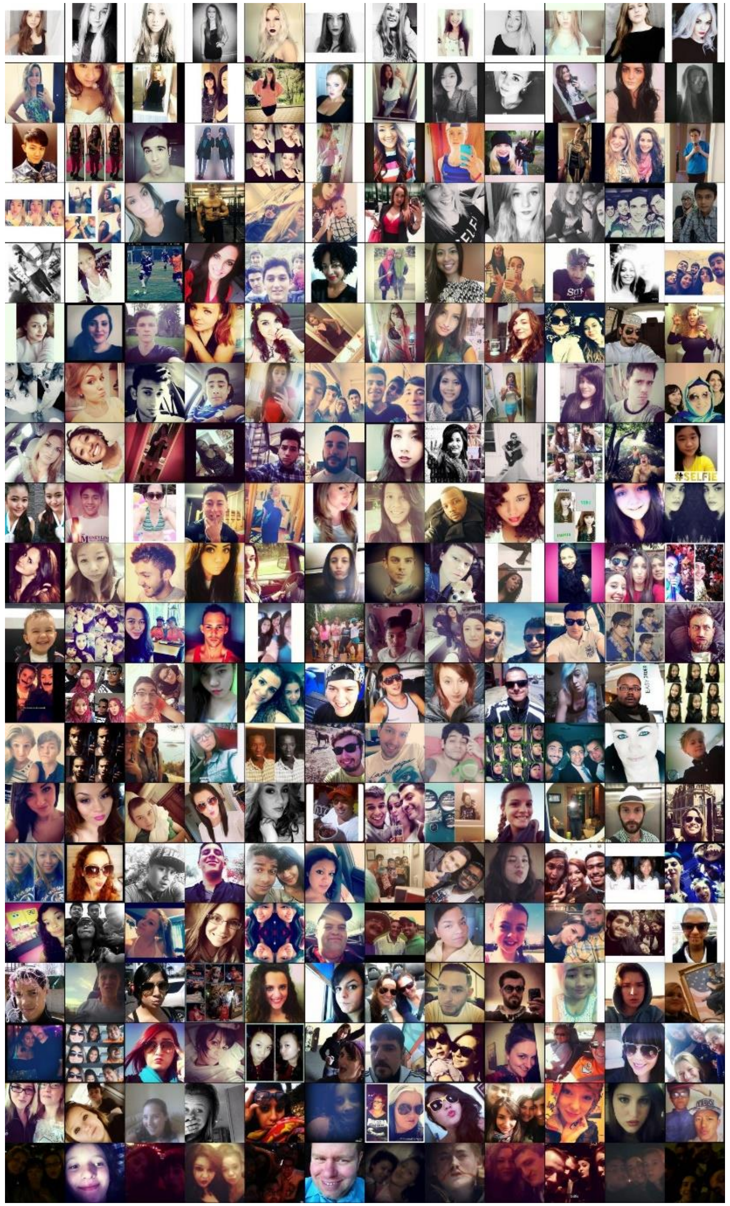


https://karpathy.github.io/2015/10/25/selfie/

CNNs for Rating Selfies

Do:

- Be female
- Have face be 1/3 of image
- Cut off forehead
- Show long hair
- Oversaturate face
- Use filter
- Add border



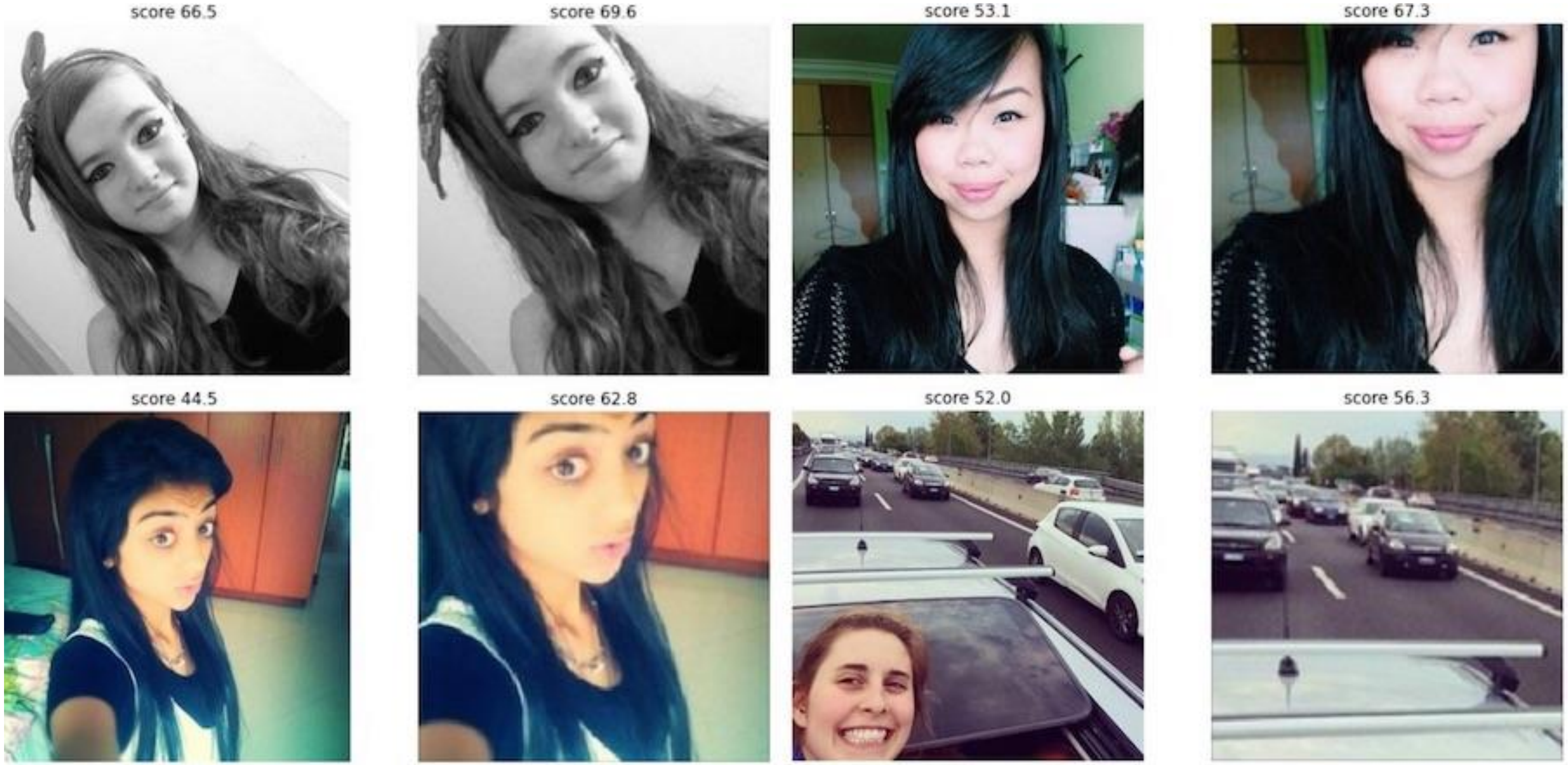
Don't:

- Use low lighting
- Make head too big
- Take group shots

https://karpathy.github.io/2015/10/25/selfie/

CNNs for Rating Selfies

Finding best image crop:



https://karpathy.github.io/2015/10/25/selfie/

Beyond Classification

- “Fully convolutional” neural networks allow “dense” prediction:

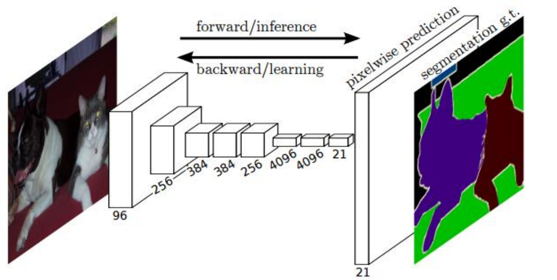


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

  – Best methods combine these with graphical models or LSTMs (CPSC 540).

- Image segmentation:

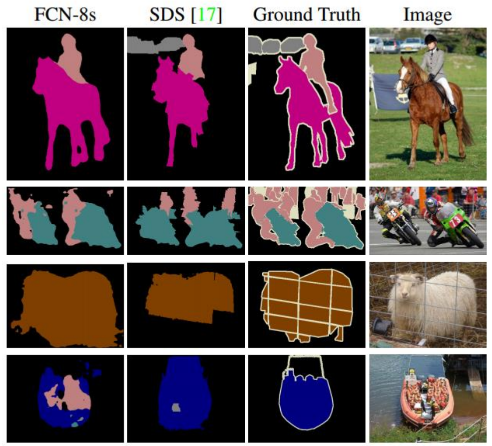


Figure 6. Fully convolutional segmentation nets produce state-of-the-art performance on PASCAL. The left column shows the output of our highest performing net, FCN-8s. The second shows the segmentations produced by the previous state-of-the-art system by Hariharan *et al.* [17]. Notice the fine structures recovered (first

https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

Beyond Classification

- "Fully convolutional" neural networks allow "dense" prediction:

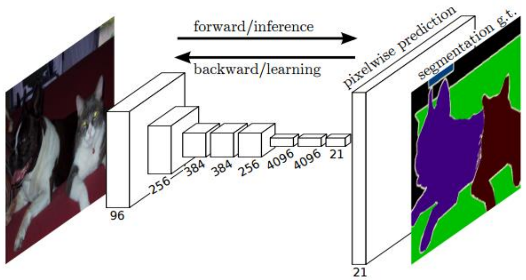
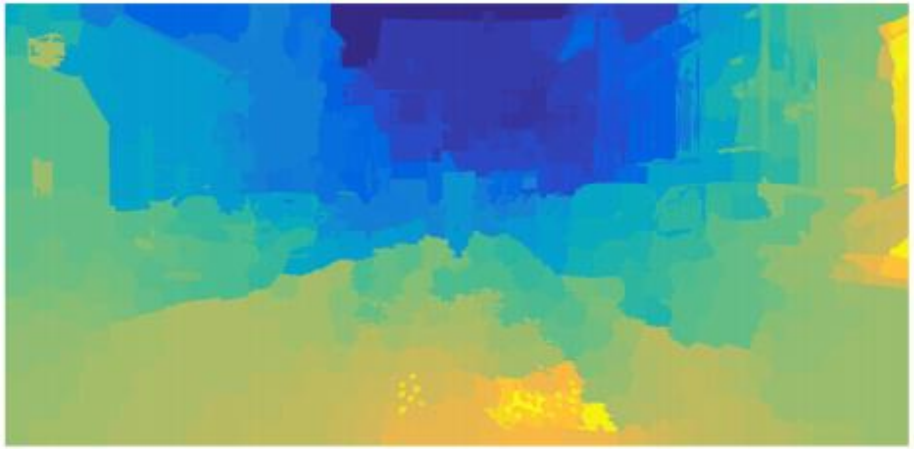


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

  – Best methods combine these with graphical models or LSTMs (CPSC 540).

- Depth Estimation:



Beyond Classification

- Image colorization:

Colorado National Park, 1941 | Textile Mill, June 1937 | Berry Field, June 1909 | Hamilton, 1936

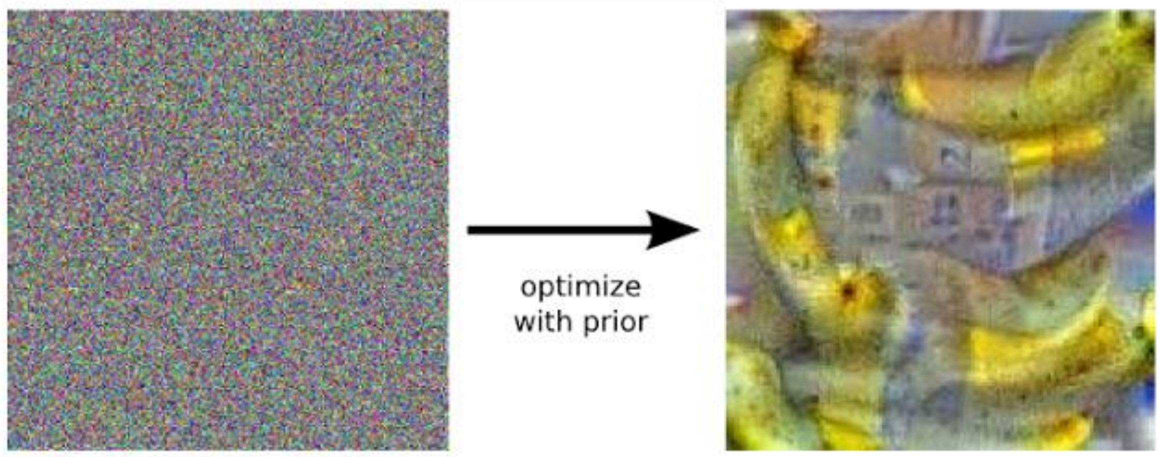
  - [Image Gallery](), [Video]()

http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/en

Inceptionism

- A crazy idea:
  - Instead of weights, use backpropagation to take gradient with respect to x_i.
- Inceptionism with trained network:
  - Fix the label y_i (e.g., "banana").
  - Start with random noise image x_i.
  - Use gradient descent on image x_i.
  - Add a spatial regularizer on xi:
    - Encourages neighbouring x_{ij} to be similar.

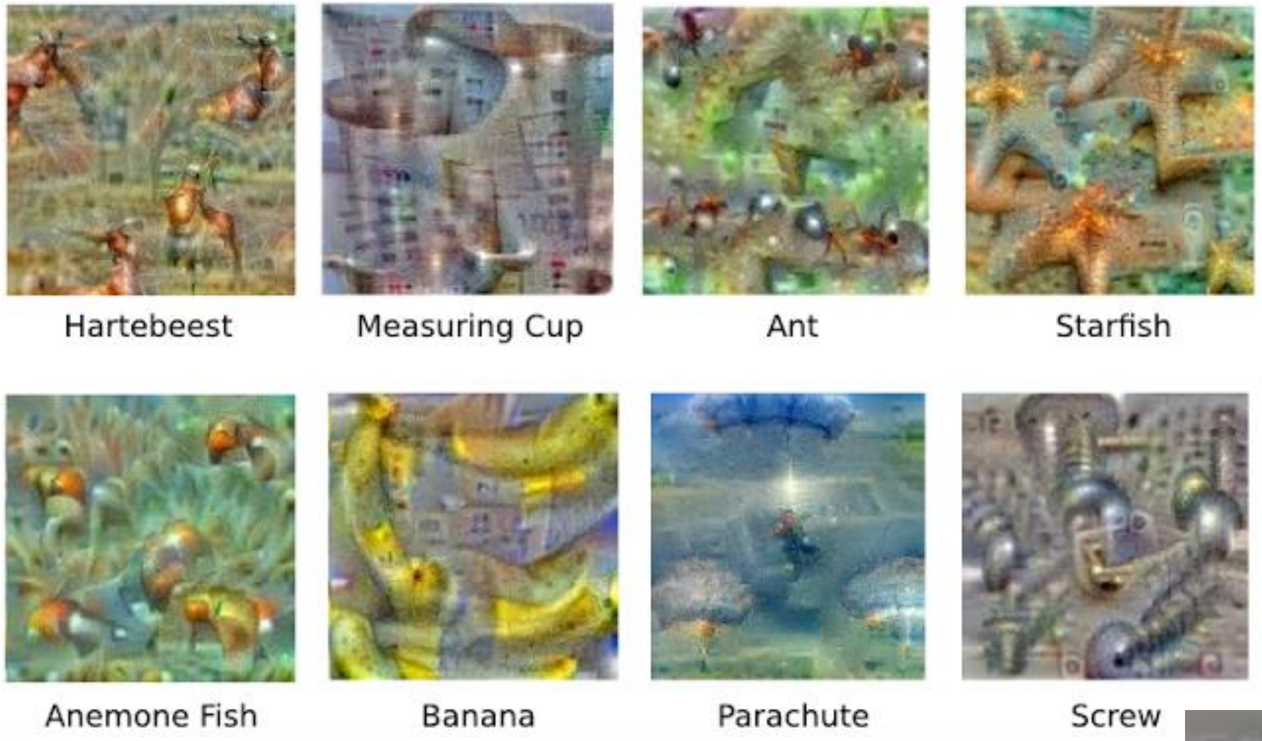
"Show what you think a banana looks like."



http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

Inceptionism

- Inceptionism for different class labels:

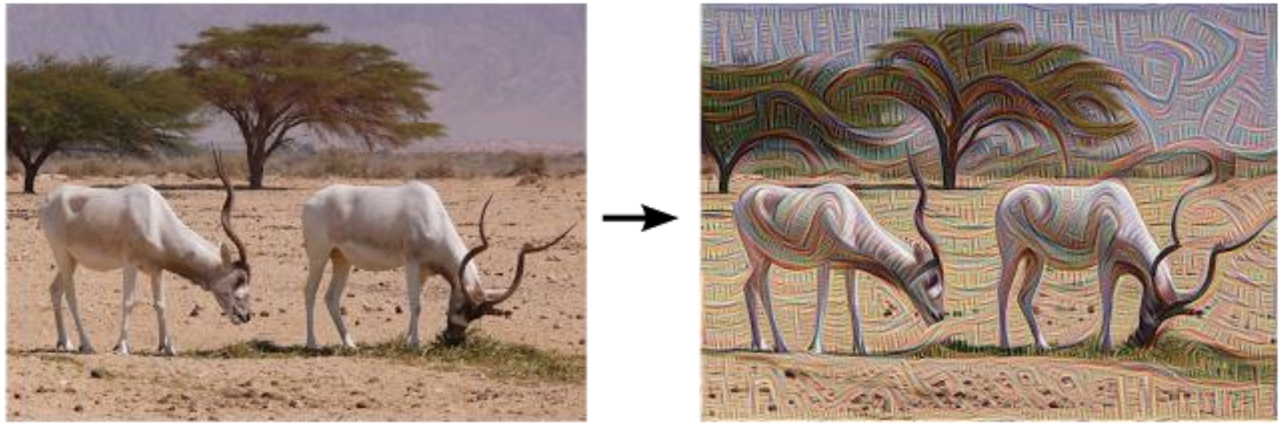


Dumbbell

http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

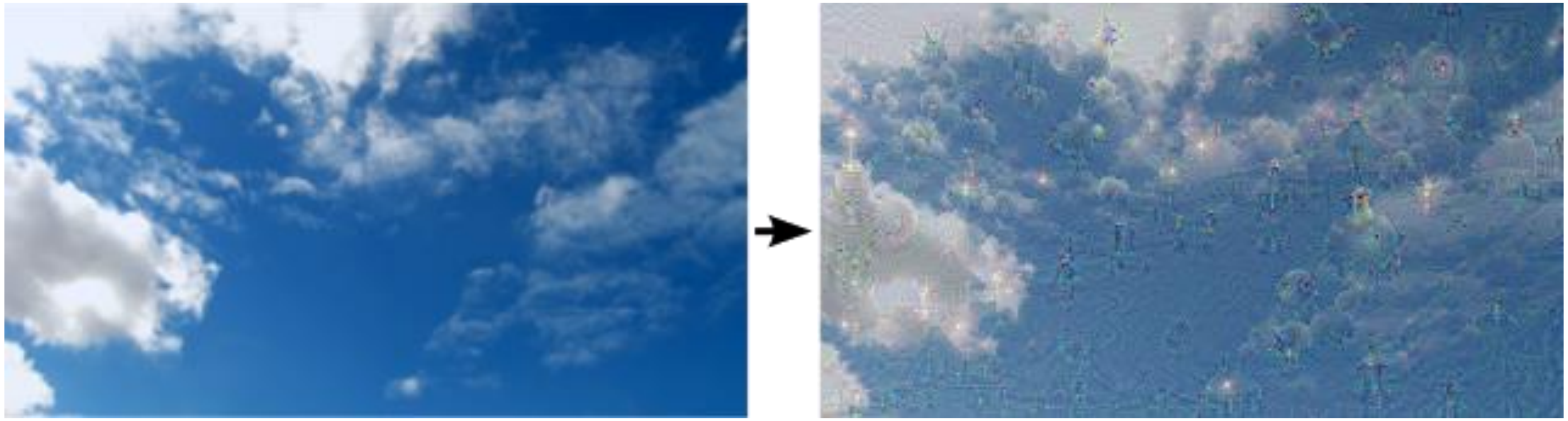
Inceptionism

- Inceptionism where we try to match $z_i^{(m)}$ values instead of y_i.
  - Shallow 'm':



Inceptionism

- Inceptionism where we try to match $z_i^{(m)}$ values instead of y_i.
  - Deepest 'm':



"Admiral Dog!"

"The Pig-Snail"

"The Camel-Bird"

"The Dog-Fish"

http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

Inceptionism

- Inceptionism where we try to match $z_i^{(m)}$ values instead of y_i.
  - "Deep dream" starts from random noise:

  - Inceptionism gallery
  - Deep Dream video

http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

Artistic Style Transfer

- Artistic style transfer:
  - Given a content image ‘C’ and a style image ‘S’.
  - Make a image that has content of ‘C’ and style of ‘S’.

Artistic Style Transfer

- Artistic style transfer:
  - Given a content image ‘C’ and a style image ‘S’.
  - Make a image that has content of ‘C’ and style of ‘S’.

Content:

Style:

https://commons.wikimedia.org/wiki/File:Tuebingen_Neckarfront.jpg
https://en.wikipedia.org/wiki/The_Starry_Night

Artistic Style Transfer

- Artistic style transfer:
  - Given a content image 'C' and a style image 'S'.
  - Make a image that has content of 'C' and style of 'S'.

- CNN-based approach applies gradient descent with 2 terms:
  - Loss function: match deep latent representation of content image 'C':
    - Difference between $z_i^{(m)}$ for deepest 'm' between x_i and 'C'.
  - Regularizer: match all latent representation covariances of style image 'S'.
    - Difference between covariance of $z_i^{(m)}$ for all 'm' between x_i and 'C'.

Artistic Style Transfer

http://arxiv.org/pdf/1508.06576v2.pdf

Image Gallery

Examples

Figure: **Left:** My friend Grant, **Right:** Grant as a pizza

Artistic Style Transfer

- Recent methods combine CNNs with graphical models (CPSC 540):

Input A | Input B | Content A + Style B | Content B + Style A

https://arxiv.org/pdf/1601.04589.pdf

Artistic Style Transfer

- Recent methods combine CNNs with graphical models (CPSC 540):

https://arxiv.org/pdf/1601.04589.pdf

Artistic Style Transfer for Video

- Combining style transfer with optical flow:
  – https://www.youtube.com/watch?v=Khuj4ASldmU
- Videos from Ricky's paper: