CPSC 340:
Machine Learning and Data Mining

More CNNs

Fall 2017

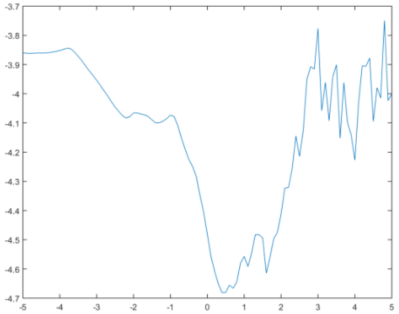
Admin

- Assignment 5:
  - 1 late day for tonight, 2 for Friday.
- Final:
  - Next Tuesday, details and previous exams posted on Piazza.
- Extra office hours:
  - 3:00-?:?? tomorrow in ICICS 146 (with me).
  - Monday we’ll have office hours at 11-12 (1 TA) and 1-2 (2 TAs).
  - Tuesday we’ll have office hours from 12-2 (1 TA).
- Assignemnt x grades:
  - If there are remaining issues (e.g., missing grades) post on Piazza soon.

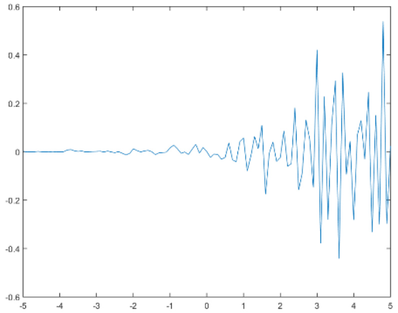
Last Time: Convolutions

- 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:

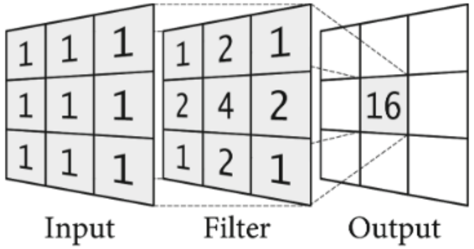
$$Wx = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ & & & \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix} x = z$$

Last Time: Convolutions

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

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

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

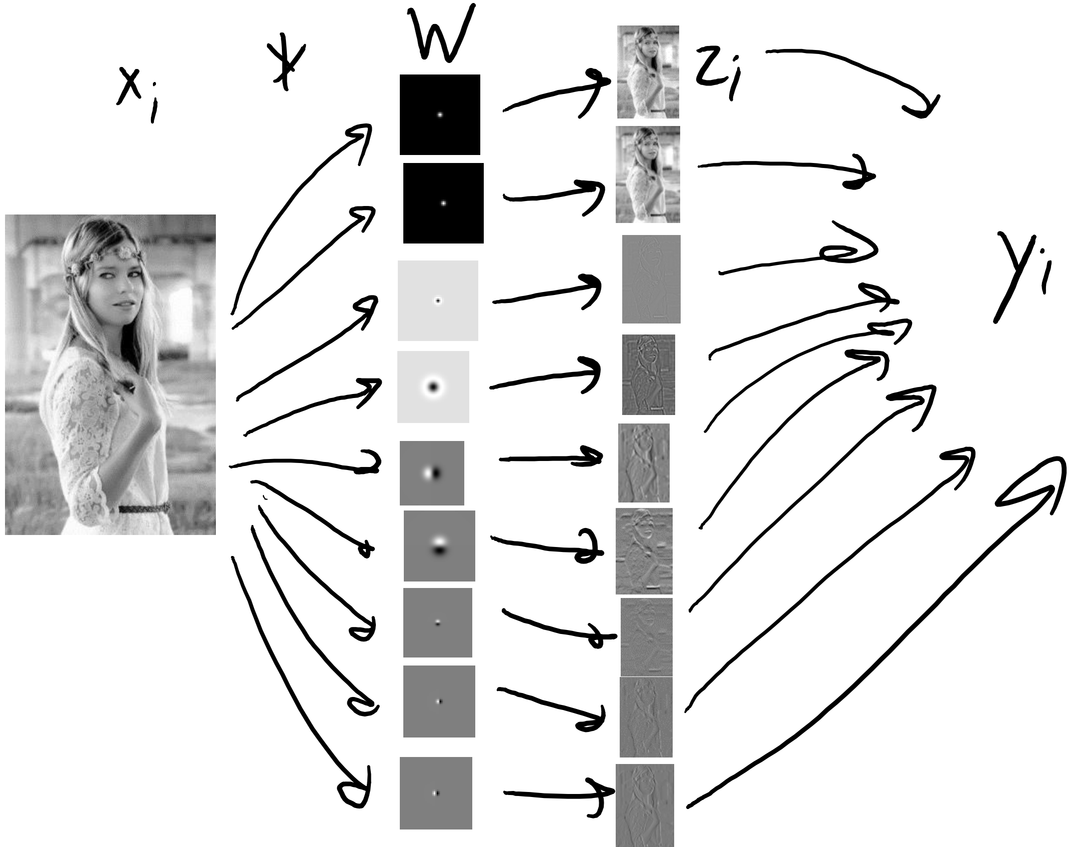


  - Vectorized ‘z’ can be written as a matrix multiplication with vectorized ‘x’:

$$W = \left[\begin{array}{cccccccc|cccccccc|cccccccc|c}
-2 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 2 & 0 & 0 & 0 & \cdots & 0 & 0\,0\cdots \\
0 & -2 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 2 & 0 & 0 & \cdots & 0 & 0\,0\cdots \\
 & & \vdots & & & \cdots & & 2\,{-1}\,0 & 0 & 0 & 0 & & \cdots & & & -1\,0\,1 & 0 & 0 & 0 & & \cdots & & & 0\,1\,2 & 0\,0\cdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 2 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0\,1\,2 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 2 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0\,0\,1 \\
 & & \vdots & & & & & & & & \vdots & & & & & & & & \vdots & & & & & &
\end{array}\right]$$

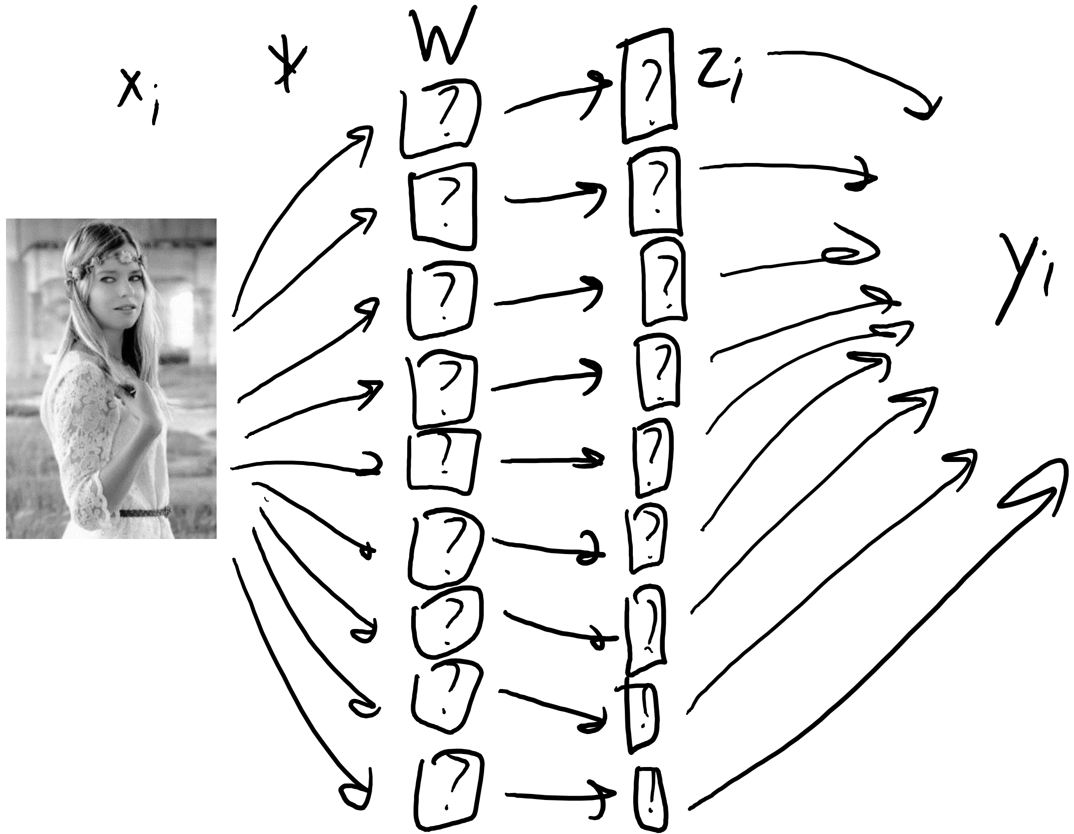
Last Time: Convolutional Neural Networks

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



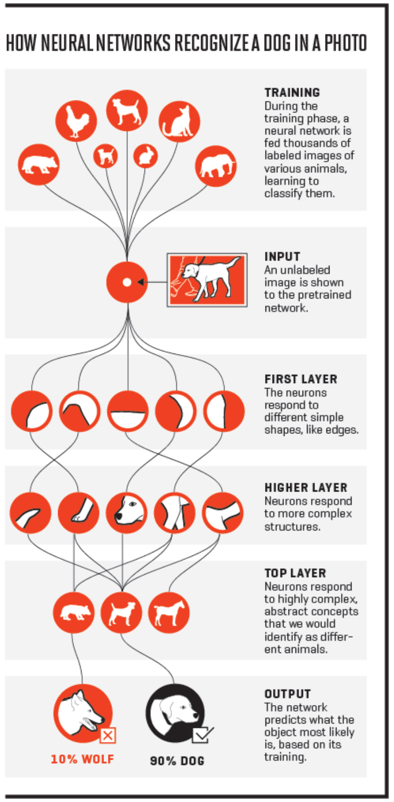
Last Time: Convolutional Neural Networks

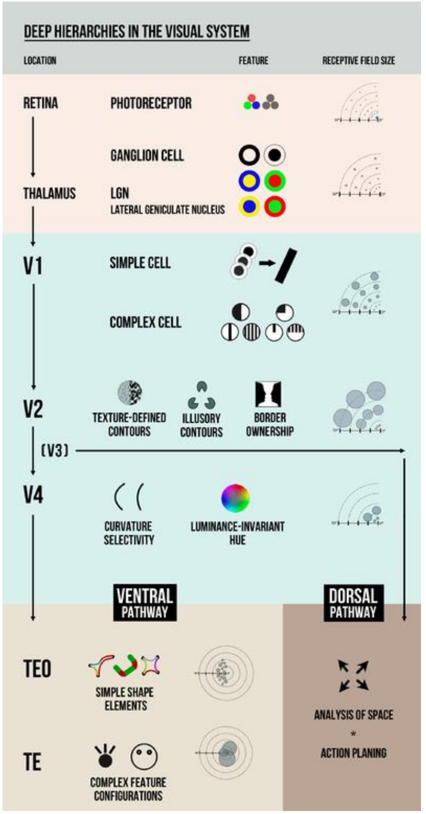
- Convolutional neural networks learn the features:
  - Learning ‘W’ and ‘v’ automatically chooses types/variances/orientations.



Last Time: Convolutional Neural Networks

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

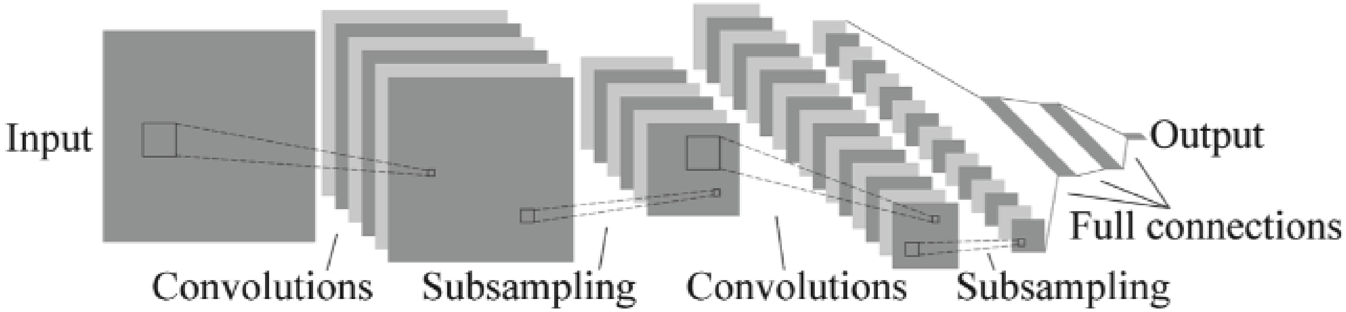




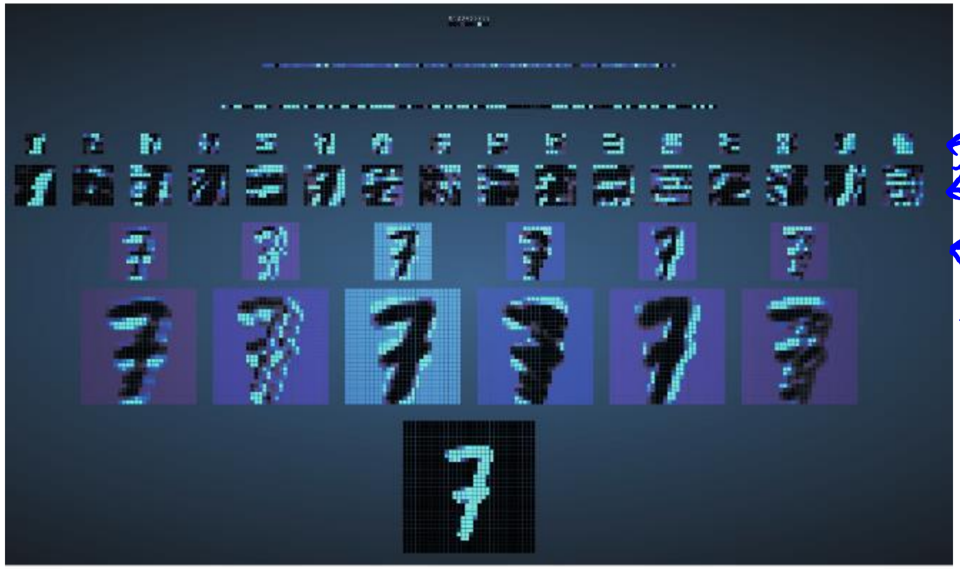
http://fortune.com/ai-artificial-intelligence-deep-machine-learning/
https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing

Last Time: Convolutional Neural Networks

- Classic convolutional neural network (LeNet):



- Visualizing the “activations” of the layers:
  - http://scs.ryerson.ca/~aharley/vis/conv
  - http://cs231n.stanford.edu



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

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

(End of testable content for final exam)

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.

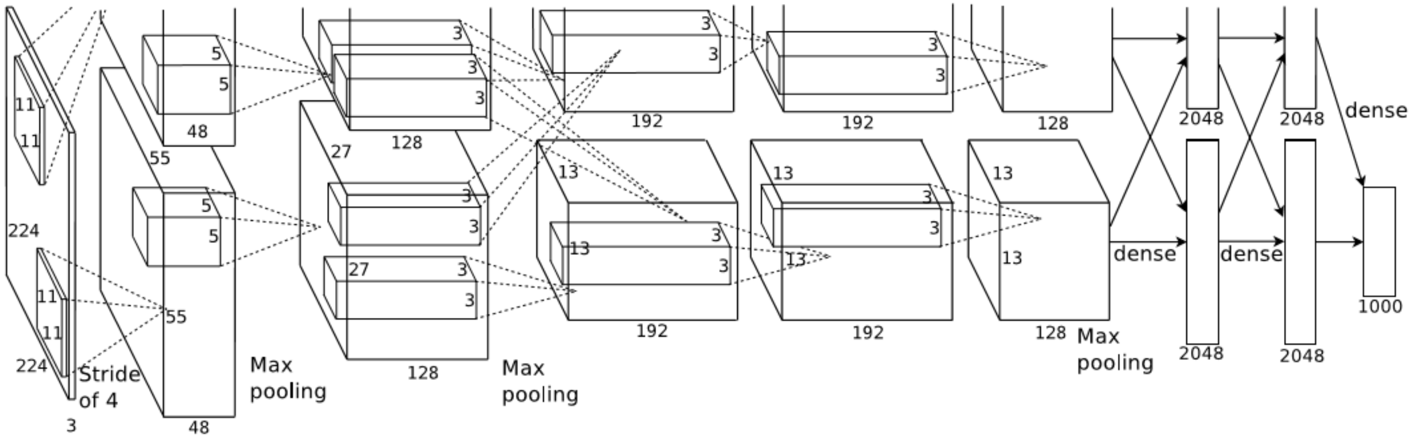


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input 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

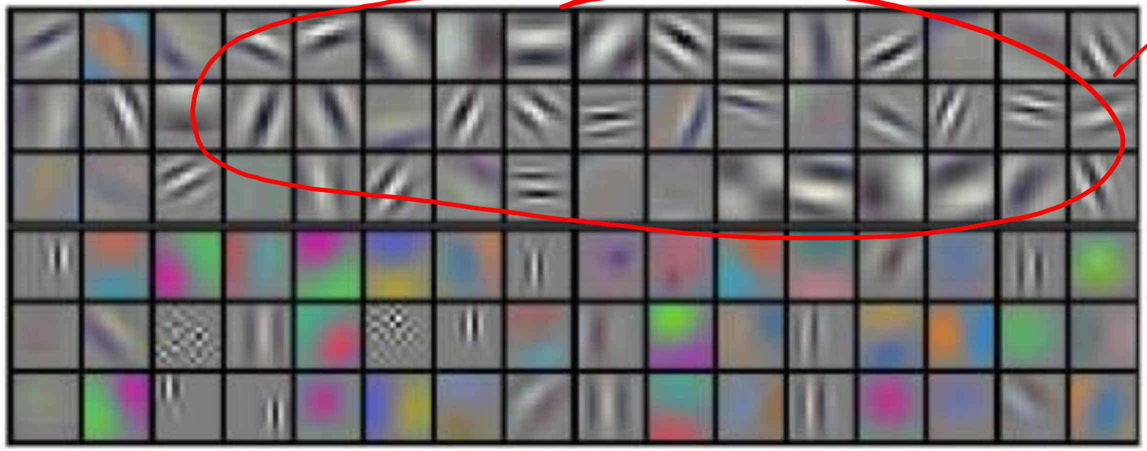
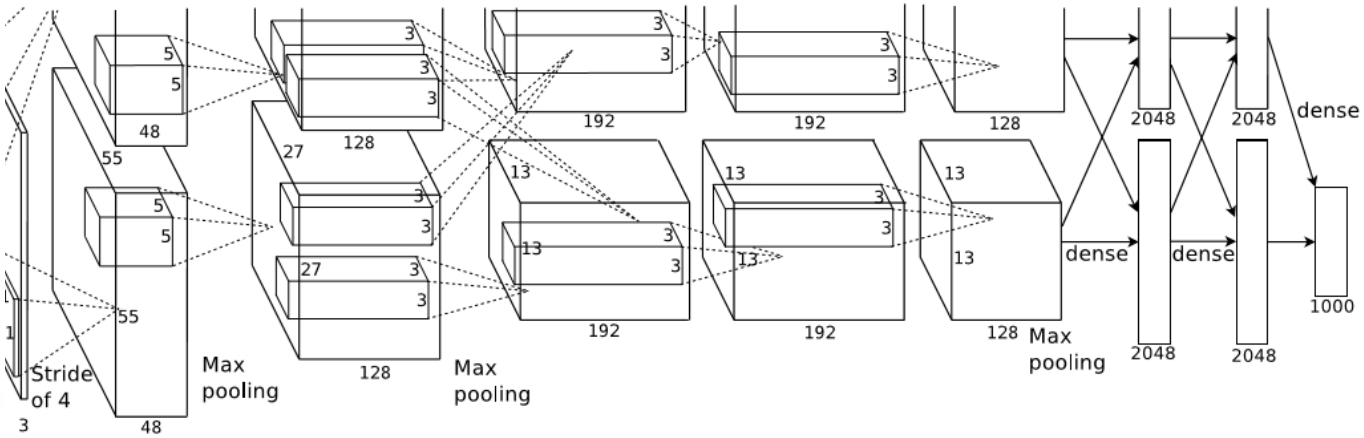


Figure 3: 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images. The



ure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities ween the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts ıe 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 7x7 stride 2 instead of 11x11 stride 4).
  – Introduced deconvolutional networks to visualize what CNNs learn.

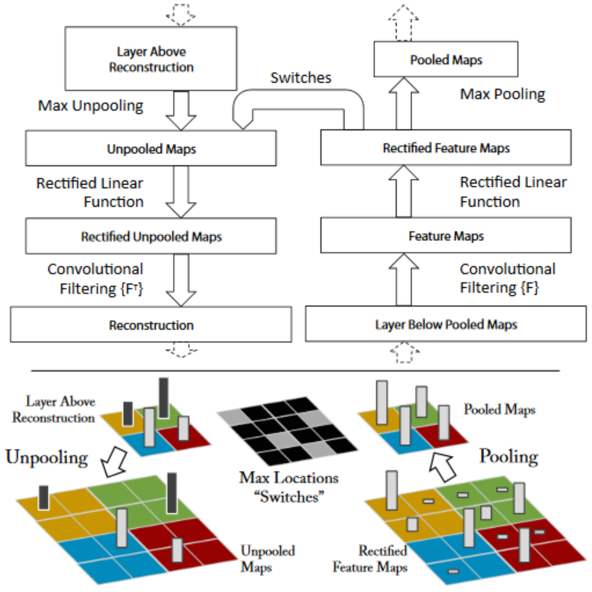
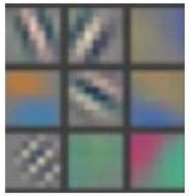
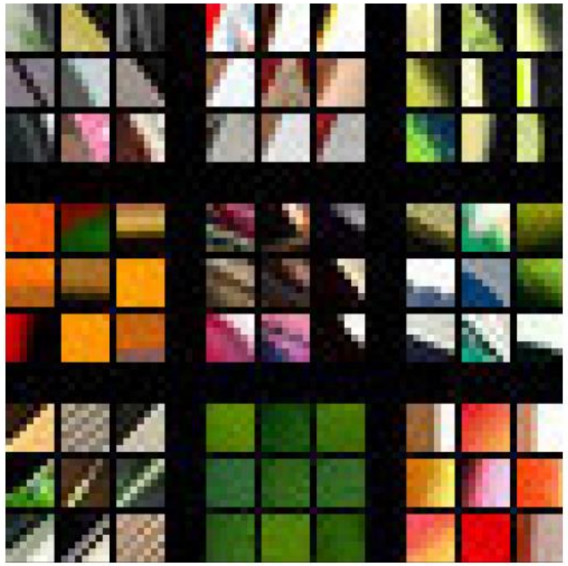


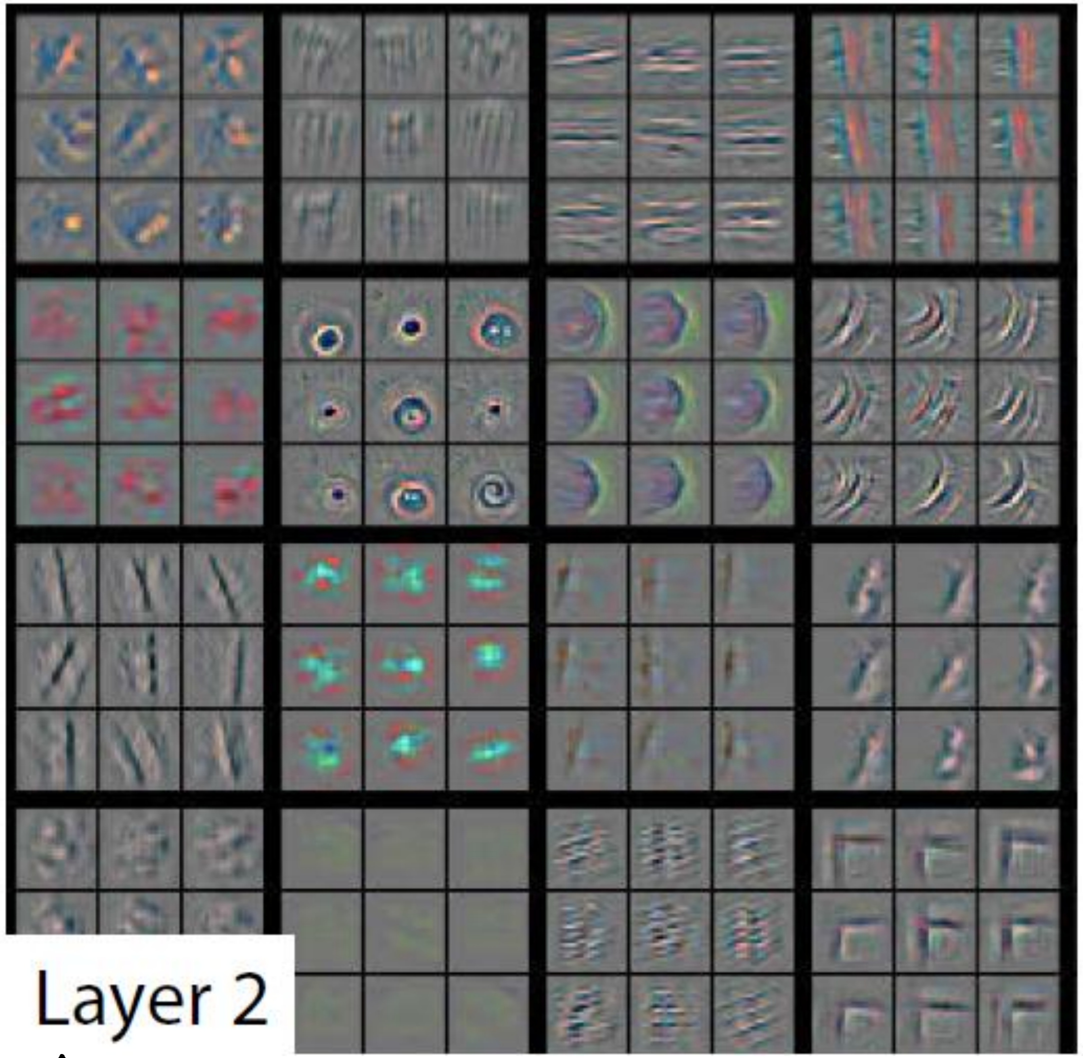
Figure 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local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

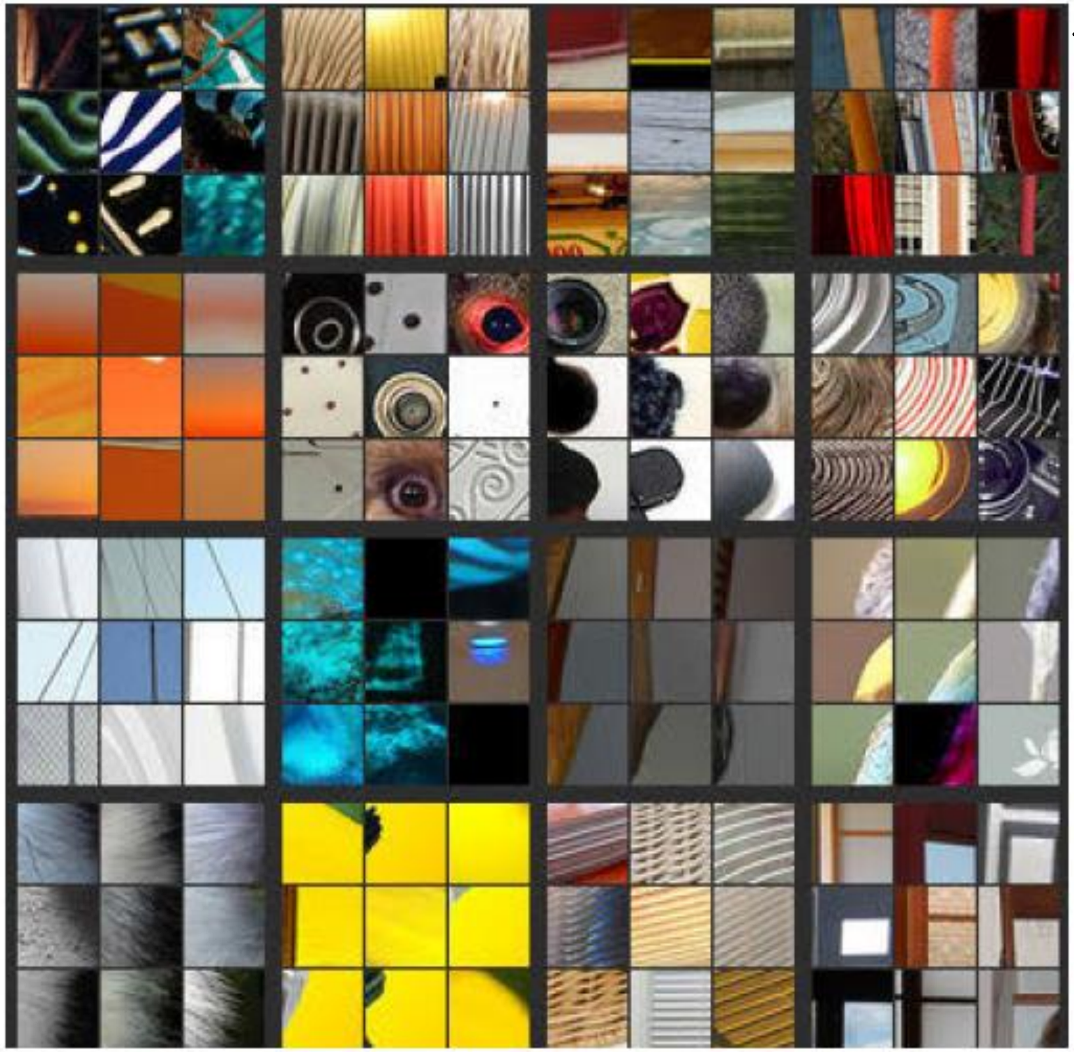


Layer 1



ZFNet Convolutional Neural Network



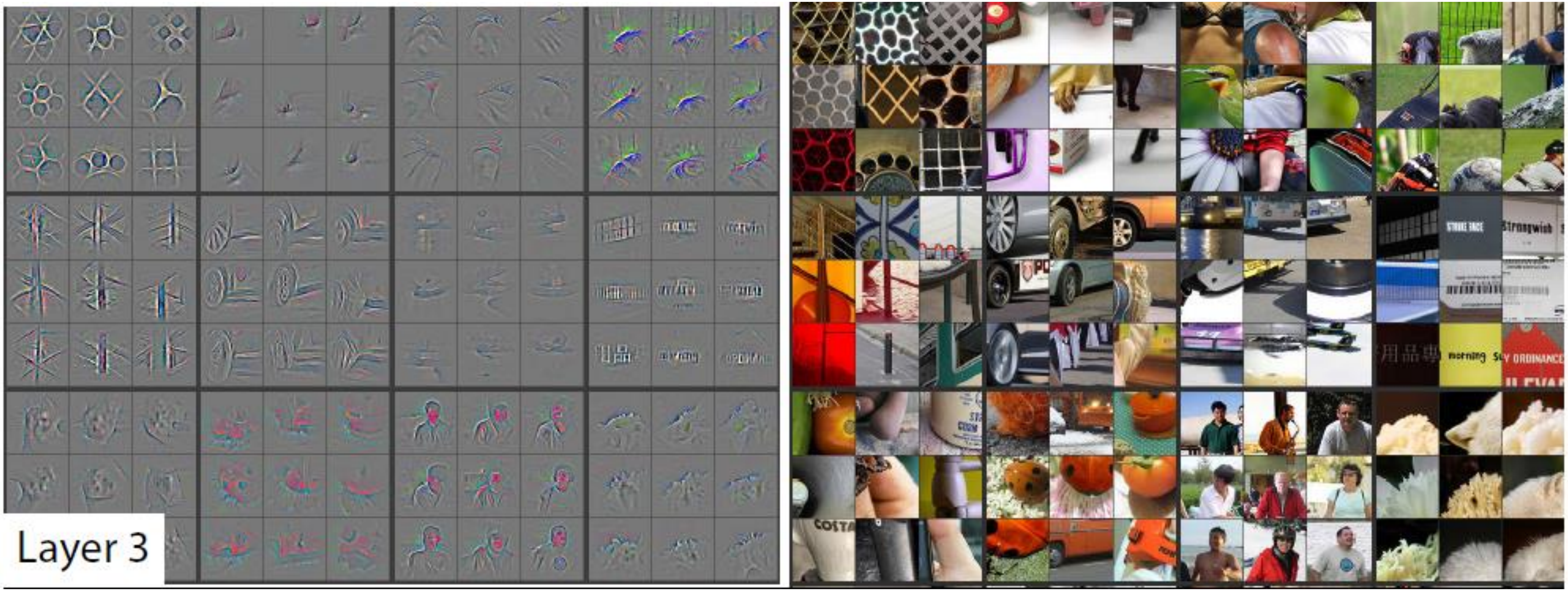


Patch from data giving largest response

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

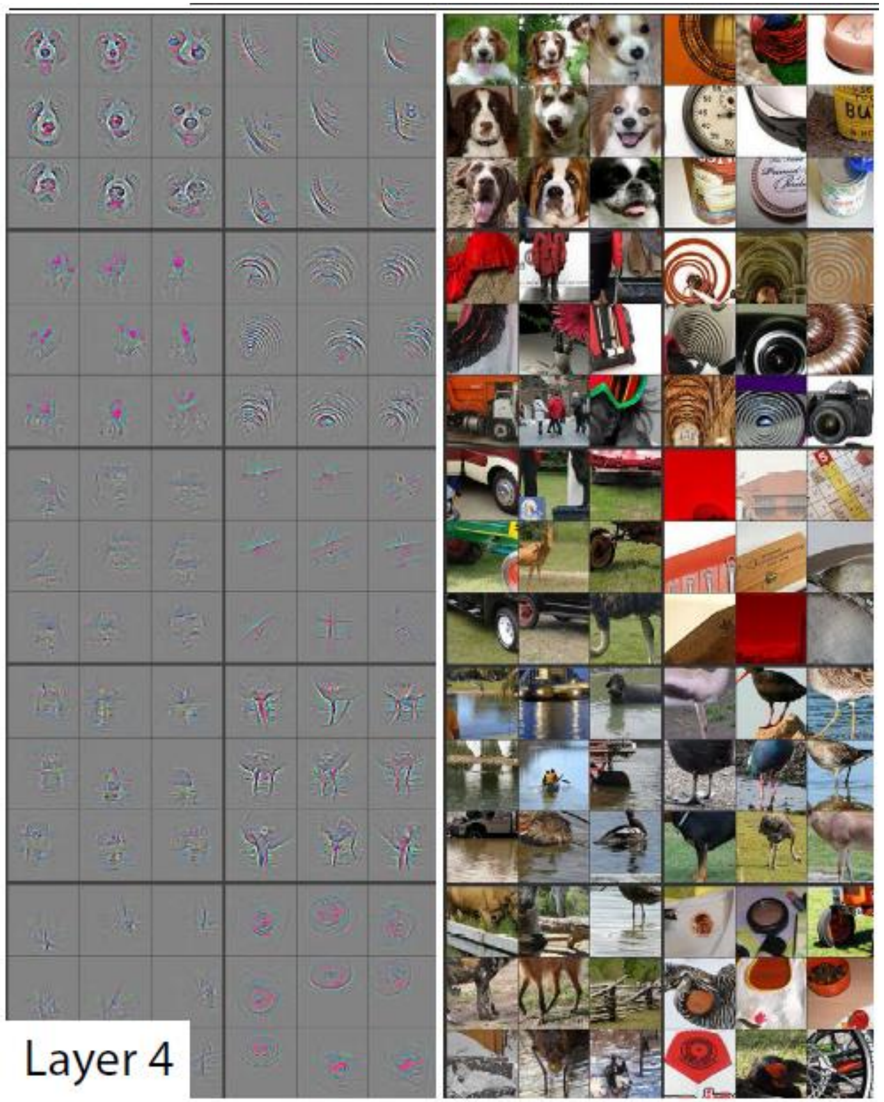
Deconvolution network giving patch that leads to largest response

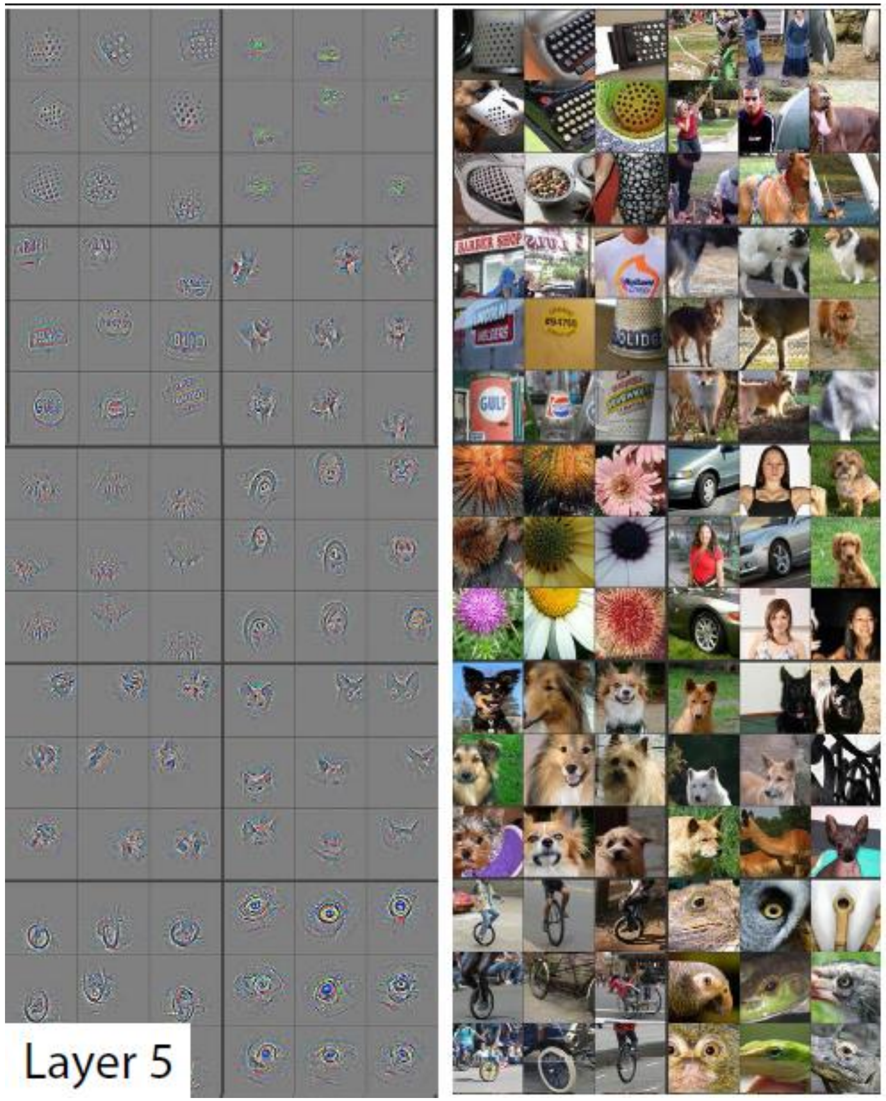
ZFNet Convolutional Neural Network



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

ZFNet Convolutional Neural Network

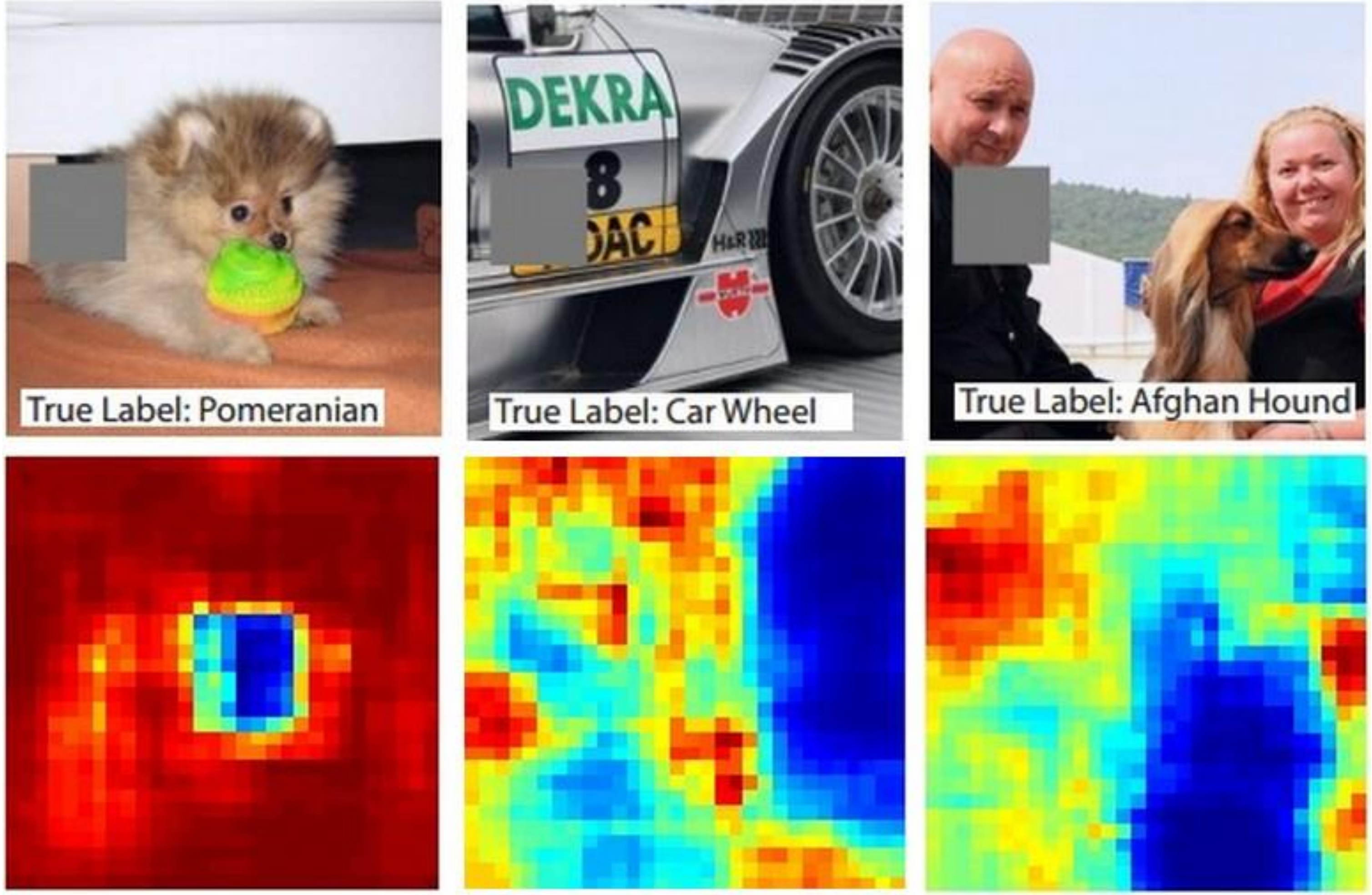




https://arxiv.org/pdf/1311.2901v3.pdf

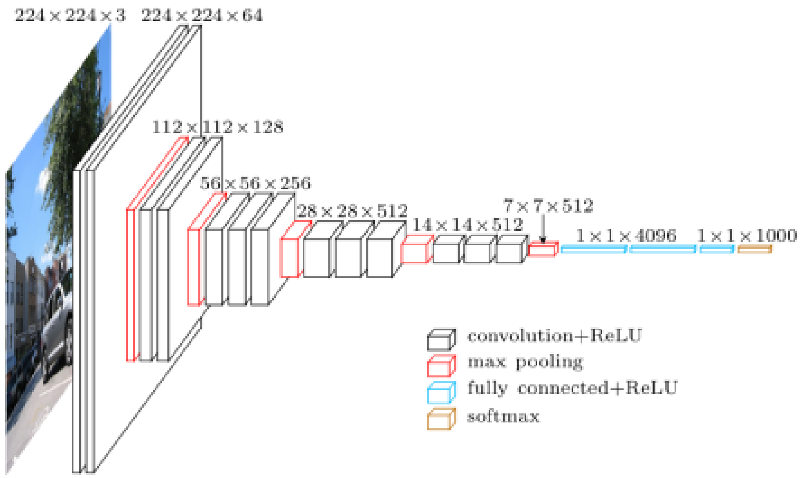
ZFNet Convolutional Neural Network

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



VGG Convolutional Neural Network

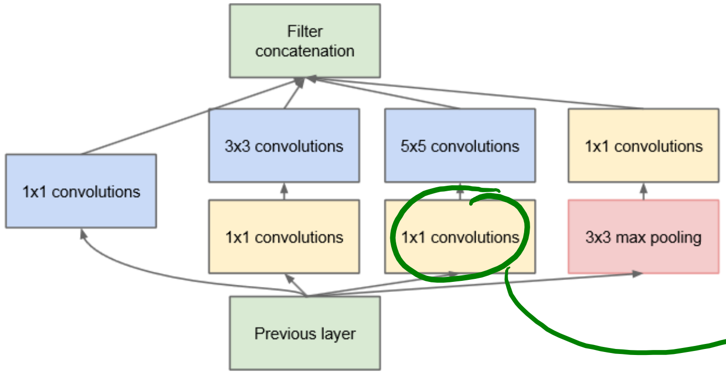
- Image 2014 "Localization" Task won by a 19-layer VGG network:
  - 7.3% error for classification (2nd place).
  - Uses 3x3 convolution layers with stride of 1:
    - 3x3 followed by 3x3 simulates a 5x5, and another 3x3 simulates a 7x7, and so on.
    - Speeds things up and reduces number of parameters.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



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

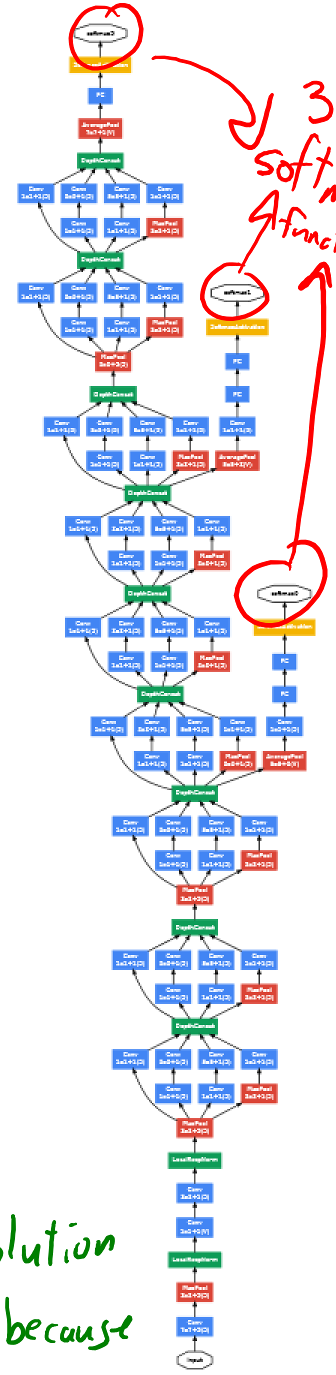
GoogLeNet

- 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

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



3 softmax functions

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.

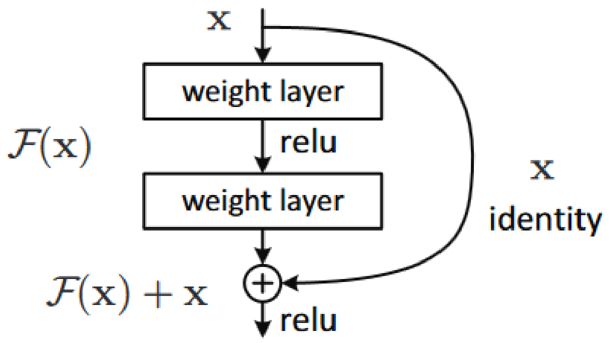


Figure 2. Residual learning: a building block.

    - Network just focuses on "residual": what is not captured in original signal.
    - Along with VGG, this is another of the most popular architectures.

https://arxiv.org/pdf/1512.03385v1.pdf

DenseNet

- More recent variation is "DenseNets":
  - Each layer gets to see all the values in the previous layers.
  - Gets rid of vanishing gradients.

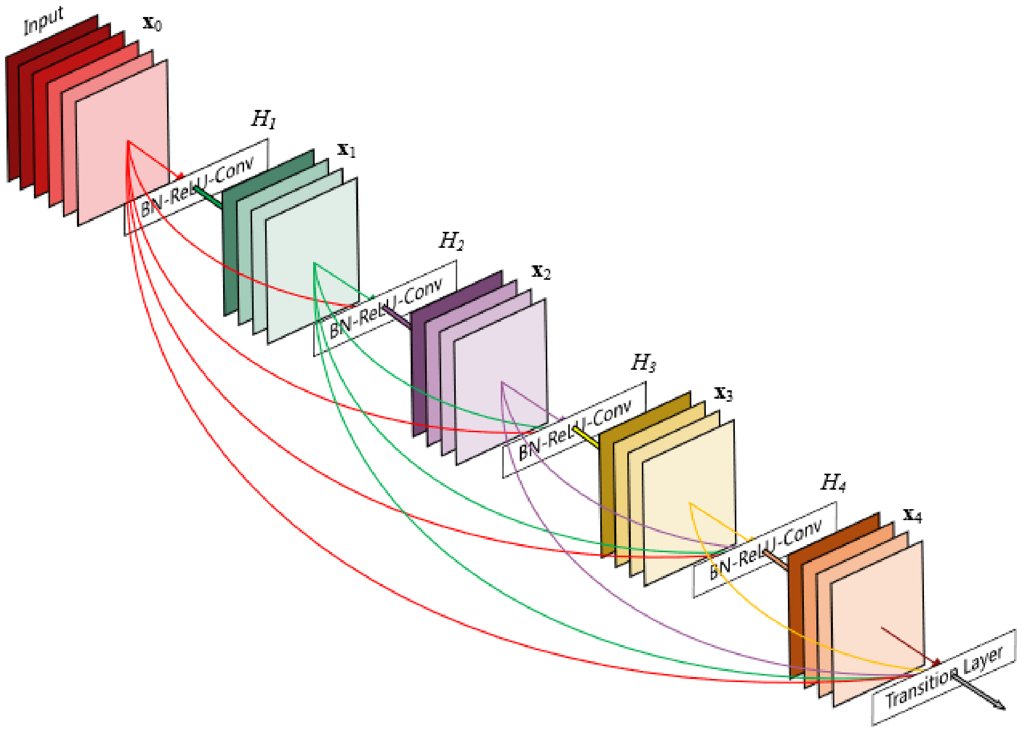


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

https://arxiv.org/pdf/1512.03385v1.pdf

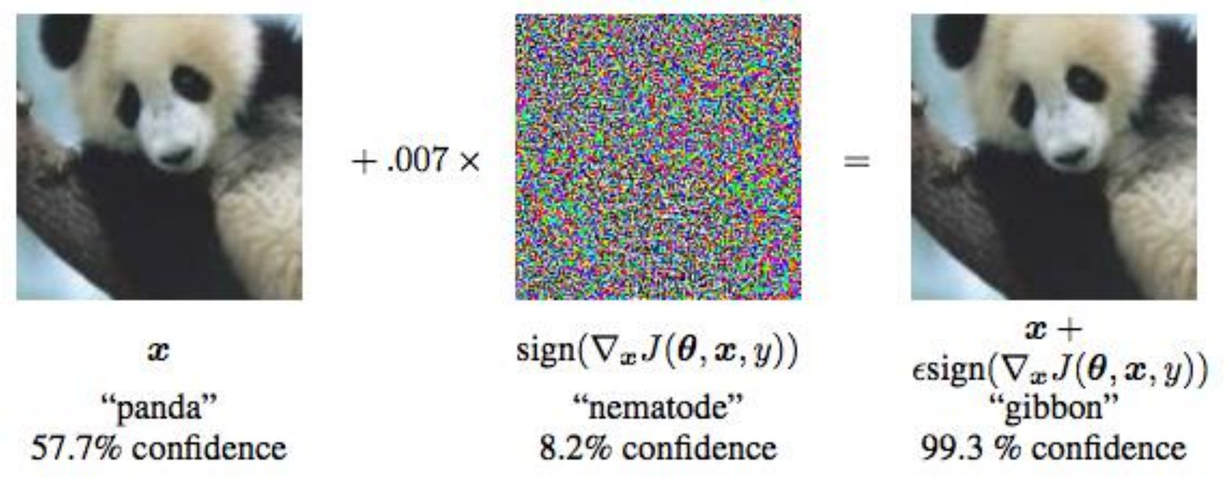
Mission Accomplished?

- For speech recognition and object detection:
  - No other methods have ever given the current level of performance.
  - Deep models continue to improve performance on these and related tasks.
  - We don’t know how to scale up other universal approximators.
  - There is likely some overfitting to popular datasets like ImageNet.

- CNNs are now making their way into products.
  - Apple face recognition.
  - Amazon Go: https://www.youtube.com/watch?v=NrmMk1Myrxc
    - Trolling by French company Monoprix here.
  - Self-driving cars.

Mission Accomplished?

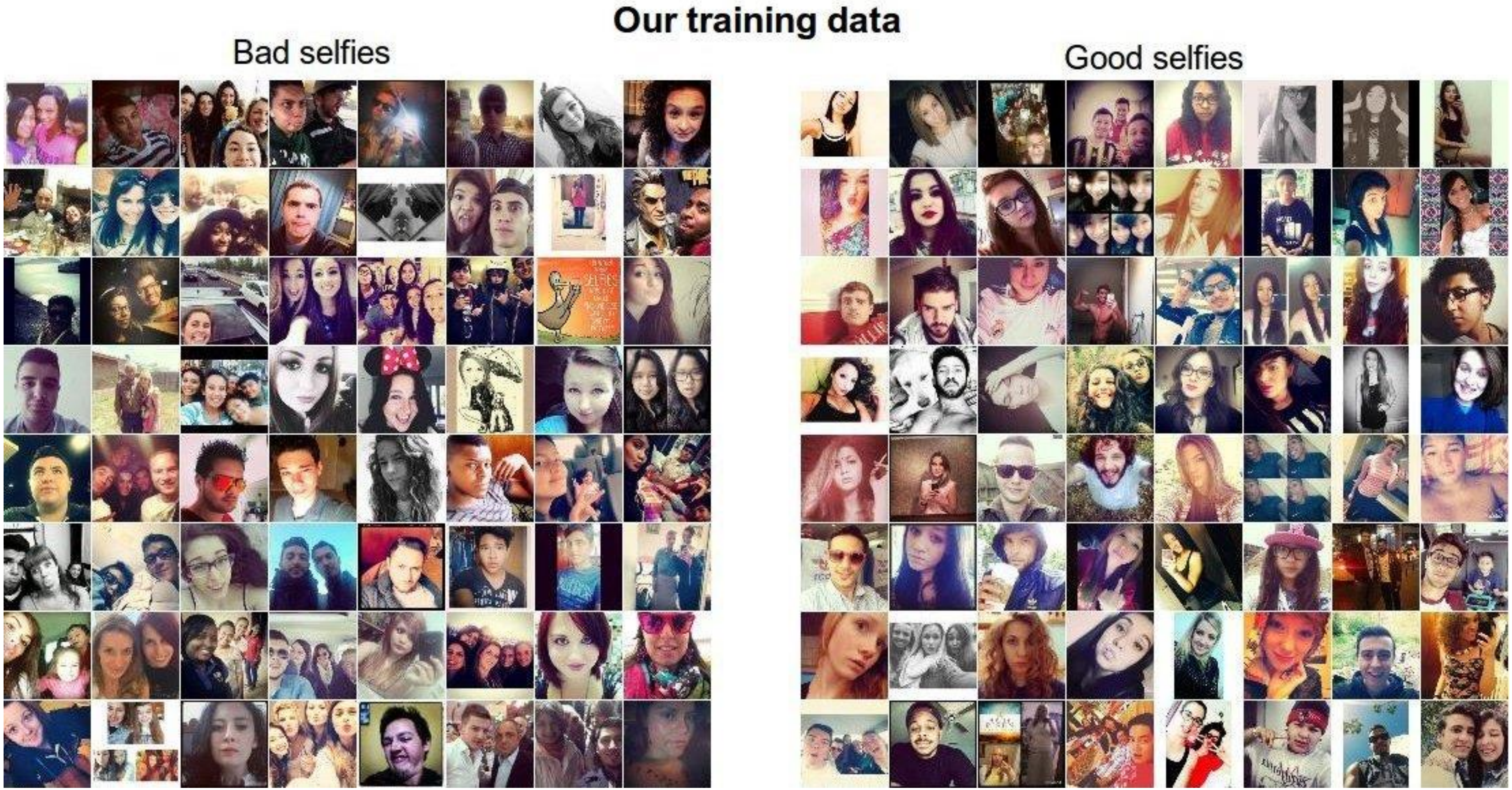
- Despite high-level of abstraction, deep CNNs are easily fooled:
  – But progress on fixing 'blind spots'.
- Recent work: imperceptible noise that changes the predicted label



- Can someone repaint a stop sign and fool self-driving cars?

https://arxiv.org/pdf/1412.6572.pdf, https://blog.openai.com/adversarial-example-research/

CNNs for Rating Selfies

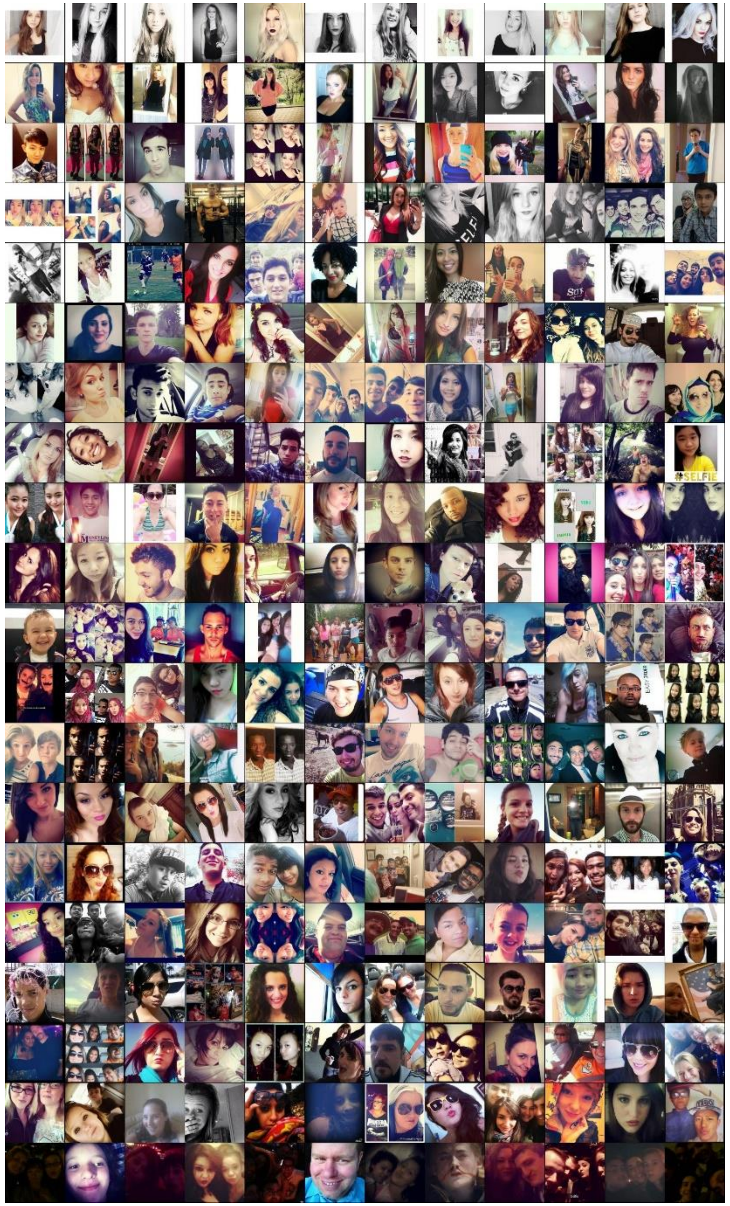


https://karpathy.github.io/2015/10/25/selfie/

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



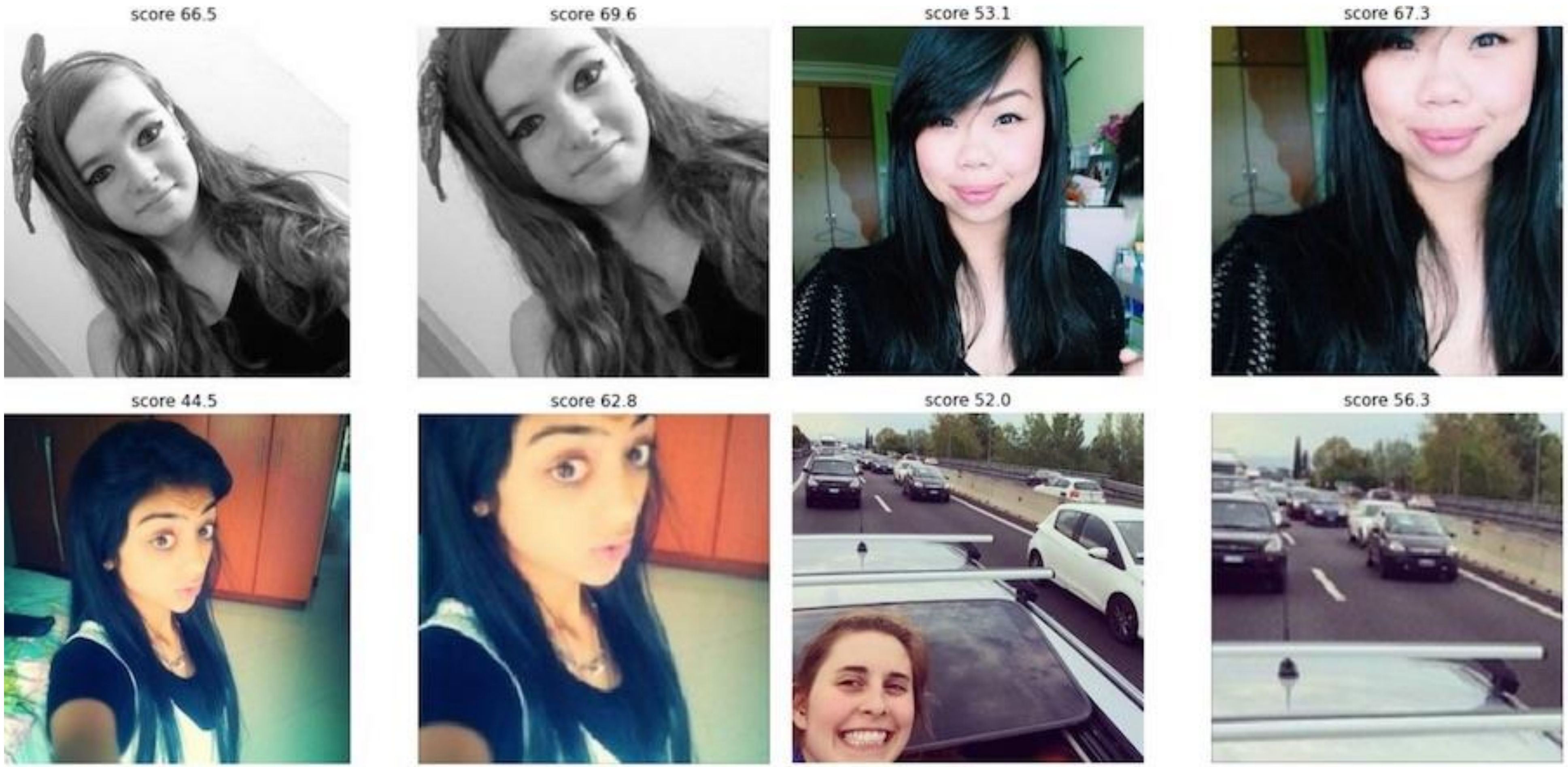
Don't:

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

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

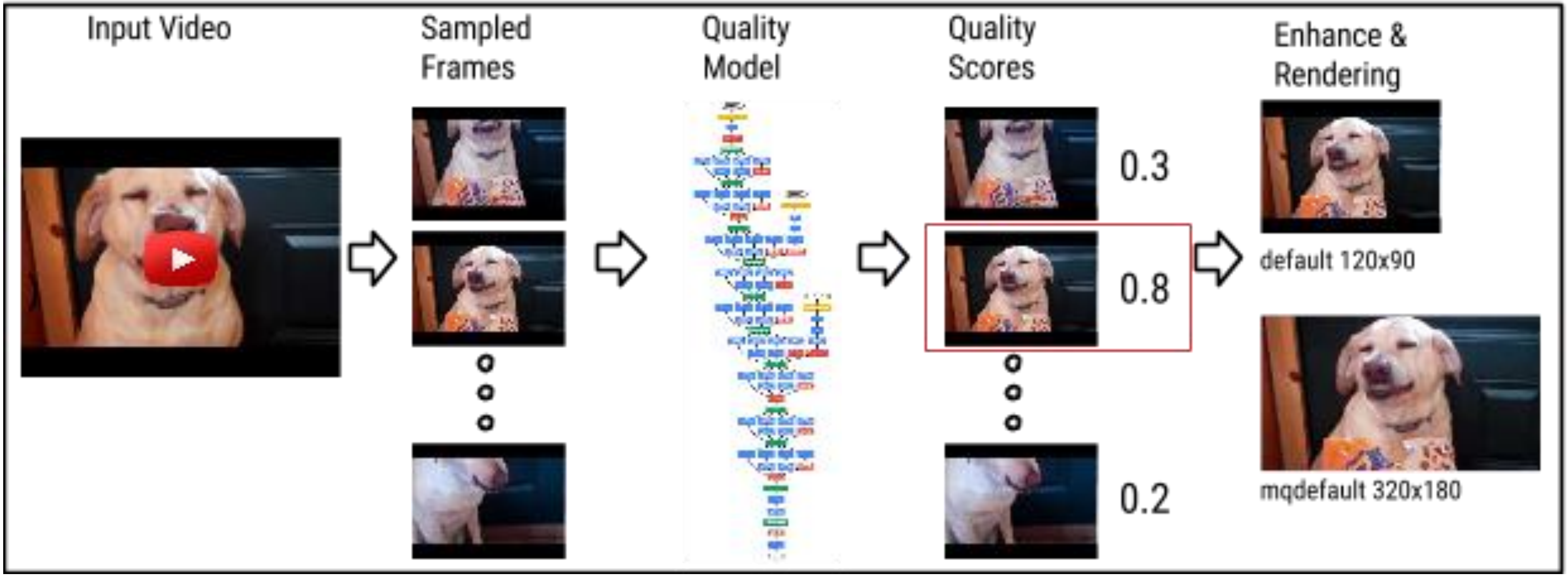
CNNs for Rating Selfies

Finding best image crop:



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

CNNs for Choosing YouTube Thumbnails



https://youtube-eng.googleblog.com/2015/10/improving-youtube-video-thumbnails-with_8.html

Beyond Classification (CPSC 540)

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

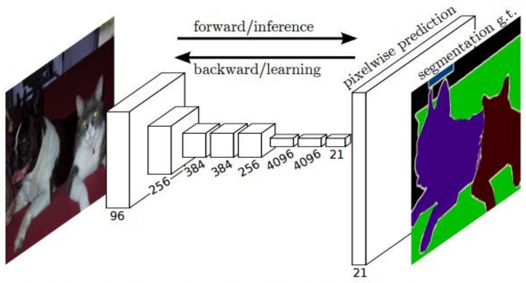


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

- Image segmentation:

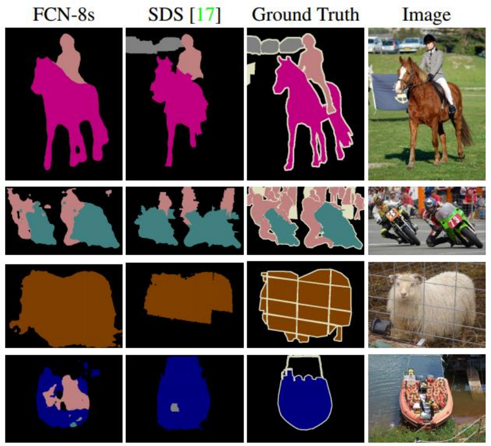


Figure 6. Fully convolutional segmentation nets produce state-of-the-art performance on PASCAL. The left column shows the output of our highest performing net, FCN-8s. The second shows the segmentations produced by the previous state-of-the-art system by Hariharan *et al.* [17]. Notice the fine structures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

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

Beyond Classification (CPSC 540)

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

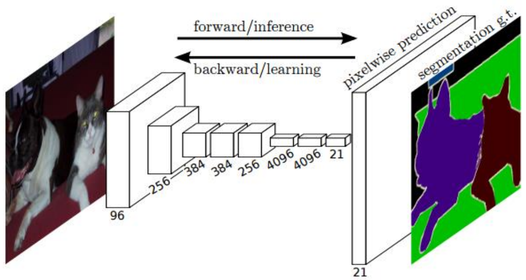
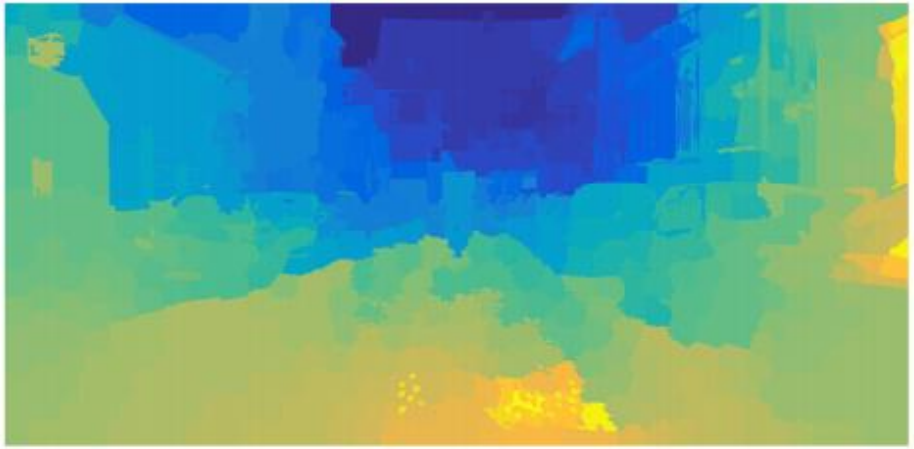


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

- Depth Estimation:



Beyond Classification

- Image colorization:

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

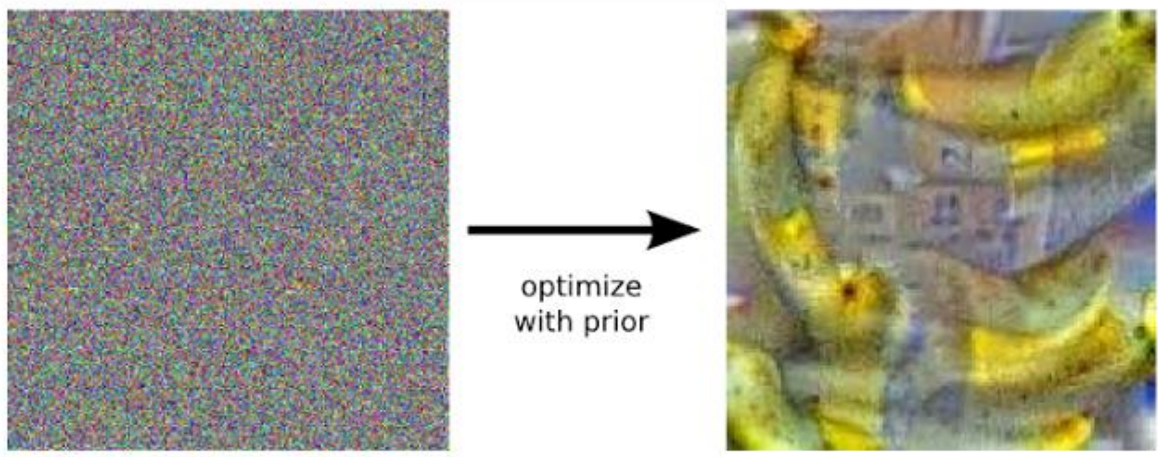
  - Image Gallery, Video

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 x_{ij}:
    - Encourages neighbouring x_{ij} to be similar.

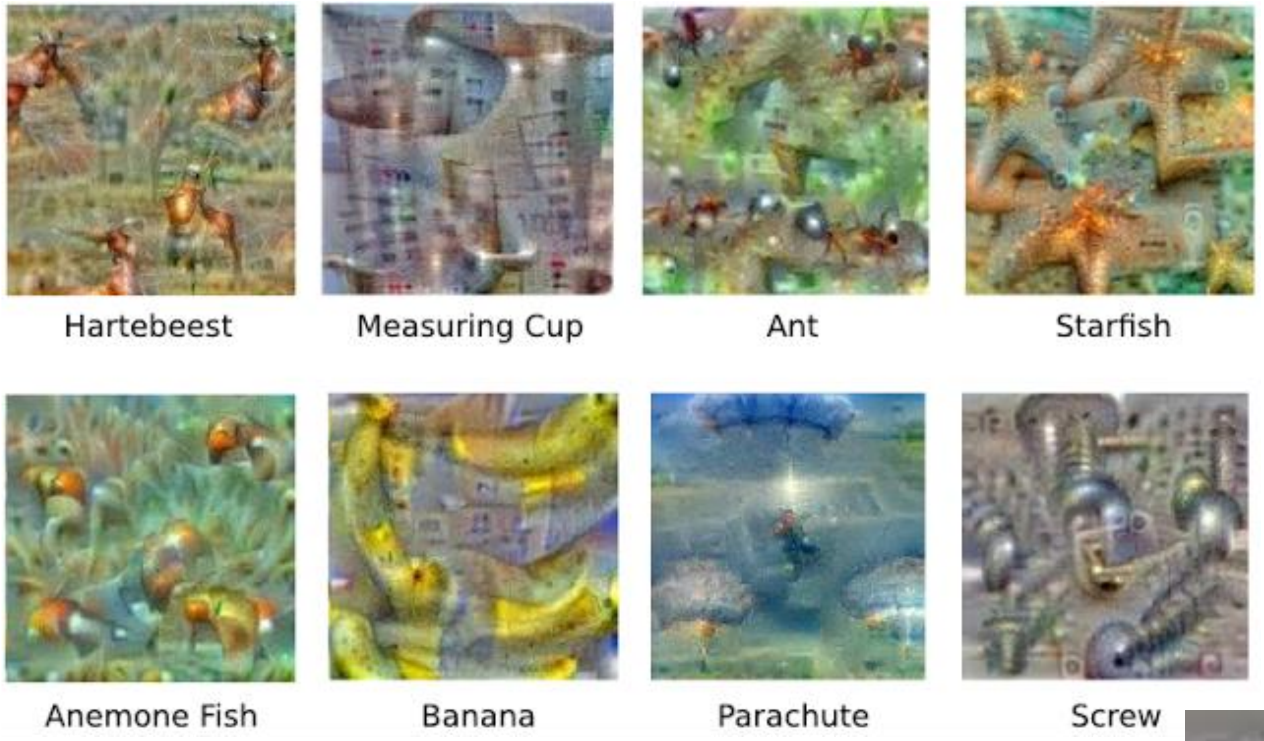
"Show what you think a banana looks like"



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

Inceptionism

- Inceptionism for different class labels:

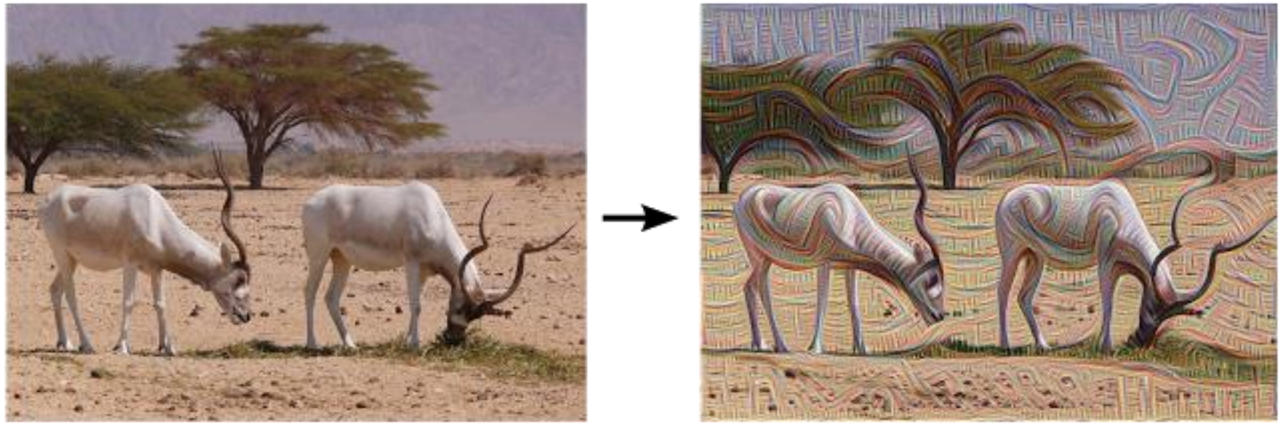


Dumbbell

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

Inceptionism

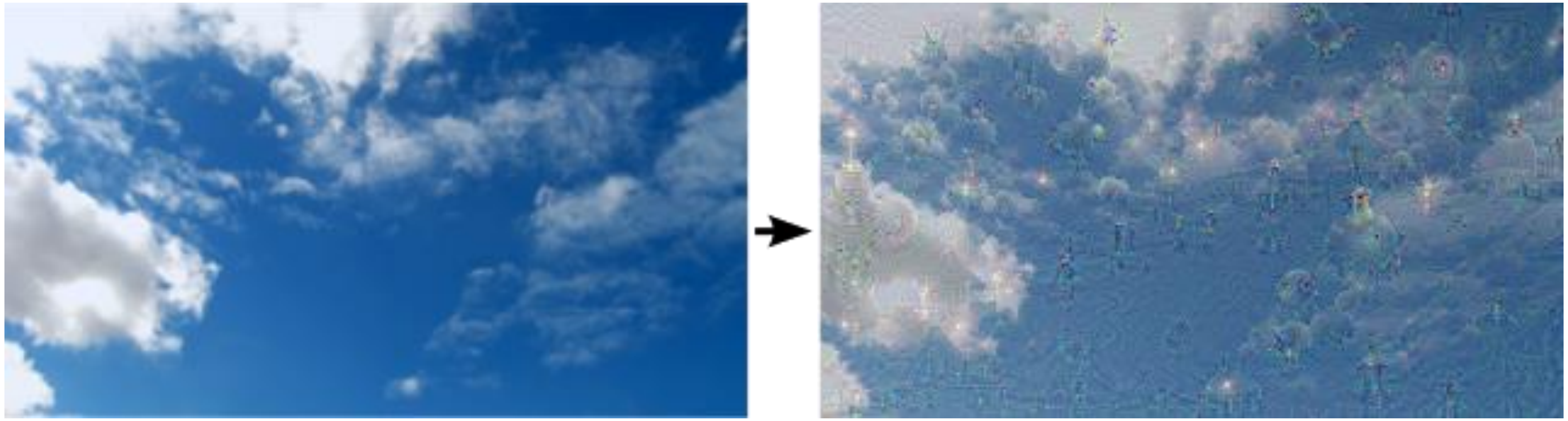
- Inceptionism where we try to match $z_i^{(m)}$ values instead of y_i.
  - Shallow 'm':

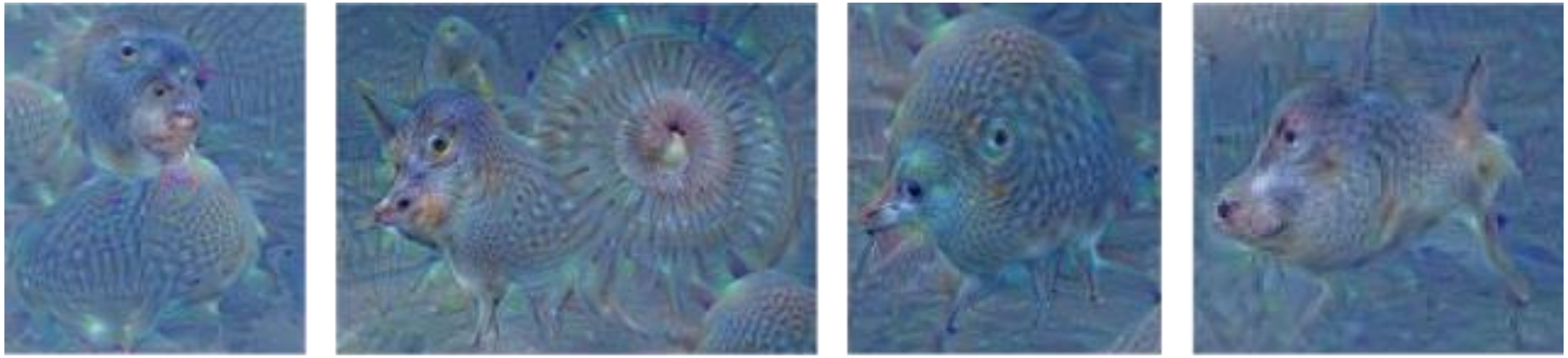


http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

Inceptionism

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

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 ‘S’.

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):

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

Artistic Style Transfer

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

Input style

Input content

Ours

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 a former CPSC 340 student/TA's paper:

(Course Wrap-Up)

CPSC 340: Overview

1. Intro to supervised learning (using counting and distances).
   - Training vs. testing, parametric vs. non-parametric, ensemble methods.
   - Fundamental trade-off, no free lunch, universal consistency.
2. Intro to unsupervised learning (using counting and distances).
   - Clustering, outlier detection, finding similar items.
3. Linear models and gradient descent (for supervised learning)
   - Loss functions, change of basis, regularization, feature selection.
   - Gradient descent and stochastic gradient.
4. Latent-factor models (for unsupervised learning)
   - Typically using linear models and gradient descent.
5. Neural networks (for supervised and multi-layer latent-factor models).

Topics from Previous Years

- Slides for other topics that were covered in previous years:
  - Association rules: find sets of items that are frequently bought together.
  - Ranking: finding “highest ranked” training examples (Google PageRank).
  - Semi-supervised: using unlabeled data to help supervised learning.
  - Sequence mining: approximate matching of patterns in large sequences.

- In previous years we did a course review on the last day:
  - Overview of topics covered in 340, and topics coming in 540.
  - Slides here: this could help with studying for the final.

CPSC 340 vs. CPSC 540

- Goals of CPSC 340: practical machine learning.
  - Make accessible by avoiding some technical details/topics/models.
  - Present most of the fundamental ideas, sometimes in simplified ways.
  - Choose models that are widely-used in practice.
- Goals of CPSC 540: research-level machine learning.
  - Covers complicated details/topics/models that we avoided.
  - Targeted at people with algorithms/math/stats/numerical background.
  - Goal is to be able to understand ICML/NIPS papers at the end of course.
- Example 540 topics:
  - How many iterations of gradient descent do we need?
  - What if y_i is a sentence or an image or a protein? (Graphical models and RNNs.)
  - What if data isn't IID?

Other ML-Related Courses

- CPSC 532R:
  - Probabilistic graphical models.
- CPSC 532L:
  - Deep learning for vision, sound, and language.
- STAT 406:
  - Similar/complementary topics, focus on mathematical details and applications.
- STAT 460/461:
  - Advanced statistical issues (what happens when 'n' goes to ∞?)
- STAT 5xx
  - These all cover related topics.
- EECE 592:
  - Deep learning and reinforcement learning.
- EOSC 510:
  - Similar/complementary topics, emphasis on EOSC applications.
- EOSC 550:
  - Optimization methods for deep learning.
- LIBR 559d:
  - Language and social media data.

Final Slide: Data Science Job Board

- Data Science Job Board: http://makedatasense.ca/jobs
  - Make a profile here if you are looking for a job in this area.
  - Usually there are more companies listed than people!

- Thanks for listening and good luck on this/other finals!

(That's all I have to say about ML for 2017…)

WORK

Data Science Job Board

Browse Data Science jobs and post your own Data Scientist profile for other companies to see.

Click here to browse jobs and post your profile