CPSC 340: Machine Learning and Data Mining

More CNNs

Fall 2018

Admin

- This lecture may go overtime.
  - Assuming there isn't a class here at 5pm.
  - I won't be offended if you leave early.
  - Extra time won't be testable.

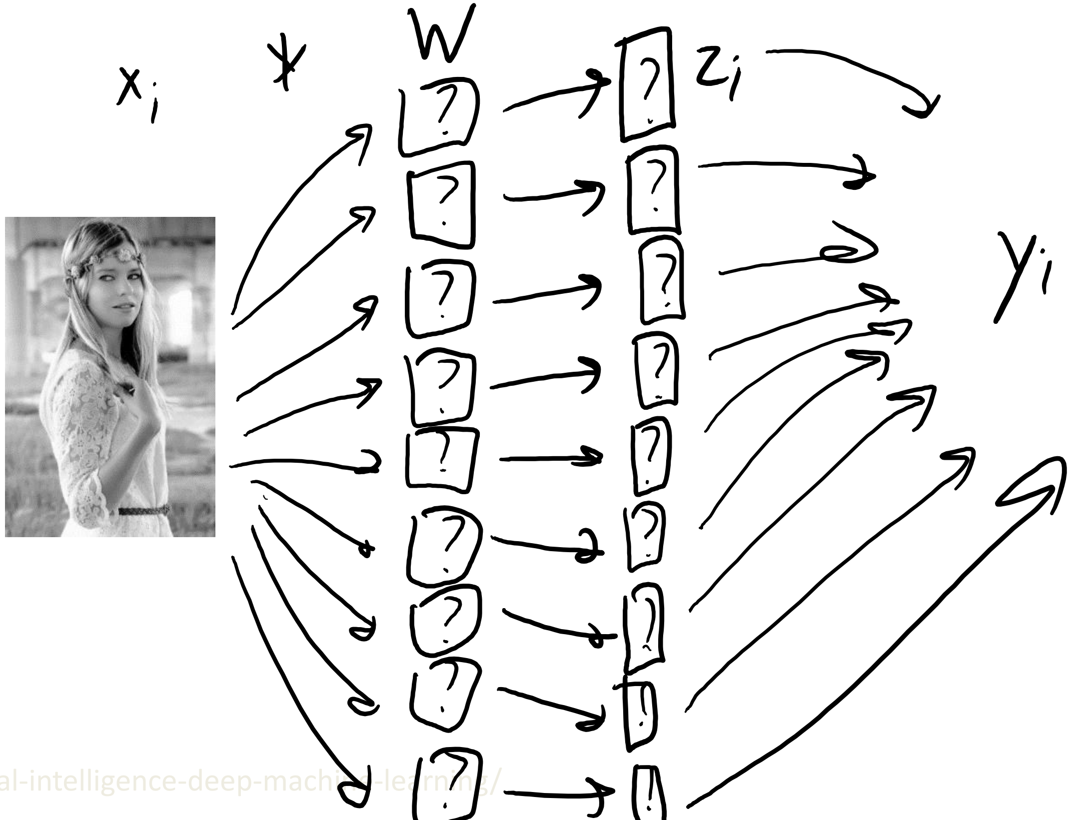
- Friday's lectures:
  - Mike will do a course review in his section.
  - Aline Tabet will give a guest lecture in this section ("ML Applications in Medicine").

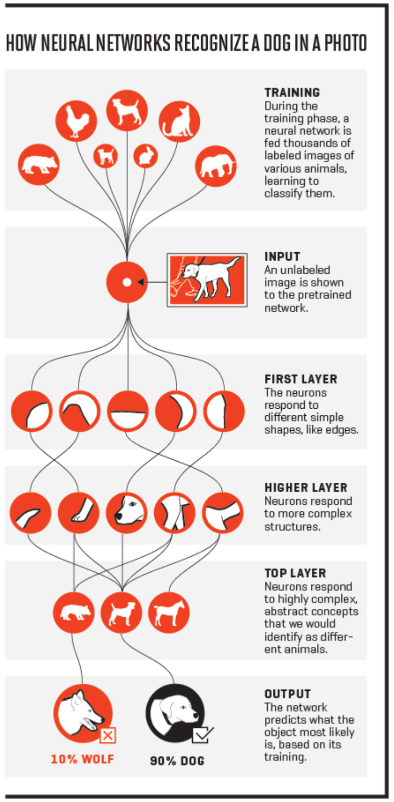
- Final: Thursday December 13th at 8:30am in WOOD 2.
  - Similar style of questions to midterm.
  - 2 pages of notes.

- CPSC 532M students: course project due December 19 (details on Piazza).

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.

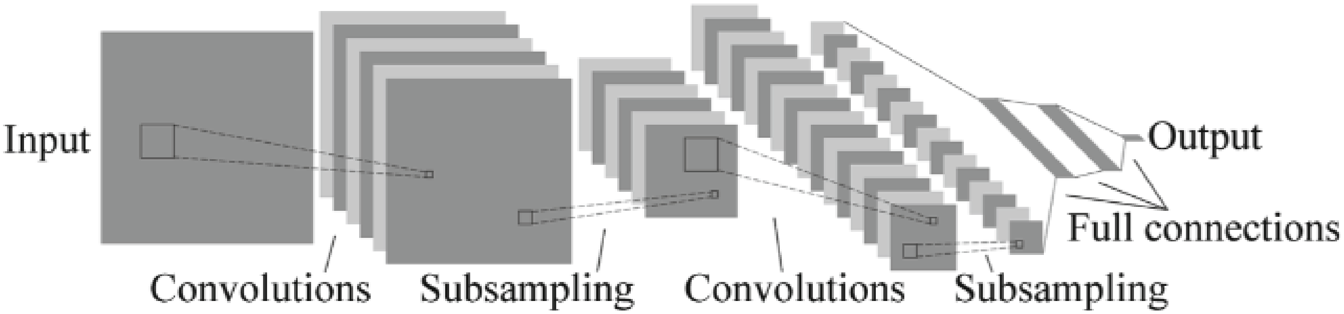




http://fortune.com/ai-artificial-intelligence-deep-machine-learning/

Last Time: Convolutional Neural Networks

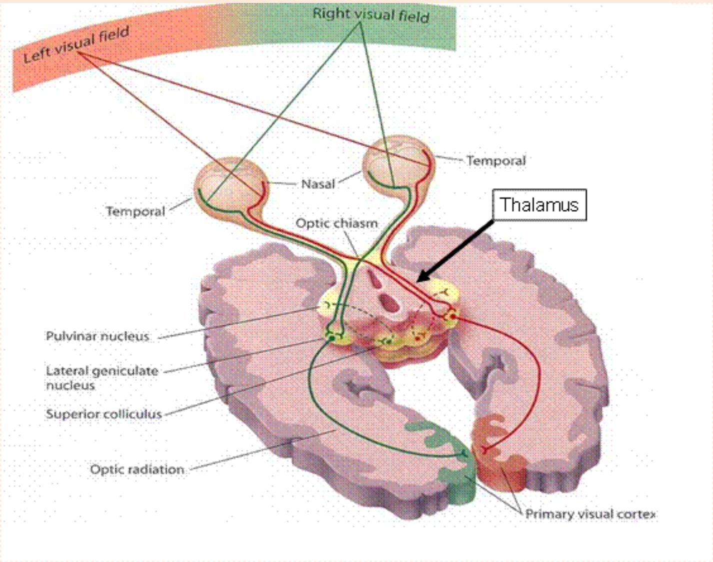
- Classic convolutional neural network (LeNet):

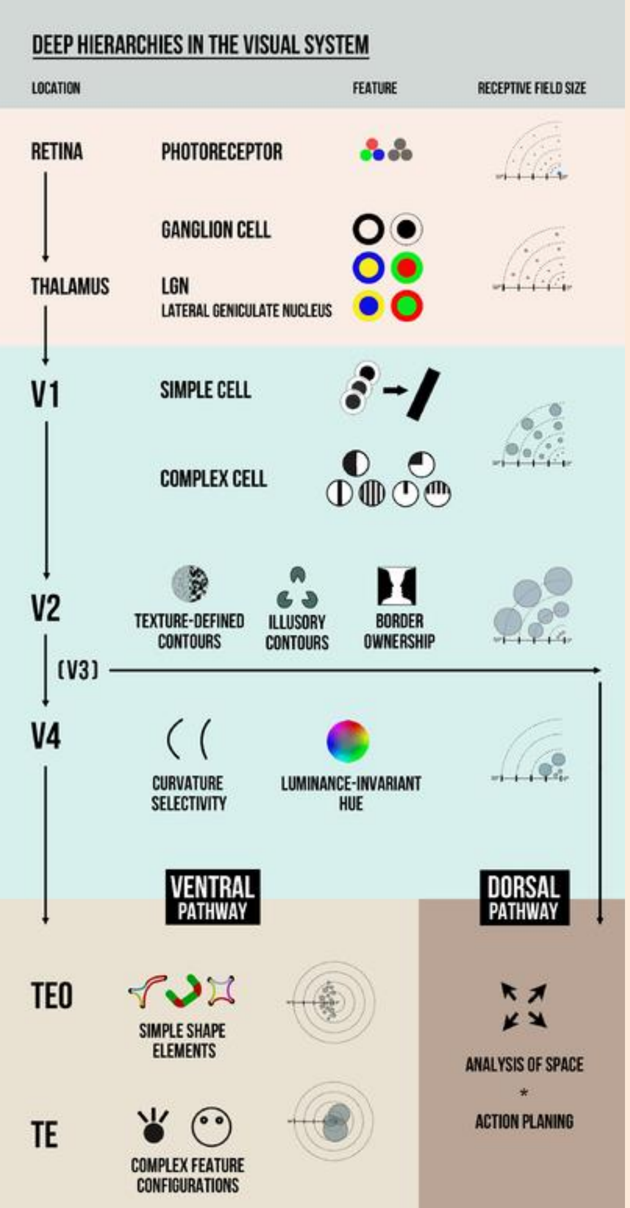


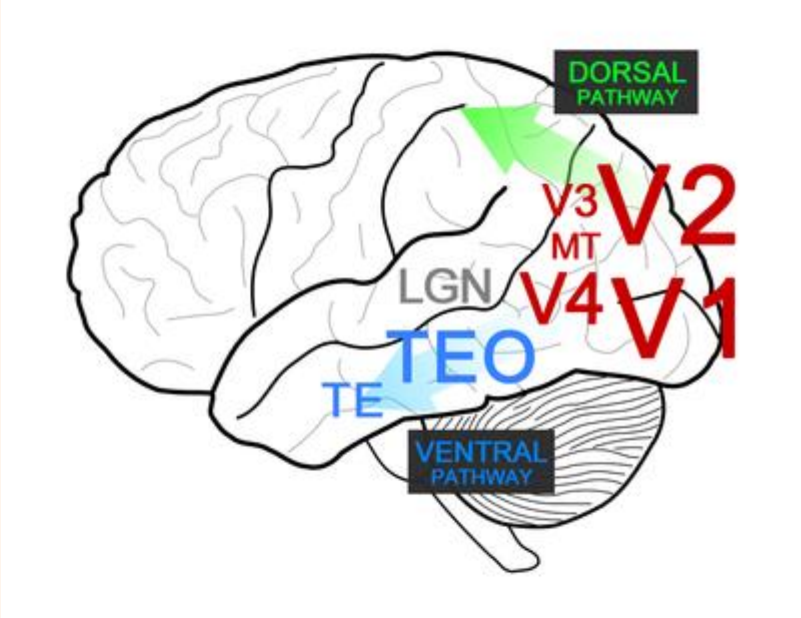
- Visualizing the "activations" of the layers:
  - http://scs.ryerson.ca/~aharley/vis/conv
  - http://cs231n.stanford.edu

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

Deep Hierarchies in the Visual System

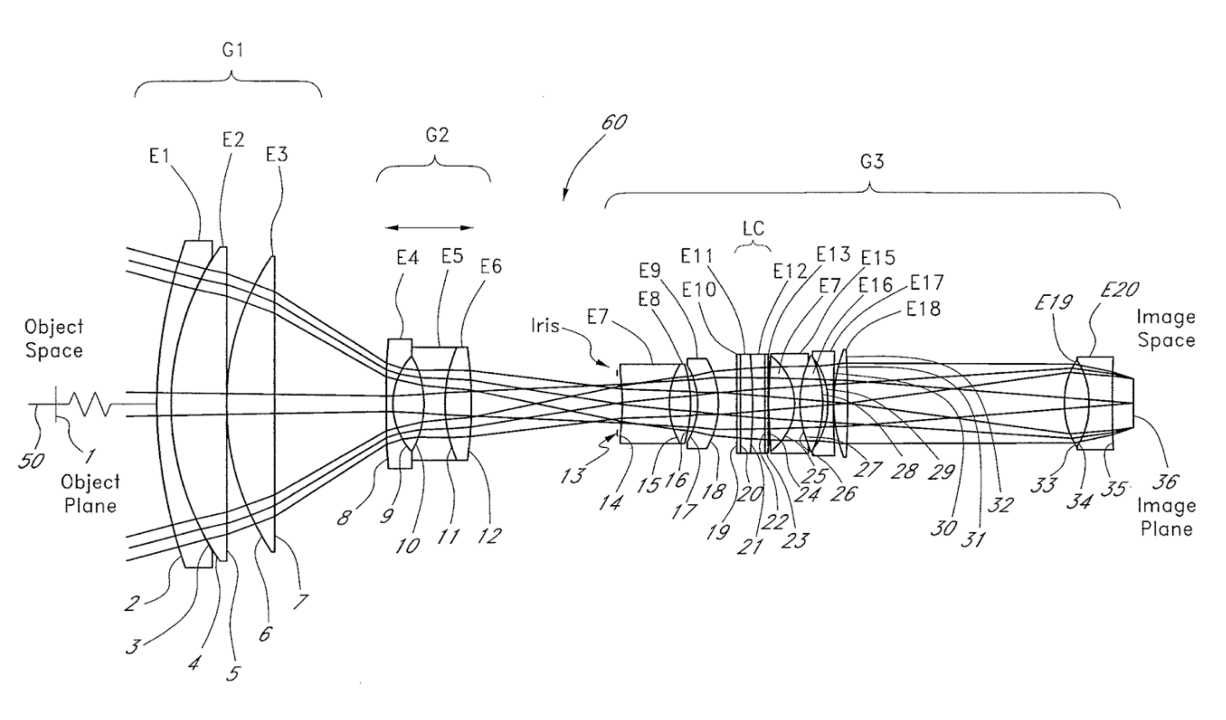


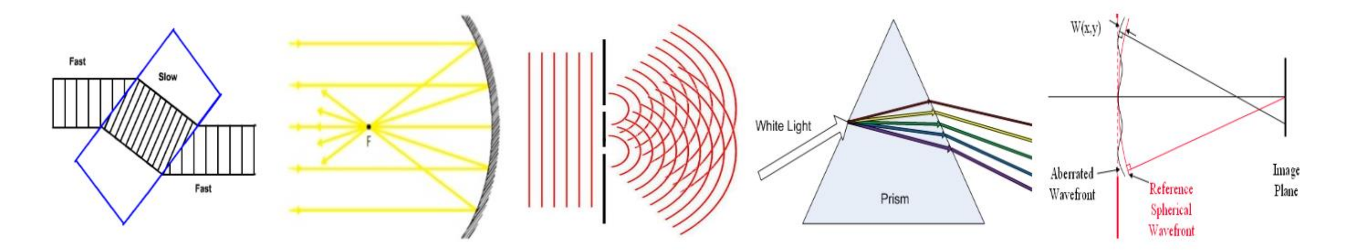




http://www.strokenetwork.org/newsletter/articles/vision.htm
https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing

Deep Hierarchies in Optics





http://www.argmin.net/2018/01/25/optics/

(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.
  - Same networks won in 2013: tweaks like smaller stride and smaller filters.

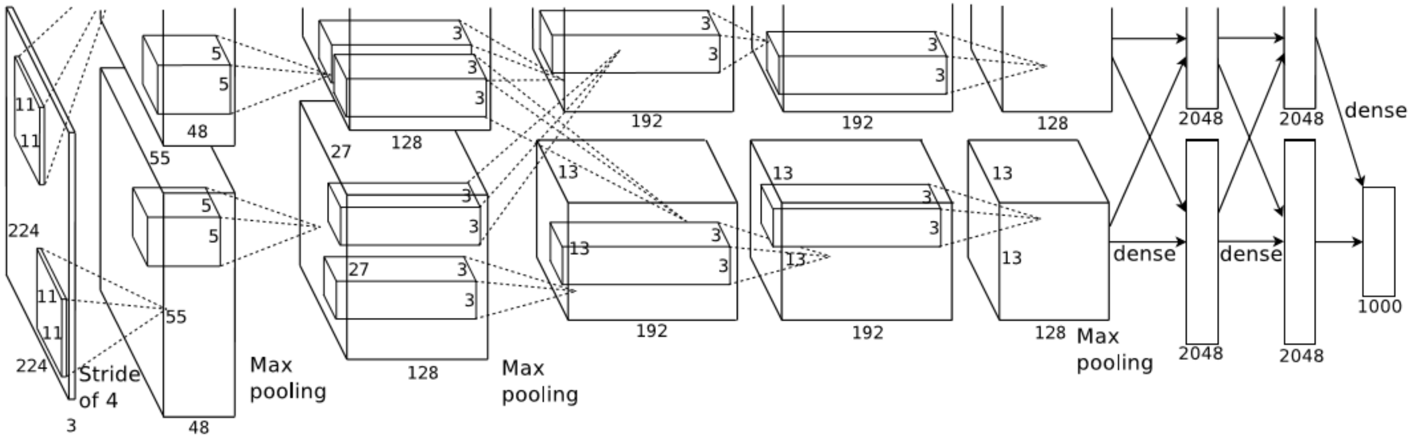
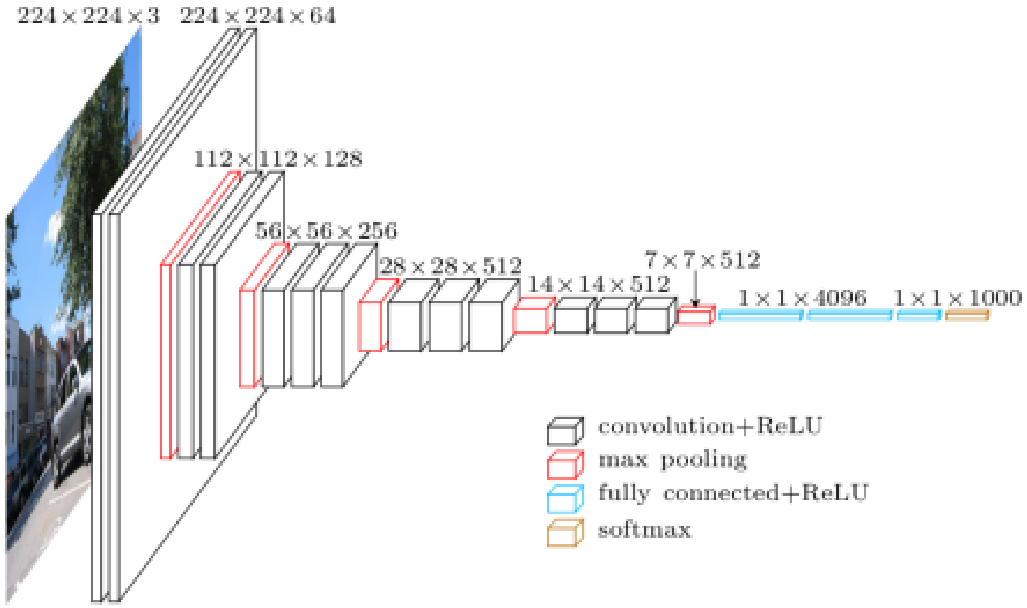


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input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

ImageNet Insights

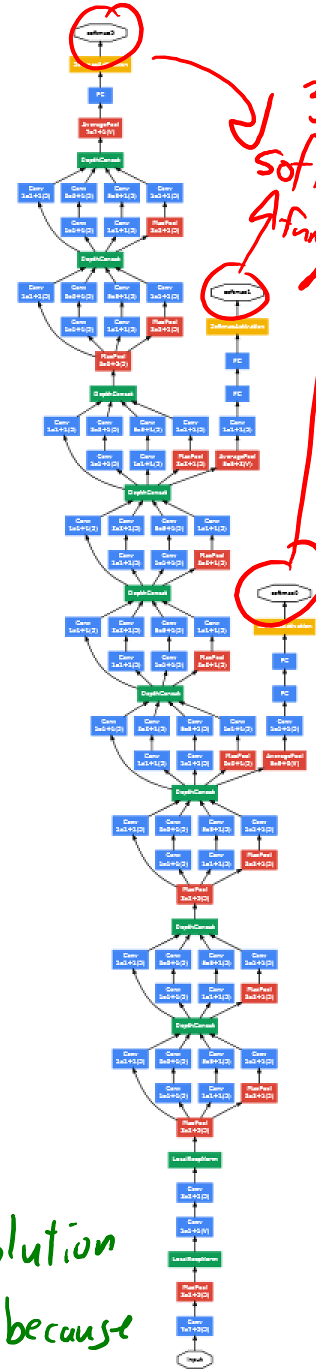
- Filters and stride got smaller over time.
  - Popular VGG approach uses 3x3 convolution layers with stride of 1.
    - 3x3 followed by 3x3 simulates a 5x5, and another 3x3 simulates a 7x7, and so on.
    - Speeds things up and reduces number of parameters.
    - Increases number of non-linear ReLU operations.



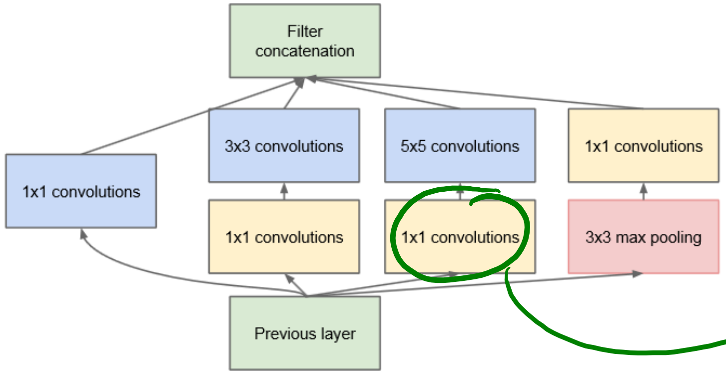
https://www.cs.toronto.edu/~frossard/post/vgg16/

ImageNet Insights

- Filters and stride got smaller over time.
  - Popular VGG approach uses 3x3 convolution layers with stride of 1.
  - GoogLeNet considered multiple filter sizes, but not as popular.
- Eventual switch to "fully-convolutional" networks.
  - No fully connected layers.



3 softmax functions



(b) Inception module with dimensionality reduction

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

ImageNet Insights

- Filters and stride got smaller over time.
  - Popular VGG approach uses 3x3 convolution layers with stride of 1.
  - GoogLeNet considered multiple filter sizes, but not as popular.
- Eventual switch to “fully-convolutional” networks.
  - No fully connected layers.
- ResNets allow easier training of deep networks.
  - Won all 5 tasks in 2015, training 152 layers for 2-3 weeks on 8 GPUs.
- Ensembles help.
  - Combine predictions of previous networks.

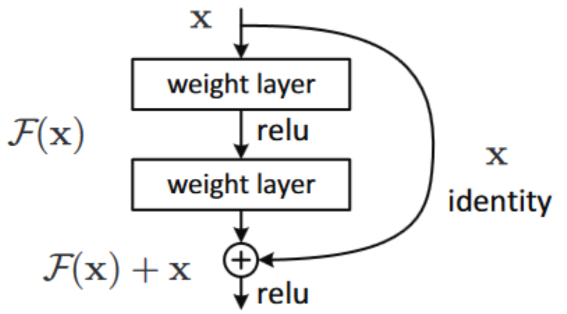
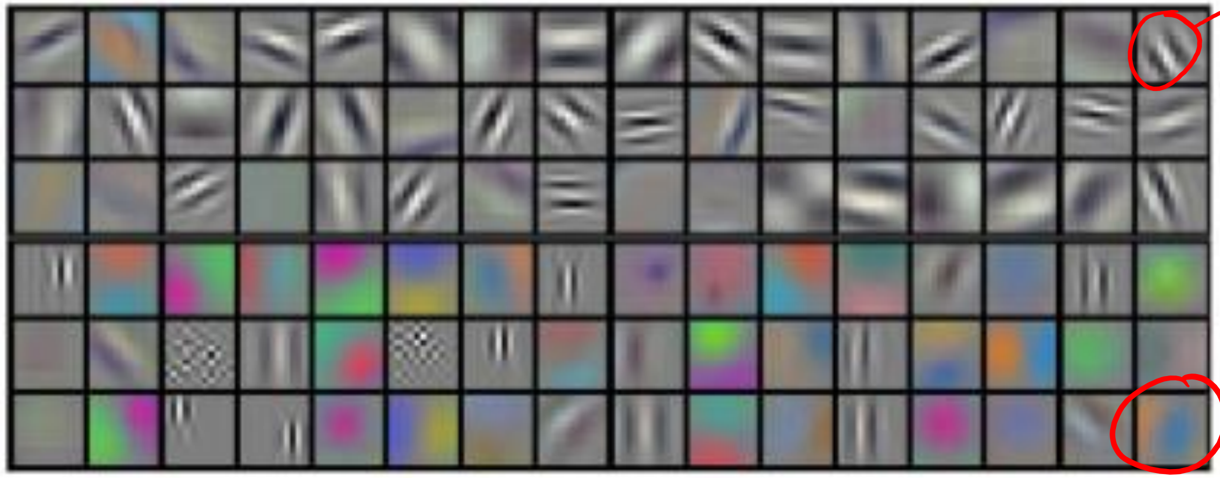


Figure 2. Residual learning: a building block.

Are CNNs learning something sensible?

- Filters learned by first layer of original AlexNet:



"Gabor" filters:
- Gaussian times sine or cosine.

"Opponent" colour coding.

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

- Note that non-orthogonal PCA gives similar results (but only 1 layer).

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

Are CNNs learning something sensible?

- It’s harder to visualize what is learned in other layers.
  – Deconvolution networks try to reconstruct what “activates” filters.

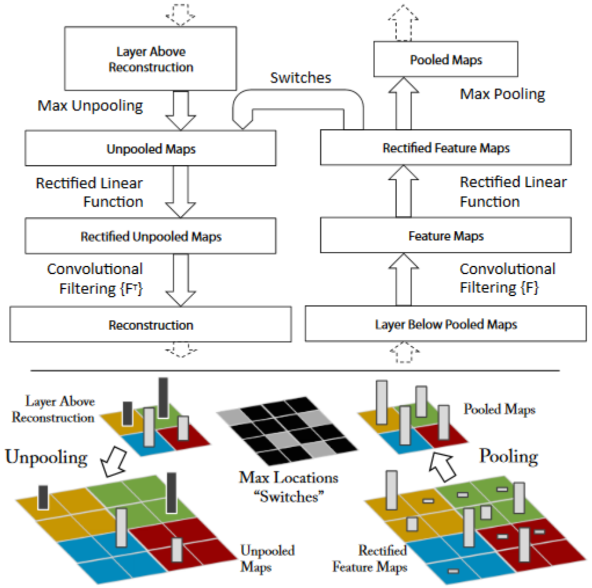
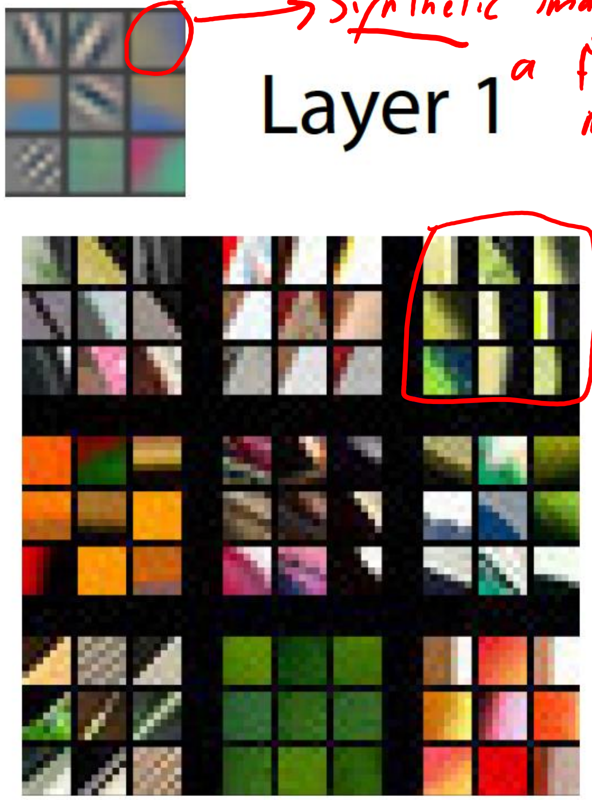


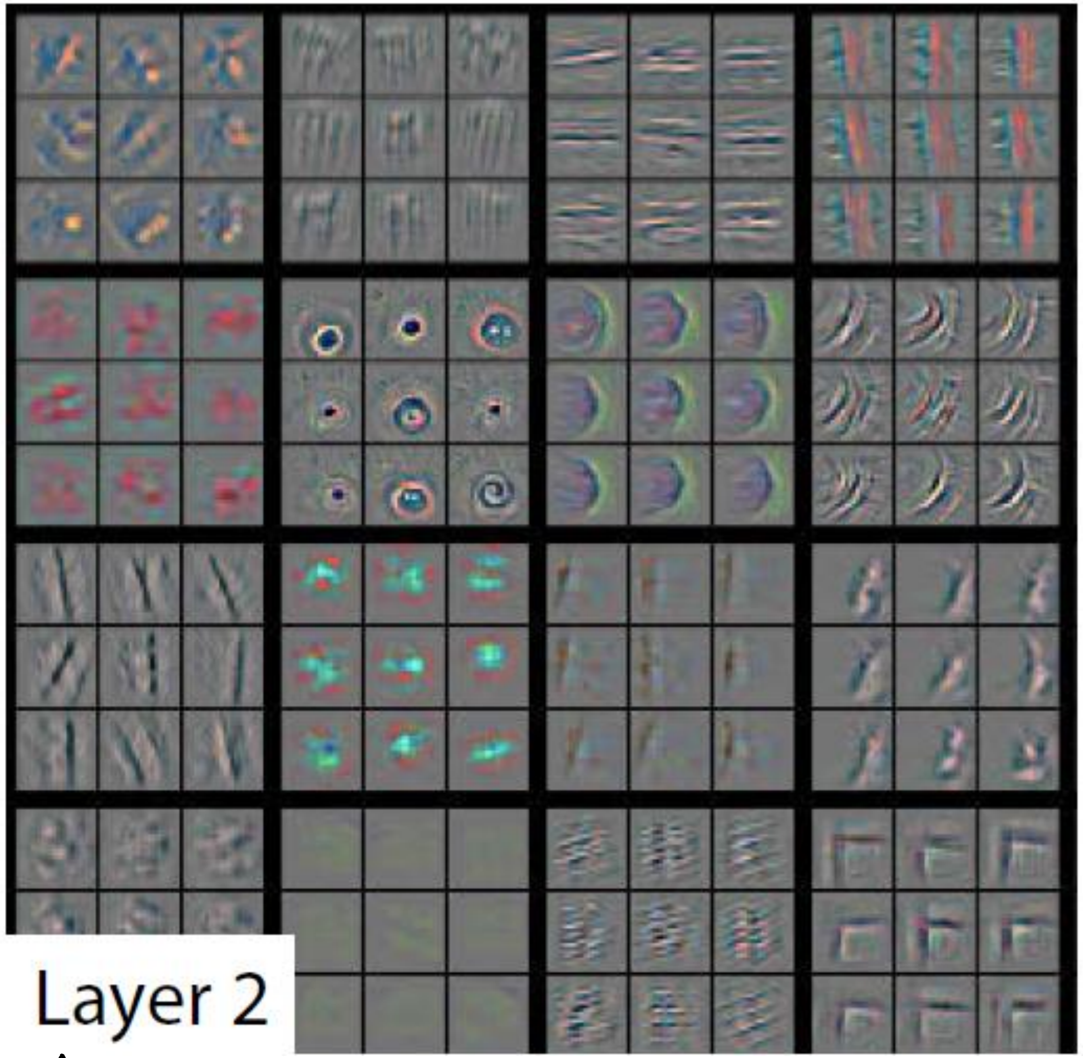
Figure 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local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

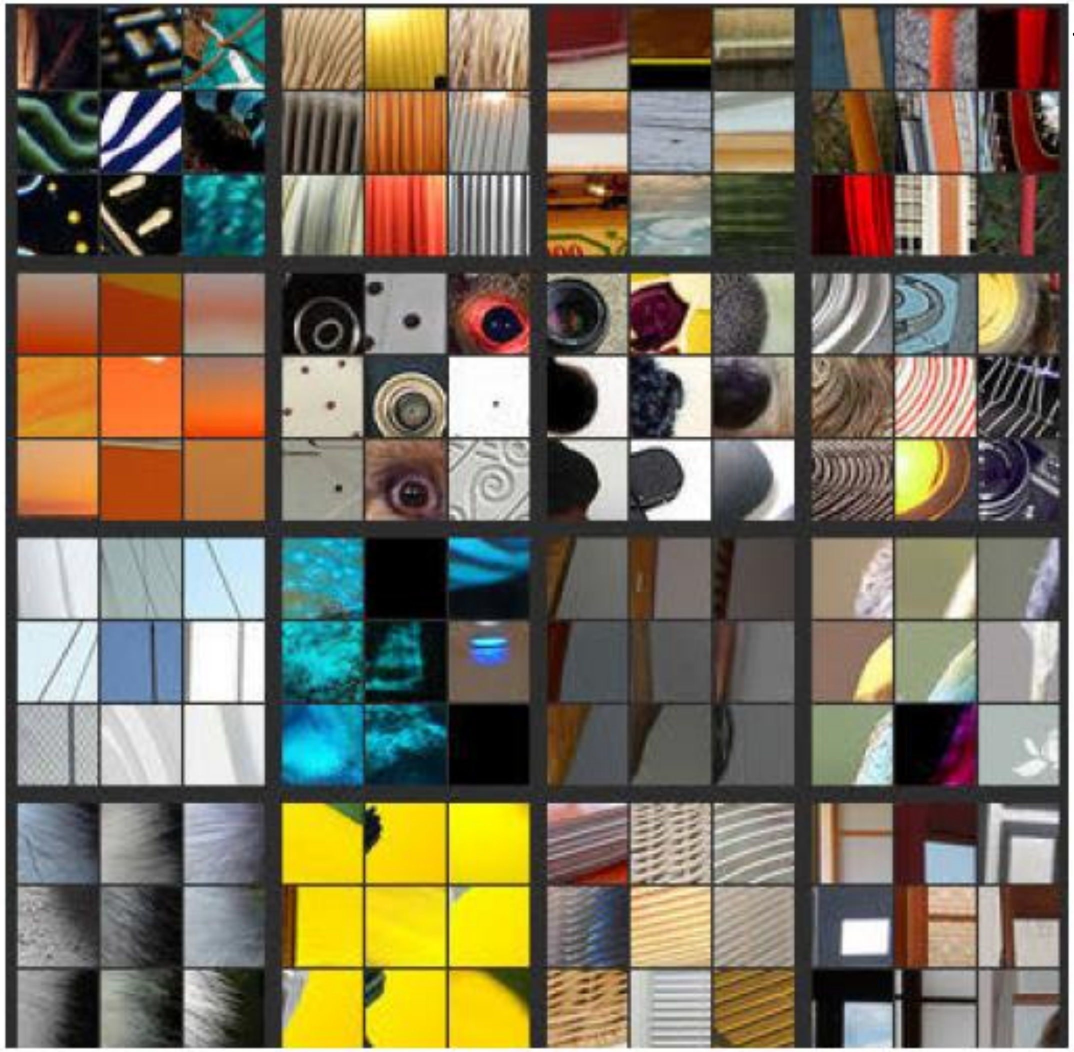


Synthetic image that makes a filter respond most strongly.

Real images where filter responds most strongly.

Are CNNs learning something sensible?



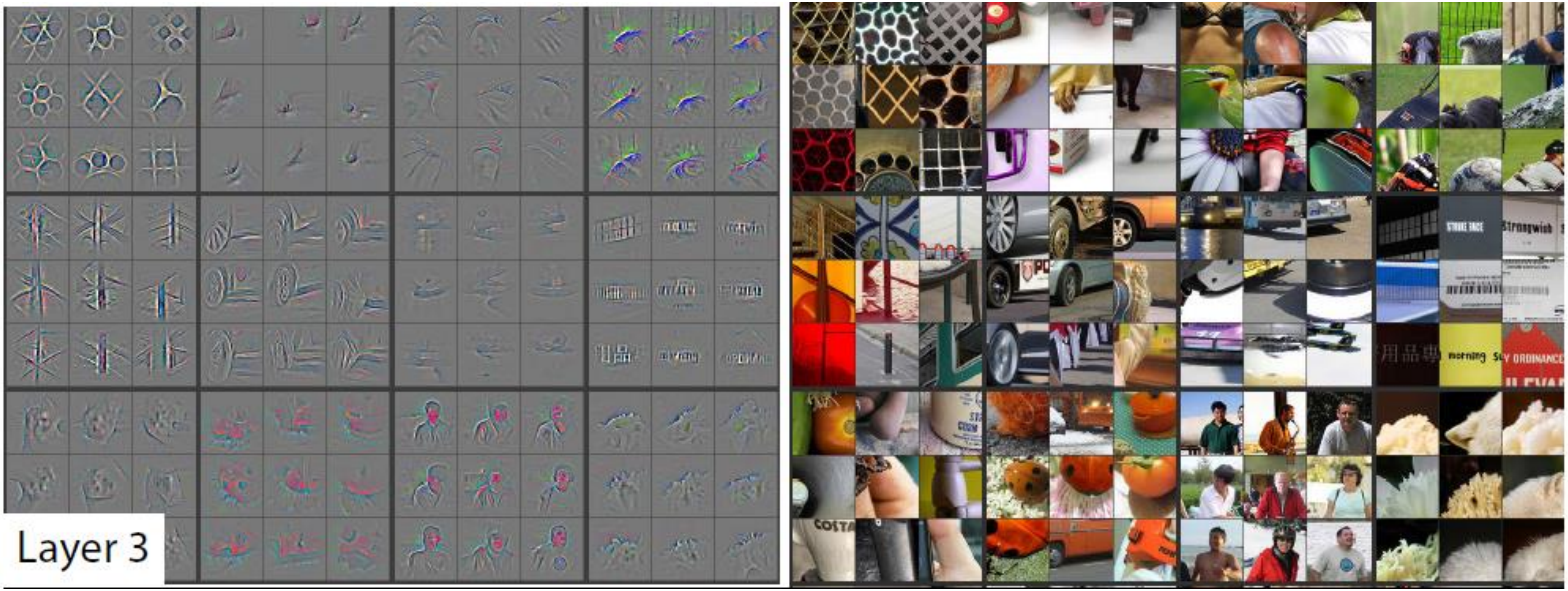


Patch from data giving largest response

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

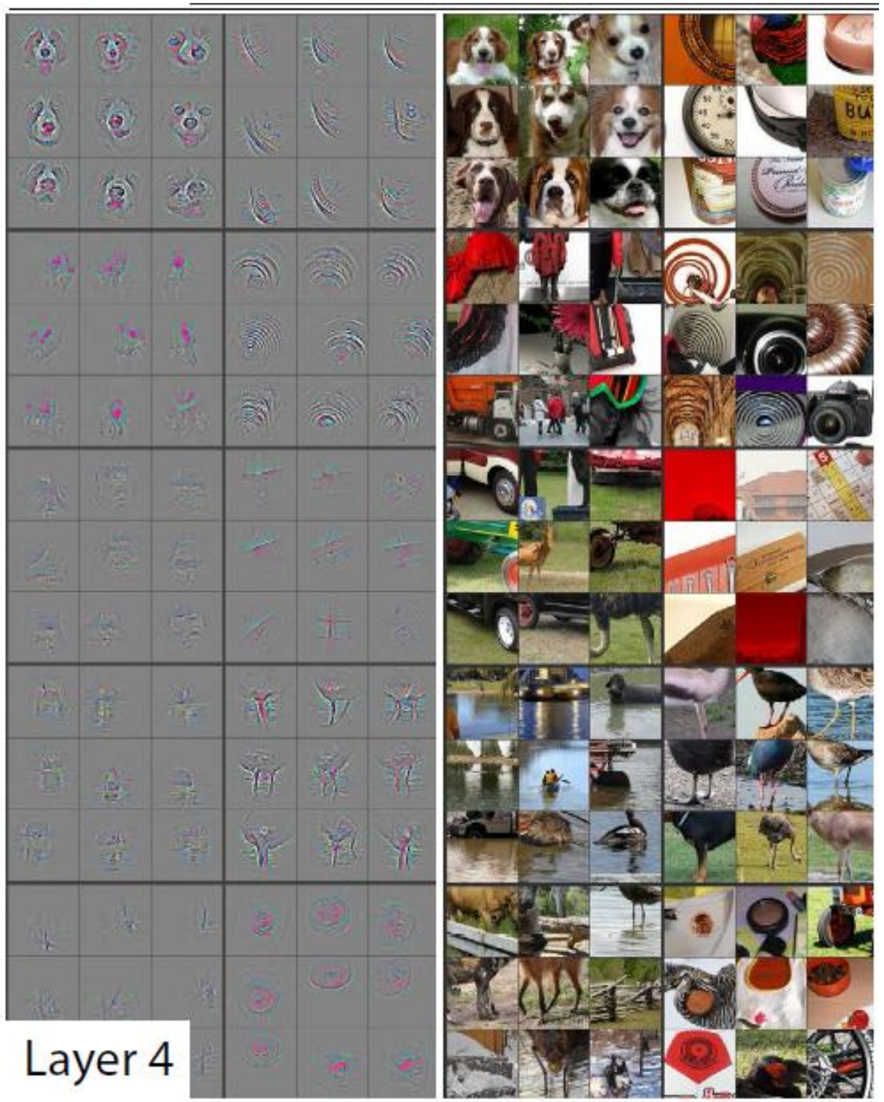
Deconvolution network giving patch that leads to largest response

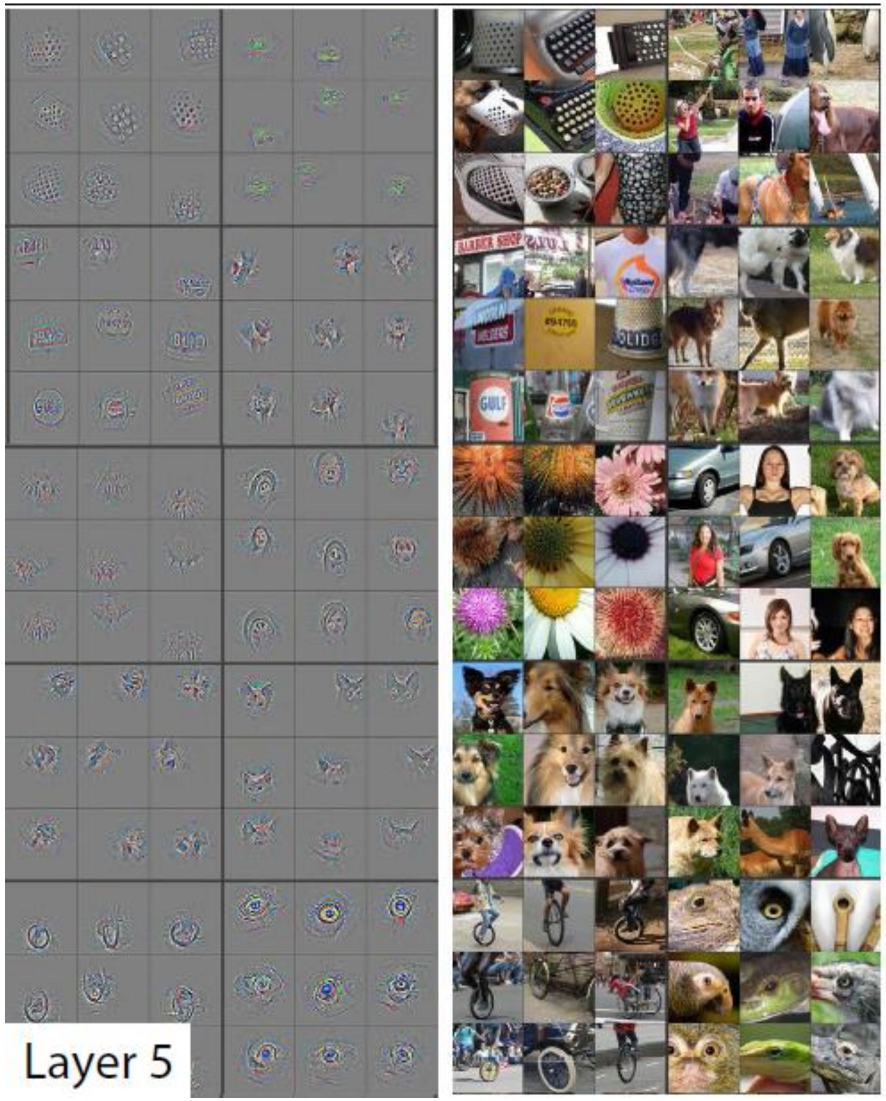
Are CNNs learning something sensible?



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

Are CNNs learning something sensible?

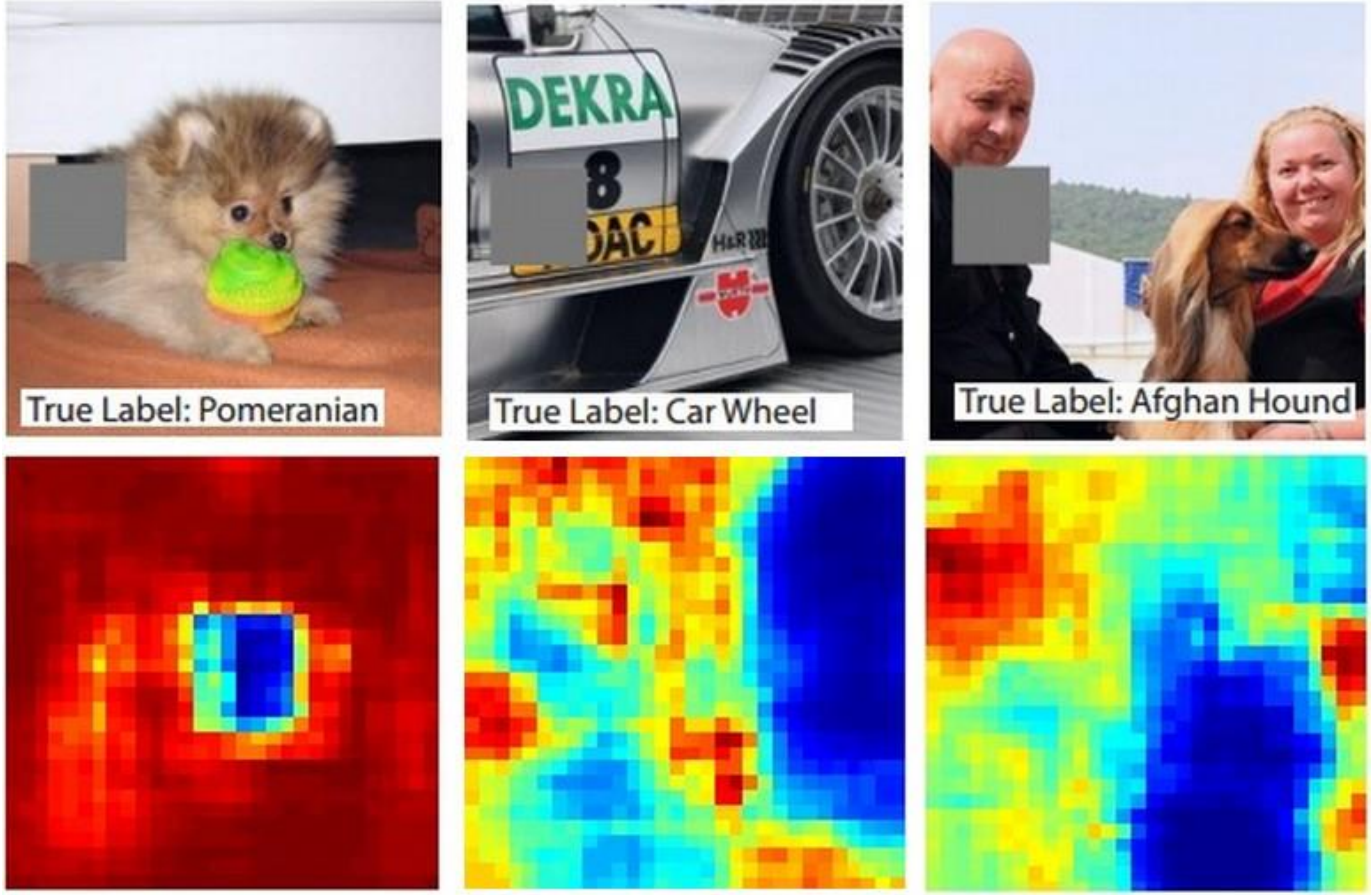




https://arxiv.org/pdf/1311.2901v3.pdf

Are CNNs learning something sensible?

- We can look at how prediction changes if we hide part of image:



http://cs231n.github.io/understanding-cnn/

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.
    - Recent work showed accuracy drop of 4-10% by using a different test set on CIFAR 10.

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

Mission Accomplished?

- We're still missing a lot of theory and understanding deep learning.

From: Boris
To: Ali

On Friday, someone on another team changed the default rounding mode of some Tensorflow internals (from truncation to "round to even").*

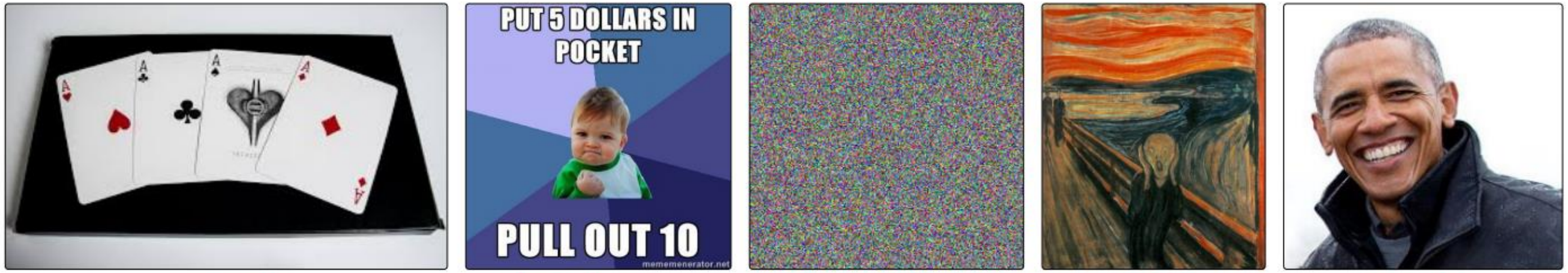
*Our training broke. Our error rate went from <25% error to ~99.97% error (on a standard 0-1 binary loss).

- "[Good CS expert says: Most firms that thinks they want advanced AI/ML really just need linear regression on cleaned-up data]."

http://www.argmin.net/2017/12/05/kitchen-sinks/

Mission Accomplished?

- Despite high-level of abstraction, deep CNNs are easily fooled:
  - Hot research topic at the moment.

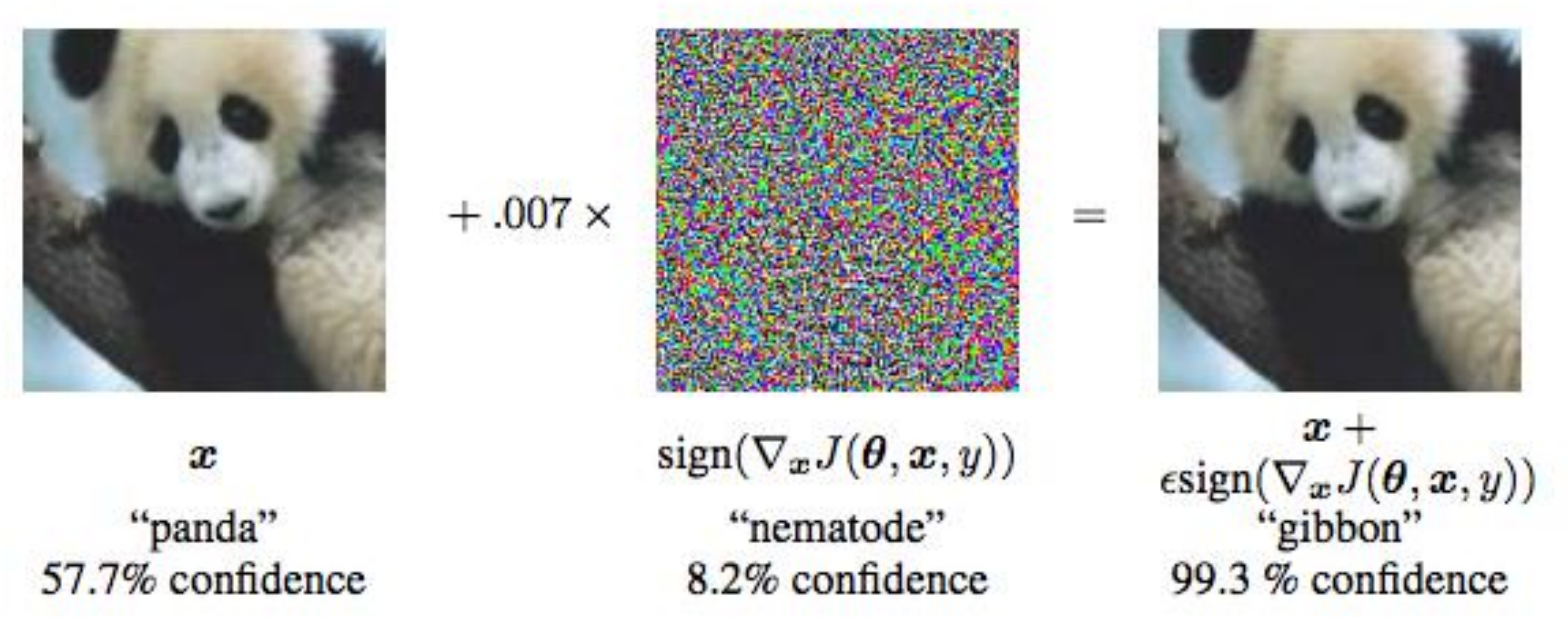


DenseNet 161 (2017)	Envelope 31%	Balance Beam 52%	Chainlink Fence 31%	Chest 37%	Tench 36%
SqueezeNet (2016)	Binder 43%	Balance Beam 18%	Poncho 32%	Jean 30%	Suit 21%
ResNet 152 (2015)	Envelope 40%	Pacifier 33%	Chain Mail 29%	Dust Cover 52%	Sweatshirt 25%
VGG 19 (2014)	Binder 51%	Dust Cover 44%	Window Screen 5%	Chest 11%	Sweatshirt 46%
AlexNet (2012)	T-shirt 16%	Dust Cover 22%	Cardigan 12%	Theater Curtain 3%	Coho 37%

Figure 1: The arbitrary predictions of several popular networks [2, 3, 4, 5, 6] that are trained on ImageNet [1] on unseen data. The red predictions are entirely wrong, the green predictions are justifiable, the orange predictions are less justifiable. The middle image is noise sampled from $\mathcal{N}(\mu = 0.5, \sigma = 0.25)$ without any modifications. This unpredictable behaviour is not limited to demonstrated architectures. We show that merely thresholding the output probability is not a reliable method to detect these problematic instances.

Mission Accomplished?

- Despite high-level of abstraction, deep CNNs are easily fooled:
  - Hot research topic at the moment.
- Recent work: imperceptible noise that changes the predicted label



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

Mission Accomplished?

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

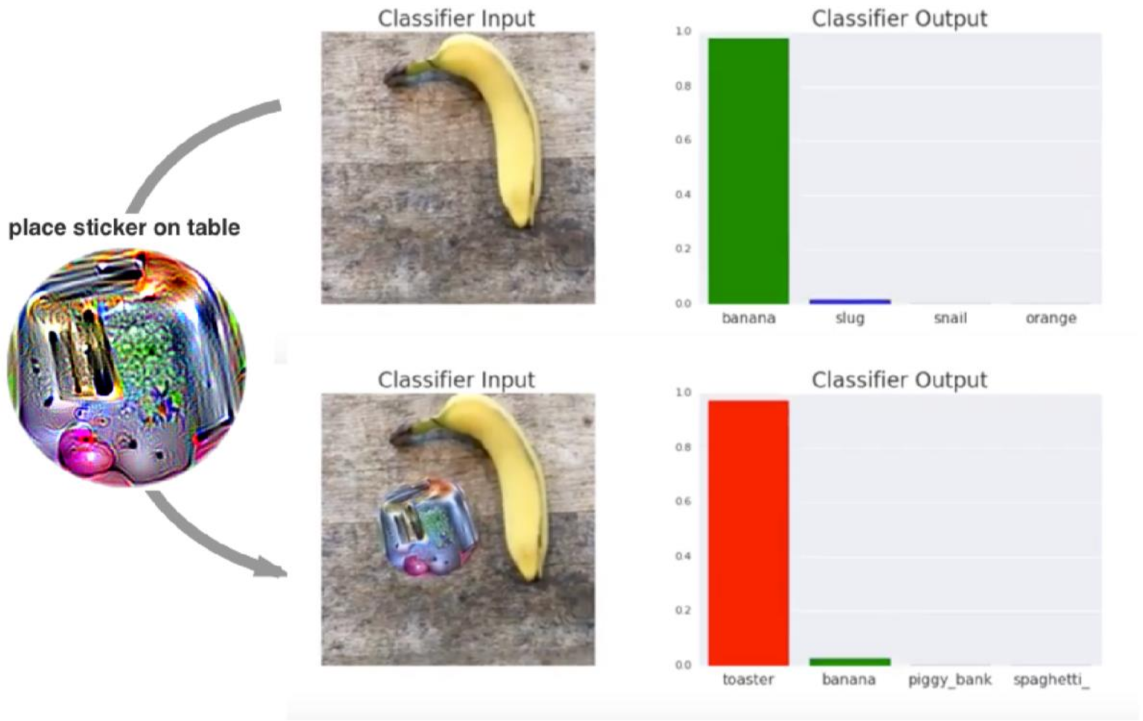


Figure 1: A real-world attack on VGG16, using a physical patch generated by the white-box ensemble method described in Section 3. When a photo of a tabletop with a banana and a notebook (top photograph) is passed through VGG16, the network reports class 'banana' with 97% confidence (top plot). If we physically place a sticker targeted to the class "toaster" on the table (bottom photograph), the photograph is classified as a toaster with 99% confidence (bottom plot). See the following video for a full demonstration: `https://youtu.be/i1sp4X57TL4`

https://arxiv.org/pdf/1712.09665.pdf

Mission Accomplished?

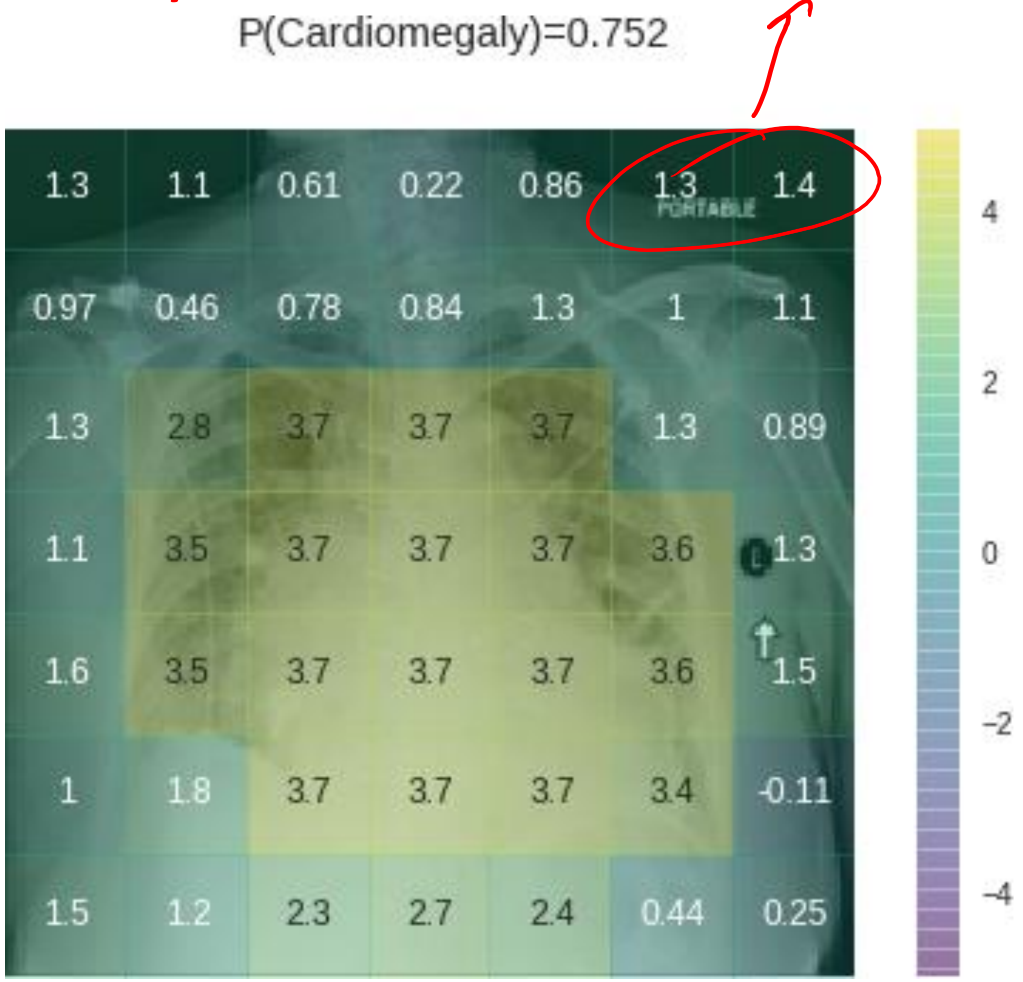
- Are the networks understanding the fundamental concepts?
  - Is being “surrounded by green” part of the definition of cow?
  - Do we need to have examples of cows in different environments?
    - Kids don’t need this.

https://www.onegreenplanet.org/news/cows-enjoy-the-beach/

Mission Accomplished?

- CNNs may not be learning what you think they are.
  - CNN for diagnosing enlarged heart:
    - Higher values mean more likely to be enlarged:
  - CNN says "portable" protocal is predictive:
    - But they are probabaly getting a "portable" scan because they're too sick to go the hospital.
  - CNN was biased by the scanning protocal.
    - Learns the scans that more-sick patients get.
    - This is not what we want in a medical test.

????



https://medium.com/@jrzech/what-are-radiological-deep-learning-models-actually-learning-f97a546c5b98

(Racially-)Biased Algorithms?

- Major issue: are we learning biased representations?
  - Biases could come from data (if data only has certain groups in certain situations).
  - Biases could come from labels (always using label of “ball” for certain sports).
  - Biases could come from learning method (model predicts “basketball” for black people more often than they appear in training data for basketball images).

Fig. 8: Pairs of pictures (columns) sampled over the Internet along with their prediction by a ResNet-101.

  - This is a major problem/issue when deploying these systems.
    - E.g., “repeat-offender prediction” that reinforces racial biases in arrest patterns.

https://arxiv.org/pdf/1711.11443.pdf

(pause)

Ensemble Methods

- Ensemble methods are classifiers that have classifiers as input.
  - Also called “meta-learning”.
- They have the best names:
  - Averaging.
  - Boosting.
  - Bootstrapping.
  - Bagging.
  - Cascading.
  - Random Forests.
  - Stacking.
- Ensemble methods often have higher accuracy than input classifiers.

Ensemble Methods

- Remember the fundamental trade-off:
  1. E_{train}: How small you can make the training error.

     vs.

  2. E_{approx}: How well training error approximates the test error.
- Goal of ensemble methods is that meta-classifier:
  – Does much better on one of these than individual classifiers.
  – Doesn't do too much worse on the other.
- This suggests two types of ensemble methods:
  1. Averaging: improves approximation error of classifiers with high E_{approx}.
  2. Boosting: improves training error of classifiers with high E_{train}.

AdaBoost: Classic Boosting Algorithm

- A classic boosting algorithm is AdaBoost.

- AdaBoost assumes we have a "base" binary classifier that:
  - Is simple enough that it doesn't overfit much.
  - Can obtain >50% weighted accuracy on any dataset.

$$\frac{1}{n}\sum_{i=1}^{n} v_i I[\hat{y}_i = y_i]$$

is example 'i' classified correctly

weights (sum to 1)

- Example: decision stumps or low-depth decision trees.
  - Easy to modify stumps/trees to use weighted accuracy as score.

AdaBoost: Classic Boosting Algorithm

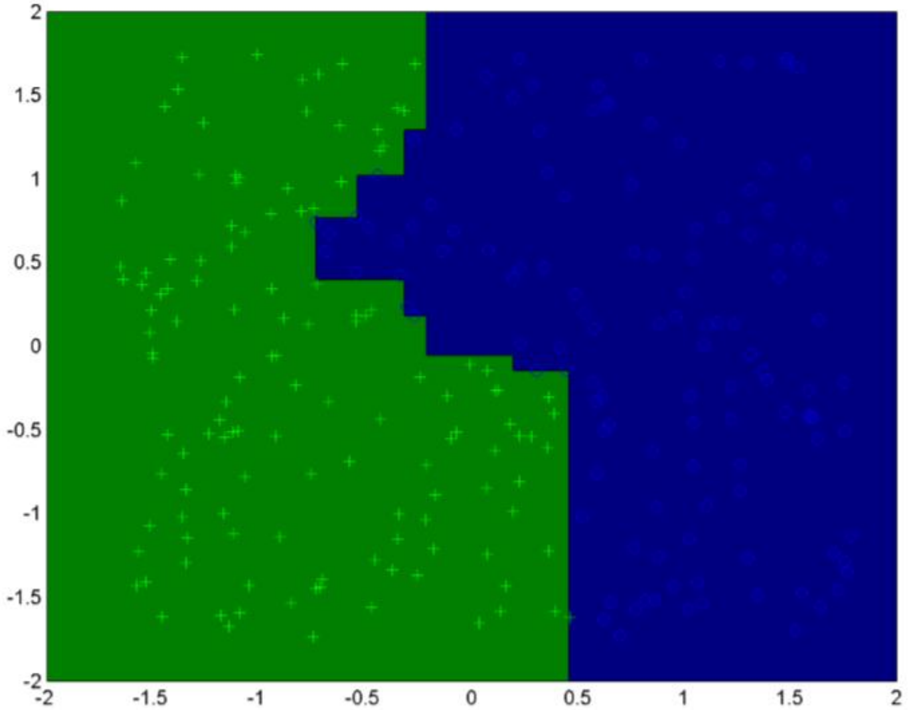
- Overview of AdaBoost:
  1. Fit a classifier on the training data.
  2. Give a higher weight to examples that the classifier got wrong.
     - Weight gets exponentially larger if you are wrong, smaller if you are right.
  3. Fit a classifier on the weighted training data.
  4. Go back to 2.

- Final prediction: weighted vote of individual classifier predictions.
  – Trees with higher (weighted) accuracy get higher weight.

- See Wikipedia for precise definitions of weights.

AdaBoost with Decision Stumps

- 2D example of AdaBoost with decision stumps (with accuracy score):
  - 100% training accuracy.
  - Ensemble of 50 decision stumps.
    - Fit sequentially, not independently.



- Are decision stumps a good base classifier?
  - They tend not to overfit.
  - Easy to get >50% weighted accuracy.

- Base classifiers that don't work:
  - Deep decision trees (no errors to "boost").
  - Decision stumps with infogain (doesn't guarantee >50% weighted accuracy).
  - Weighted logistic regression (doesn't guarantee >50% weighted accuracy).

AdaBoost Discussion

- AdaBoost with shallow decision trees gives fast/accurate classifiers.
  - Classically viewed as one of the best “off the shelf” classifiers.
  - Procedure originally came from ideas in learning theory.

- Many attempts to extend theory beyond binary case.
  - Led to “gradient boosting”, which is like “gradient descent with trees”.
  - Modern methods look like AdaBoost,
    but don’t necessarily have it as a special case.

XGBoost: Modern Boosting Algorithm

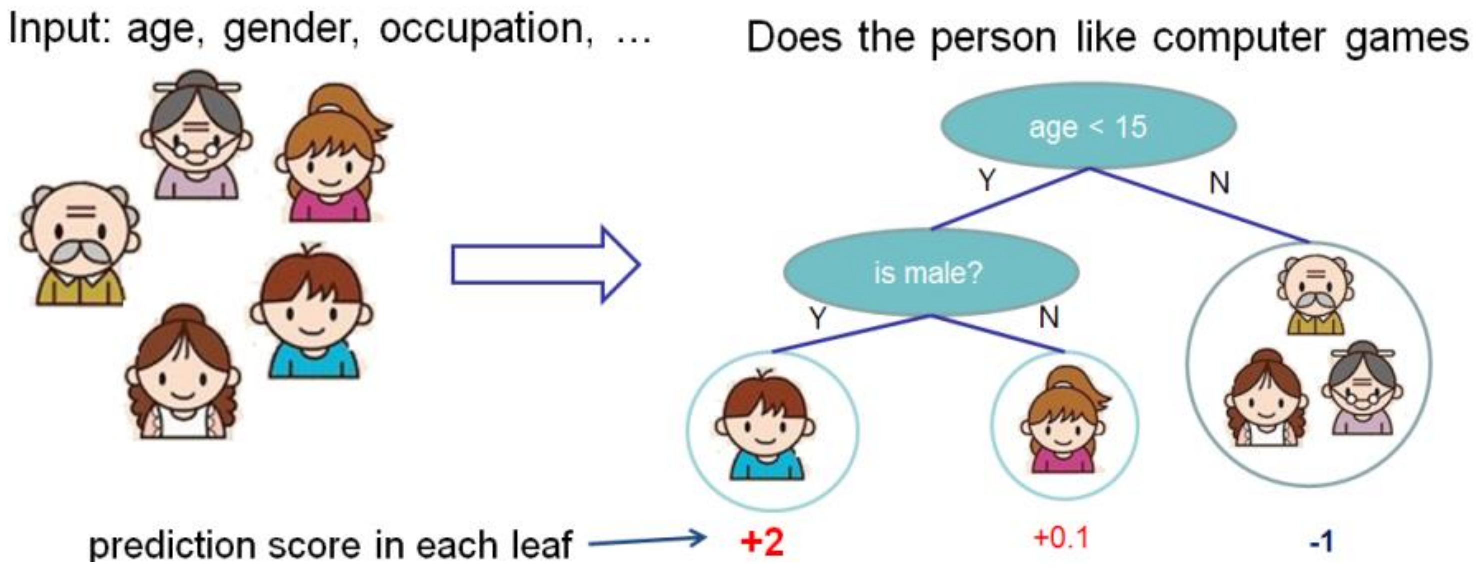
- Boosting has seen a recent resurgence, partially due to XGBoost:
  - A boosting implementation that allows huge datasets.
  - Has been part of many recent winners of Kaggle competitions.

- As base classifier, XGBoost uses regularized regression trees.

https://xgboost.readthedocs.io/en/latest/tutorials/model.html

Regularized Regression Trees

- Regression trees used in XGBoost:
  - Each split is based on 1 feature.
  - Each leaf 'L' gives a real-valued score w_L (think of this as being like $w^T x_i$).



  - Uses L0-regularization of scores w_L (which leads to pruning).
  - Also uses L2-regularization of scores w_L (avoids overfitting).

https://xgboost.readthedocs.io/en/latest/tutorials/model.html

XGBoost Tree-Fitting Procedure

- Sequence of boosting predictions:
  - Fix prediction of previous (t-1) trees.
  - Additively modify prediction with tree ‘t’.

$$
\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}
$$

- Greedily grow trees so scores minimize regularized squared error:

$$
\text{obj}^{(t)} = \sum_{i=1}^{n} (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^{t} \Omega(f_i)
$$

old prediction

score from tree

L_0- and L_2- regularization of scores (2 hyper-parameters)

  - Has simple closed-form solution for any split.
  - Same speed as fitting decision trees from Week 2 of 340.

XGBoost Discussion

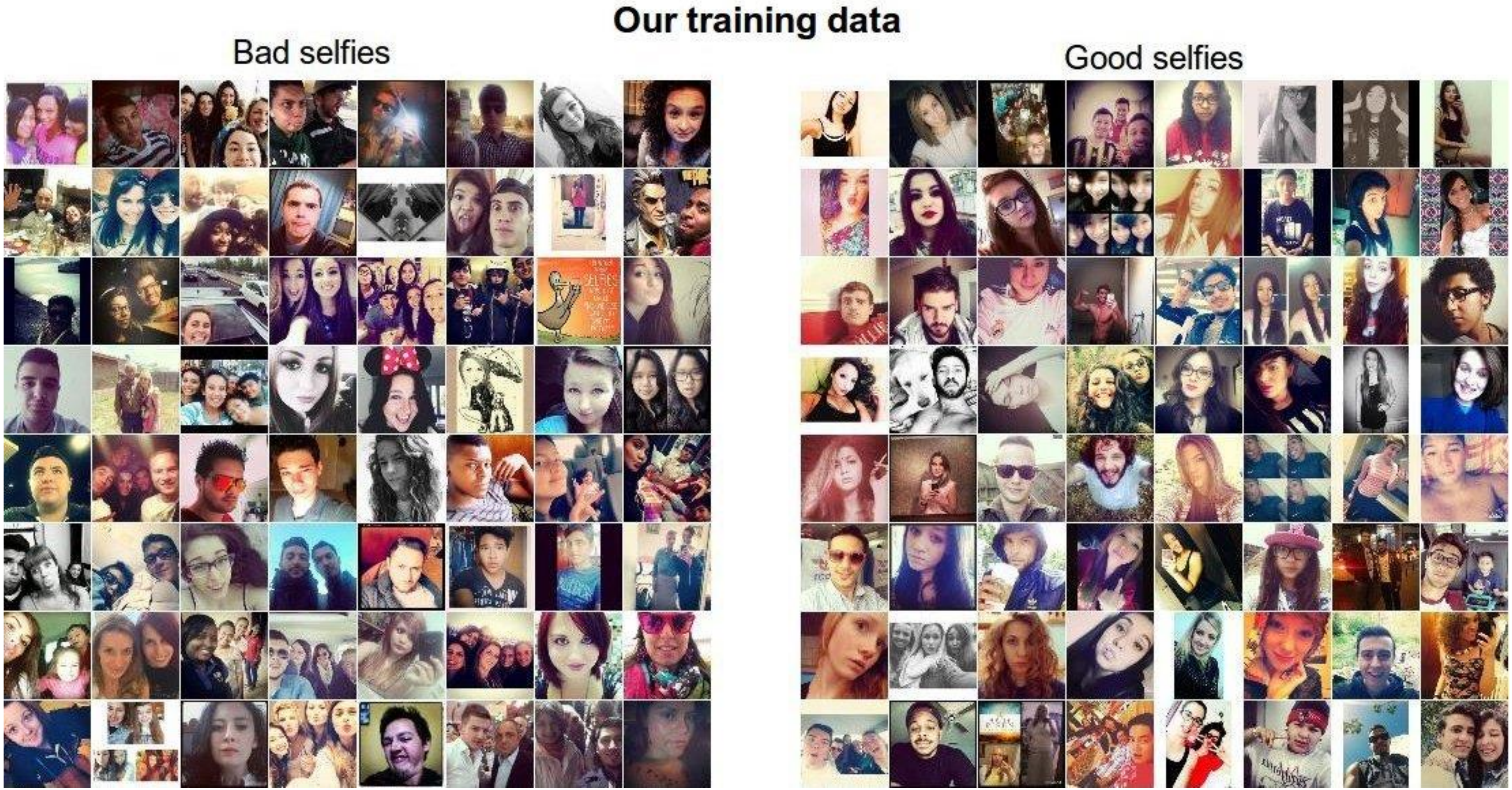
- Cost of fitting trees in XGBoost is same as usual decision tree cost.
  - Can't be done in parallel like random forest, since fitting trees sequentially.
  - XGBoost includes a lot of tricks to make this efficient.

- Rather than pruning the trees if score doesn't improve, grows full trees.
  - And then prunes parts that aren't increasing the score.

- How do you maintain efficiency if not using squared error?
  - For non-quadratic losses like logistic, there is no closed-form solution.
  - XGBoost approximates non-quadratic losses with second-order Taylor expansion.
    - Maintains efficiency of least squares case for other losses.

(pause)

CNNs for Rating Selfies

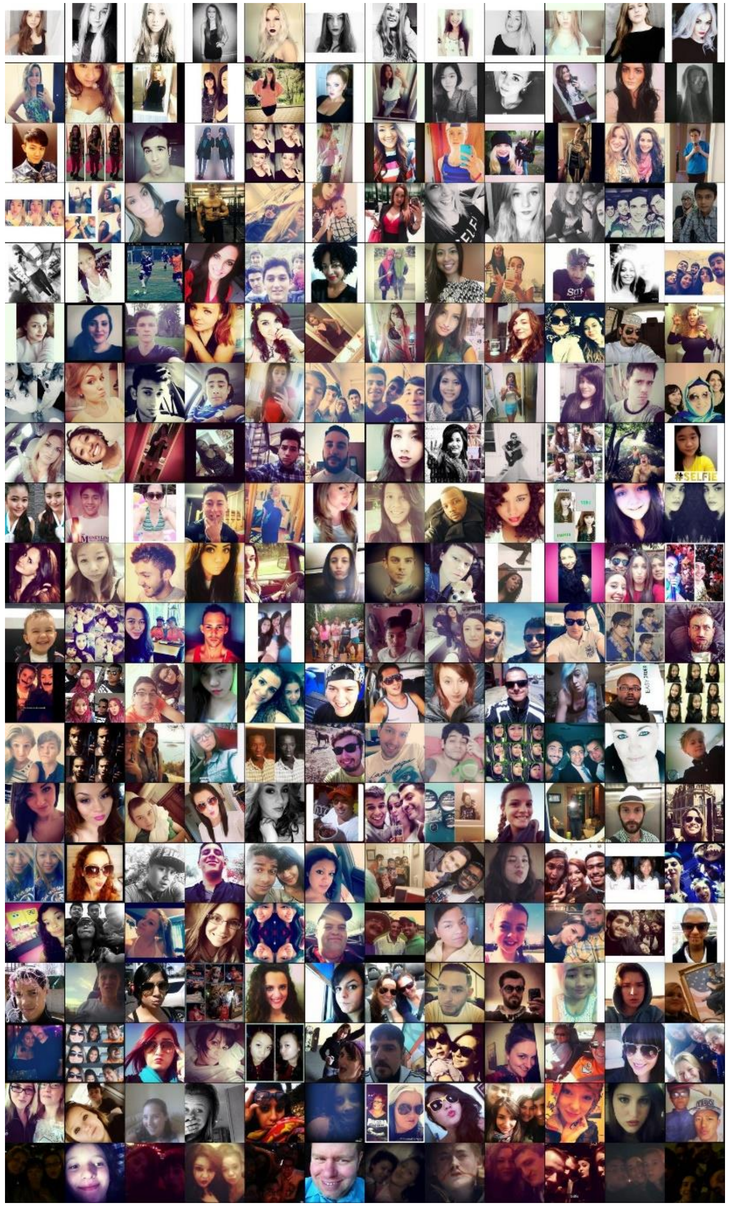


https://karpathy.github.io/2015/10/25/selfie/

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



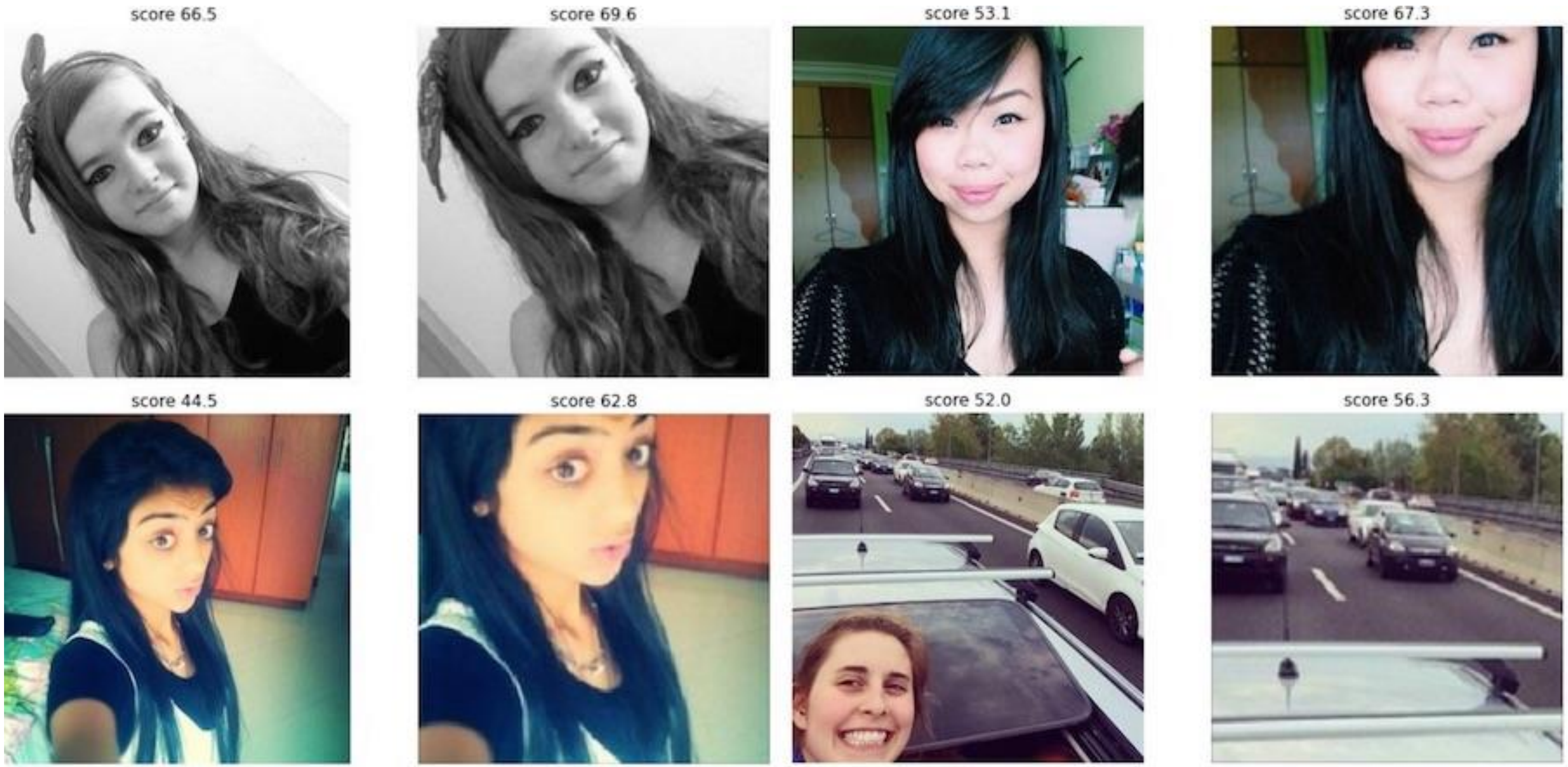
Don't:

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

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

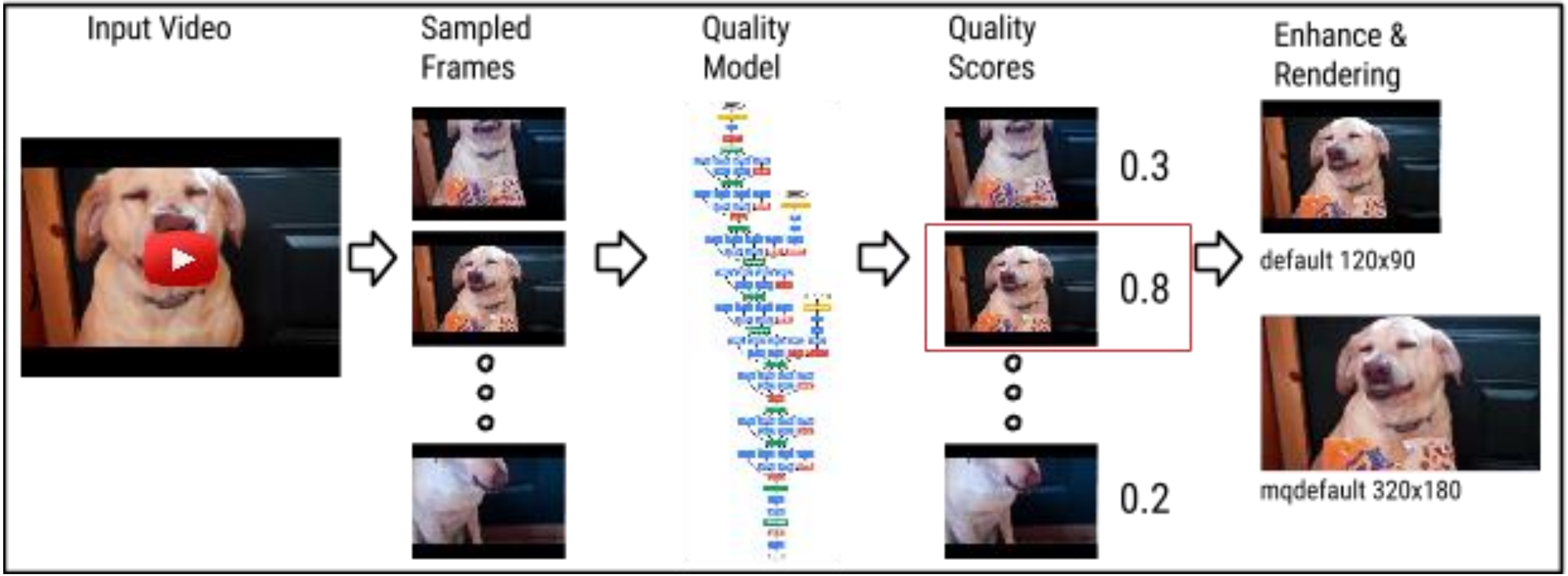
CNNs for Rating Selfies

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:

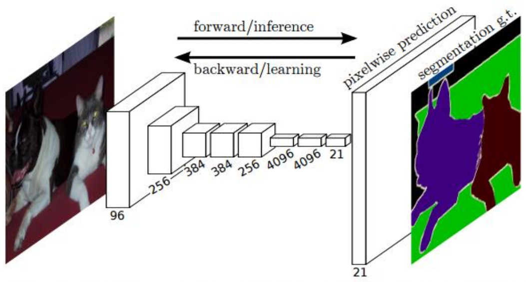


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

- Image segmentation:

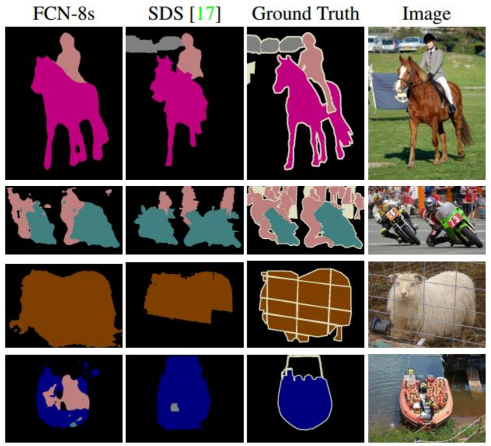
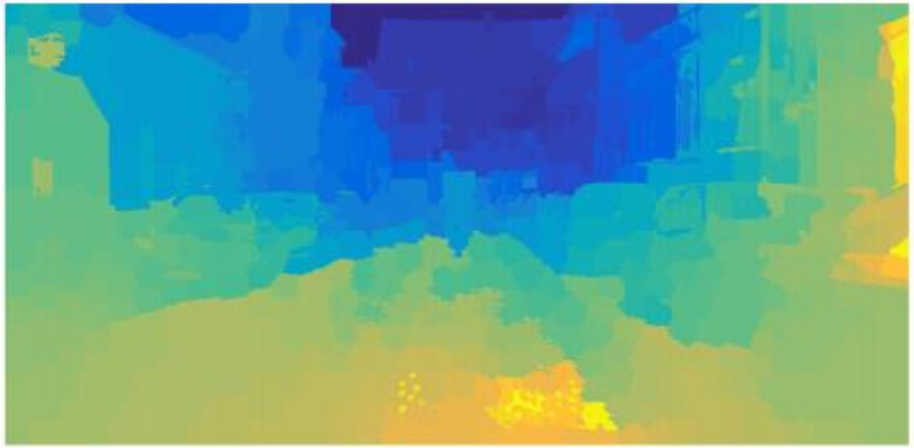


Figure 6. Fully convolutional segmentation nets produce state-of-the-art performance on PASCAL. The left column shows the output of our highest performing net, FCN-8s. The second shows the segmentations produced by the previous state-of-the-art system by Hariharan *et al.* [17]. Notice the fine structures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

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

Beyond Classification (CPSC 540)

- Depth Estimation:



- "A Year in Computer Vision"

Beyond Classification (CPSC 540)

- Image colorization:

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

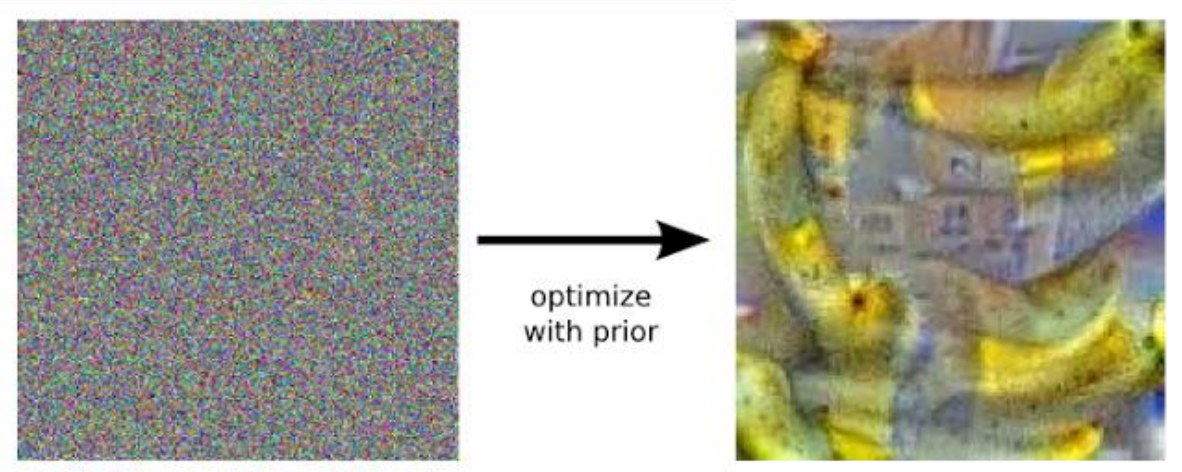
  – Image Gallery, Video

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

Inceptionism

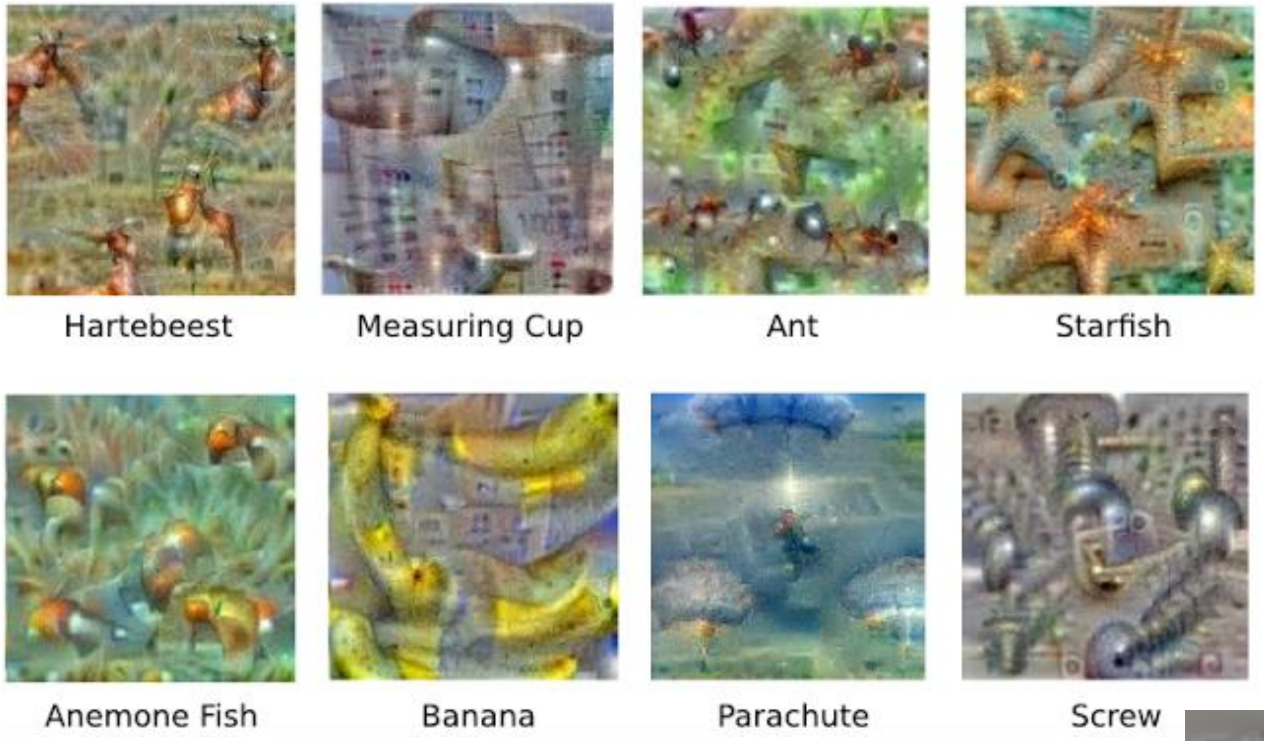
- A crazy idea:
  - Instead of weights, use backpropagation to take gradient with respect to x_i.
- Inceptionism with trained network:
  - Fix the label y_i (e.g., "banana").
  - Start with random noise image x_i.
  - Use gradient descent on image x_i.
  - Add a spatial regularizer on x_{ij}:
    - Encourages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

"Show what you think a banana looks like"



Inceptionism

- Inceptionism for different class labels:

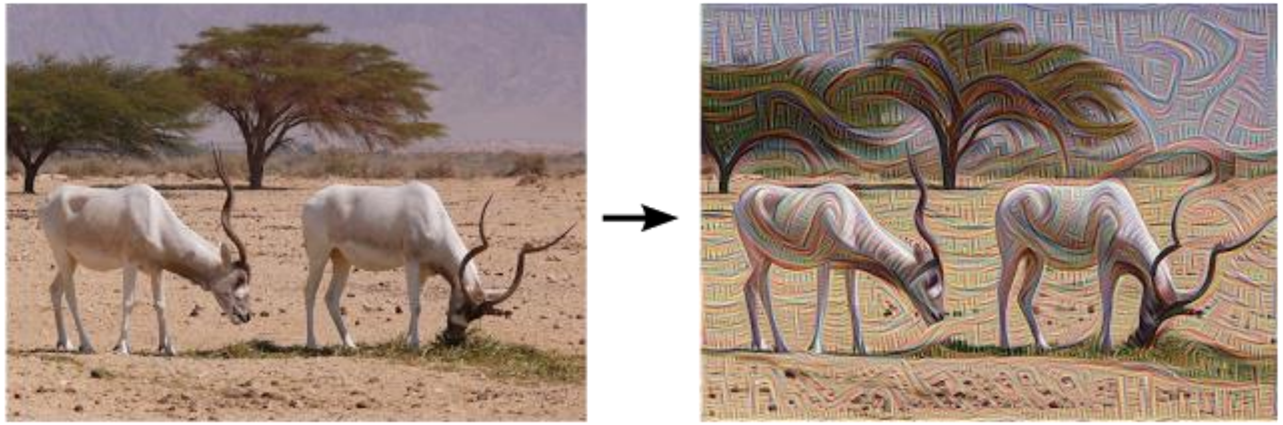


Dumbbell

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

Inceptionism

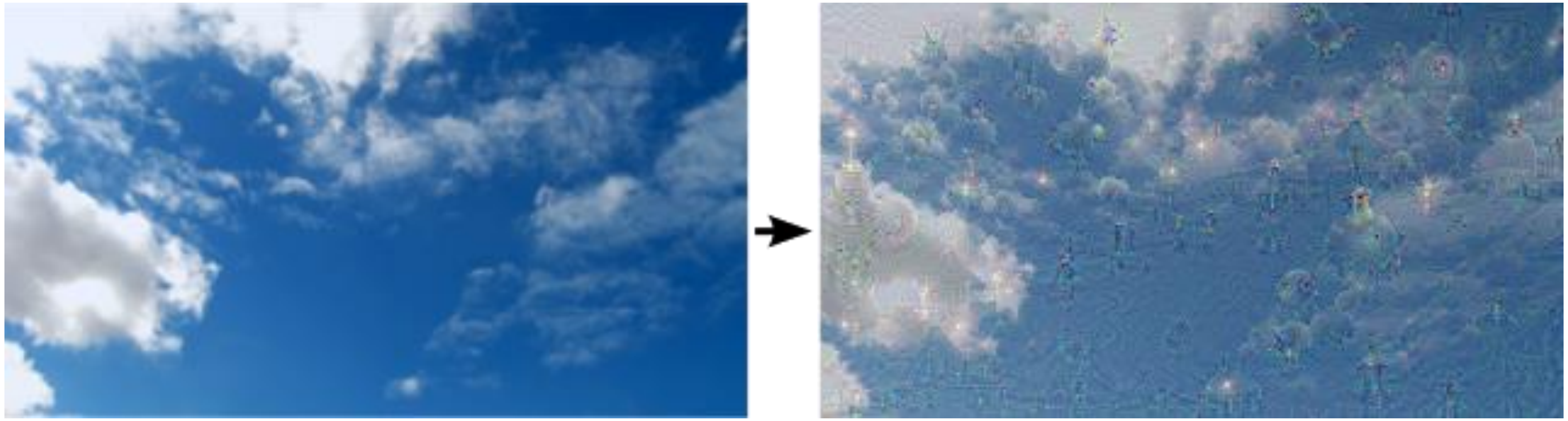
- Inceptionism where we try to match $z_i^{(m)}$ values instead of y_i.
  - Shallow 'm':



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:

  - 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'.

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

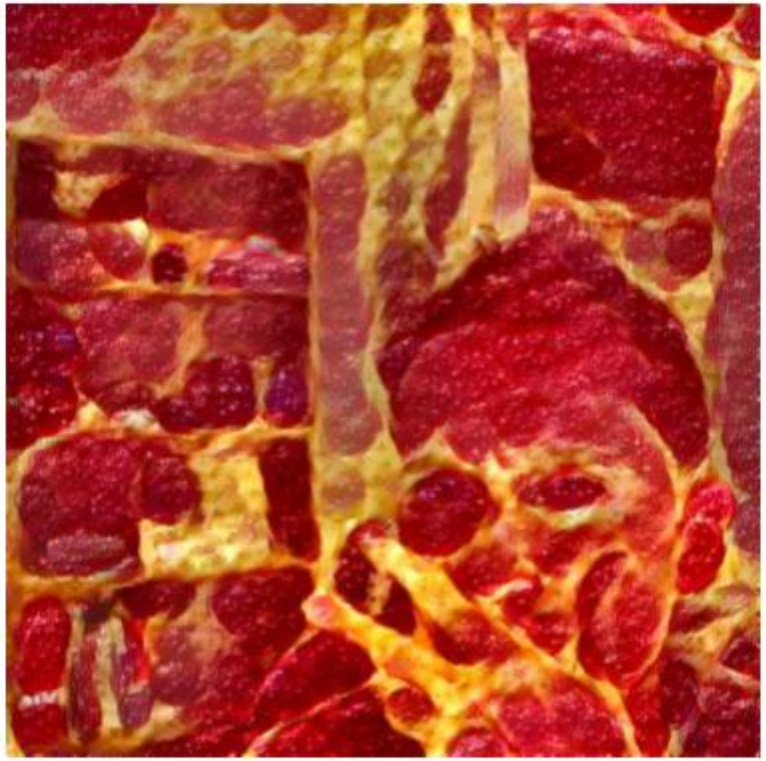


Figure: **Left:** My friend Grant, **Right:** Grant as a pizza

Artistic Style Transfer

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

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

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

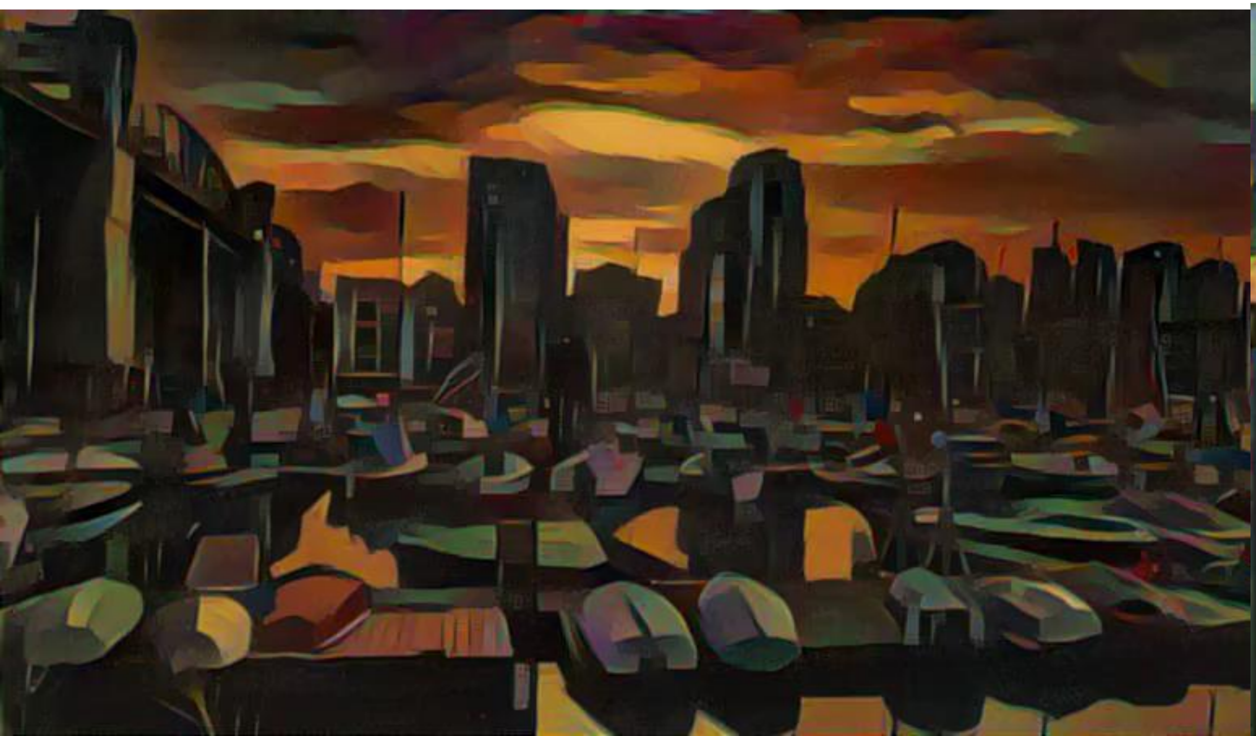
Artistic Style Transfer

- 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 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:
  - 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 532L:
  - Deep learning for vision, sound, and language.
- CPSC 532W:
  - Probabilistic programming.
- EECE 592:
  - Deep learning and reinforcement learning.
- 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.
- EOSC 510:
  - Similar/complementary topics, emphasis on EOSC applications.
- Courses by Muhammad Abdul-Majeed (text data) or Eldad Haber (optimization).

Final Slide

- “Calling Bullshit in the Age of Big Data”:
  - https://www.youtube.com/playlist?list=PLPnZfvKID1Sje5jWxt-4CSZD7bUI4gSPS
  - Every “data scientist“ should watch all these lectures.
  - You should be able to recognize non-sense,
    and not accidently produce non-sense!

- Thank you for your patience.
  - This was our first multi-section offering.

- Good luck with the next steps!