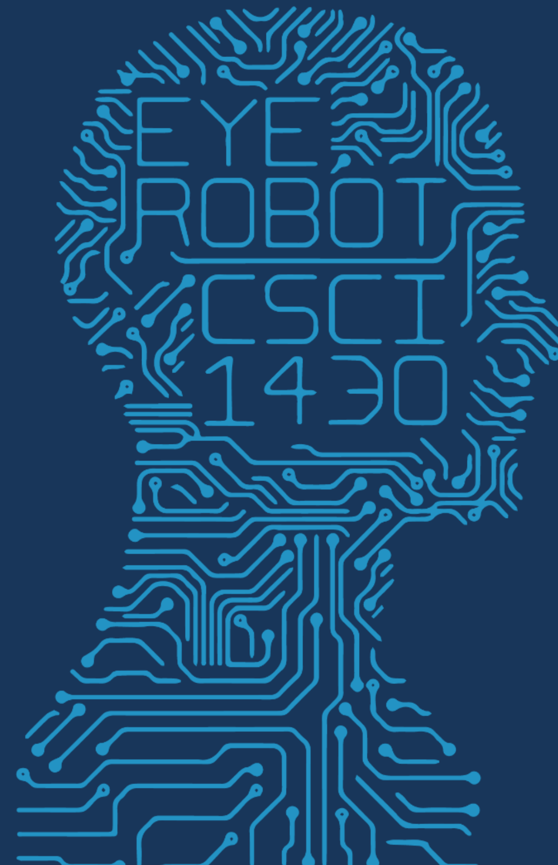
1950

Future Vision



2017 MWF 1pm 368

Computer Vision

Predicting Financial Crime: Augmenting the Predictive Policing Arsenal

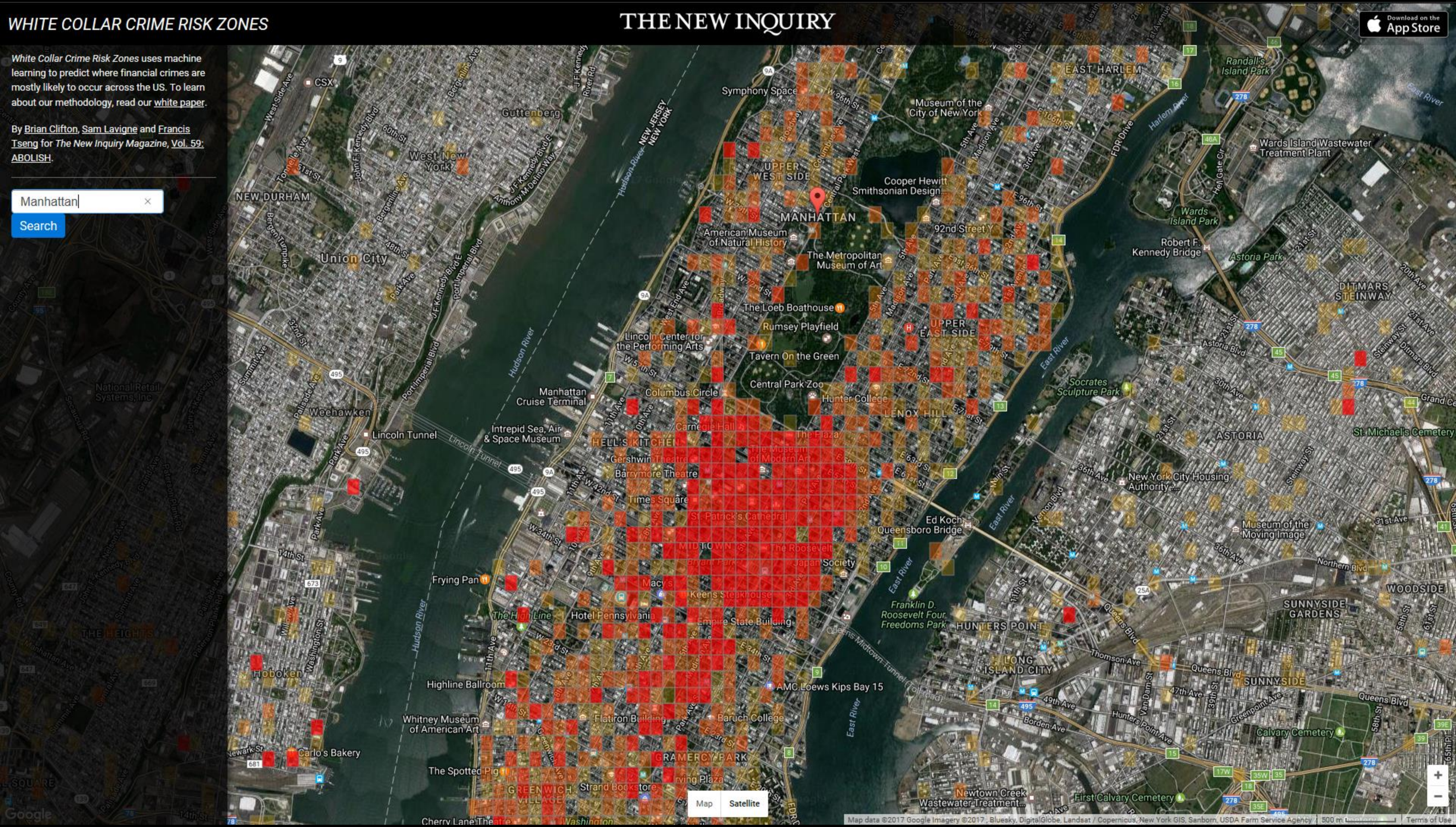
Brian Clifton[1], Sam Lavigne[1], and Francis Tseng[1]

[1] The New Inquiry
`https://thenewinquiry.com/`

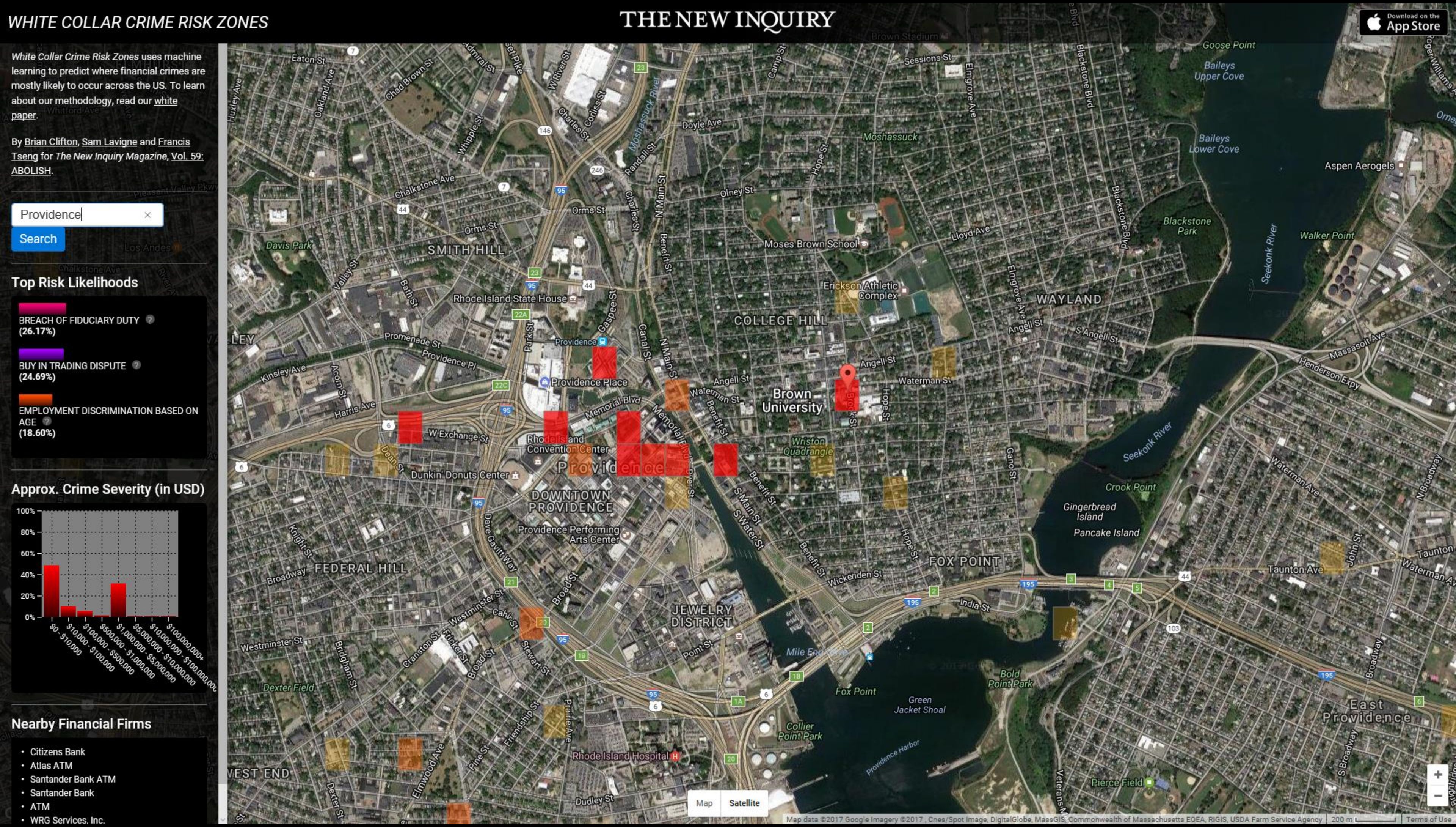
Abstract. Financial crime is a rampant but hidden threat. In spite of this, predictive policing systems disproportionately target "street crime" rather than white collar crime. This paper presents the White Collar Crime Early Warning System (WCCEWS), a white collar crime predictive model that uses random forest classifiers to identify high risk zones for incidents of financial crime.

Keywords: Criminal justice; crime models; capitalism, financial malfeasance; white collar crime; police patrol.

By SAM LAVIGNE, FRANCIS TSENG and BRIAN CLIFTON APRIL 26, 2017



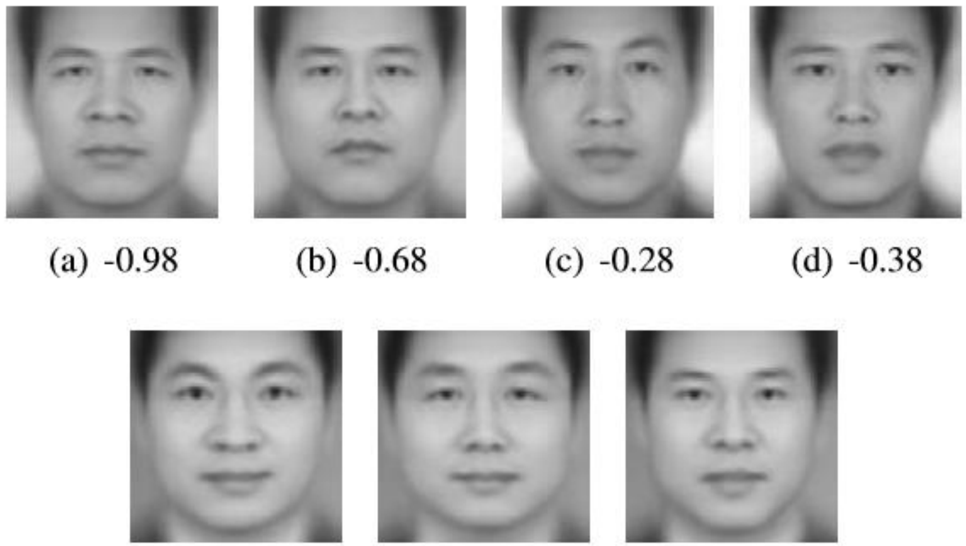
[https://whitecollar.thenewinquiry.com/]



[https://whitecollar.thenewinquiry.com/]

Recently researchers have demonstrated the effectiveness of applying machine learning techniques to facial features to quantify the "criminality" of an individual[21].

21 X. Wu and X. Zhang, "Automated inference on criminality using face images," *CoRR*, vol. abs/1611.04135, 2016.



(a) -0.98 (b) -0.68 (c) -0.28 (d) -0.38

(e) 0.76 (f) 0.98 (g) 0.66

Figure 13. (a), (b), (c) and (d) are the four subtypes of criminal faces corresponding to four cluster centroids on the manifold of S_c; (e), (f) and (g) are the three subtypes of non-criminal faces corresponding to three cluster centroids on the manifold of S_n. The number associated with each face is the average score of human judges (-1 for criminals; 1 for non-criminals).

Recently researchers have demonstrated the effectiveness of applying machine learning techniques to facial features to quantify the "criminality" of an individual[21].

[21] X. Wu and X. Zhang, "Automated inference on criminality using face images," *CoRR*, vol. abs/1611.04135, 2016.

We therefore plan to augment our model with facial analysis and psychometrics to identify potential financial crime at the individual level. As a proof of concept, we have downloaded the pictures of 7000 corporate executives whose LinkedIn profiles suggest they work for financial organizations, and then averaged their faces to produce generalized white collar criminal subjects unique to each high risk zone. Future efforts will allow us to predict white collar criminality through real-time facial analysis.

Face detection + facial landmark detection +
image warping + averaging/PCA!

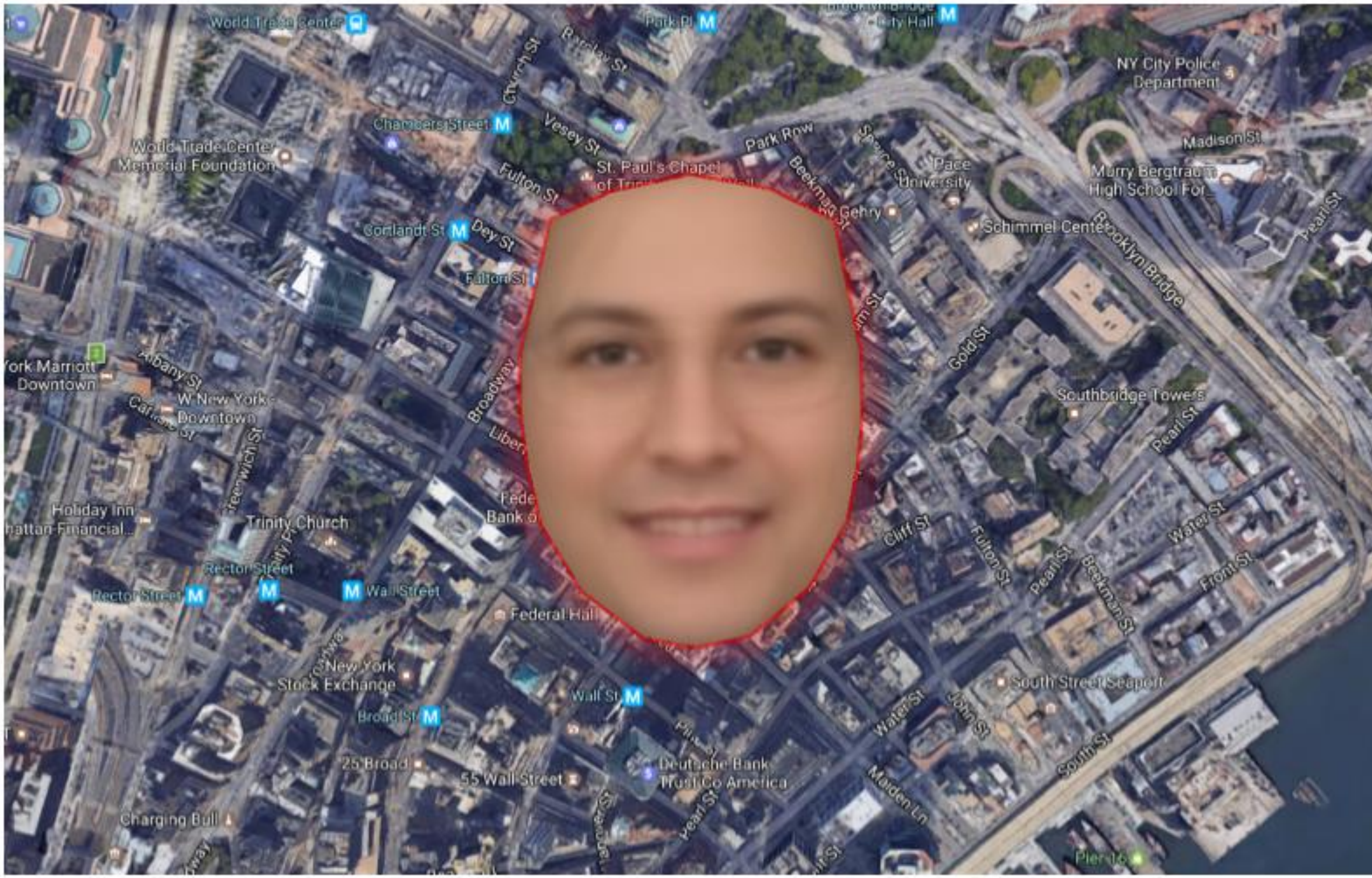
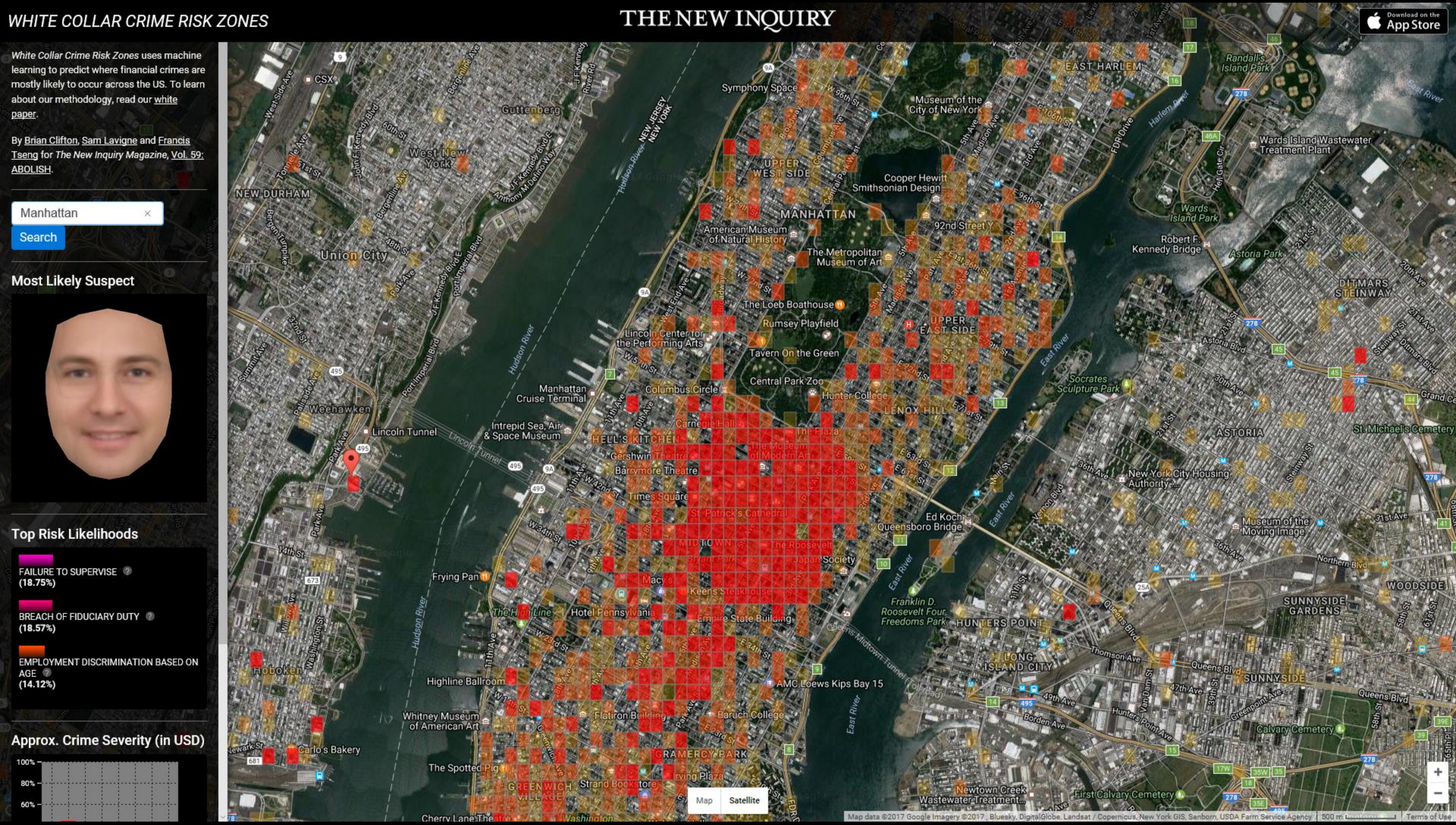


Fig. 7: Predicted White Collar Criminal for 40.7087811, -74.0064149

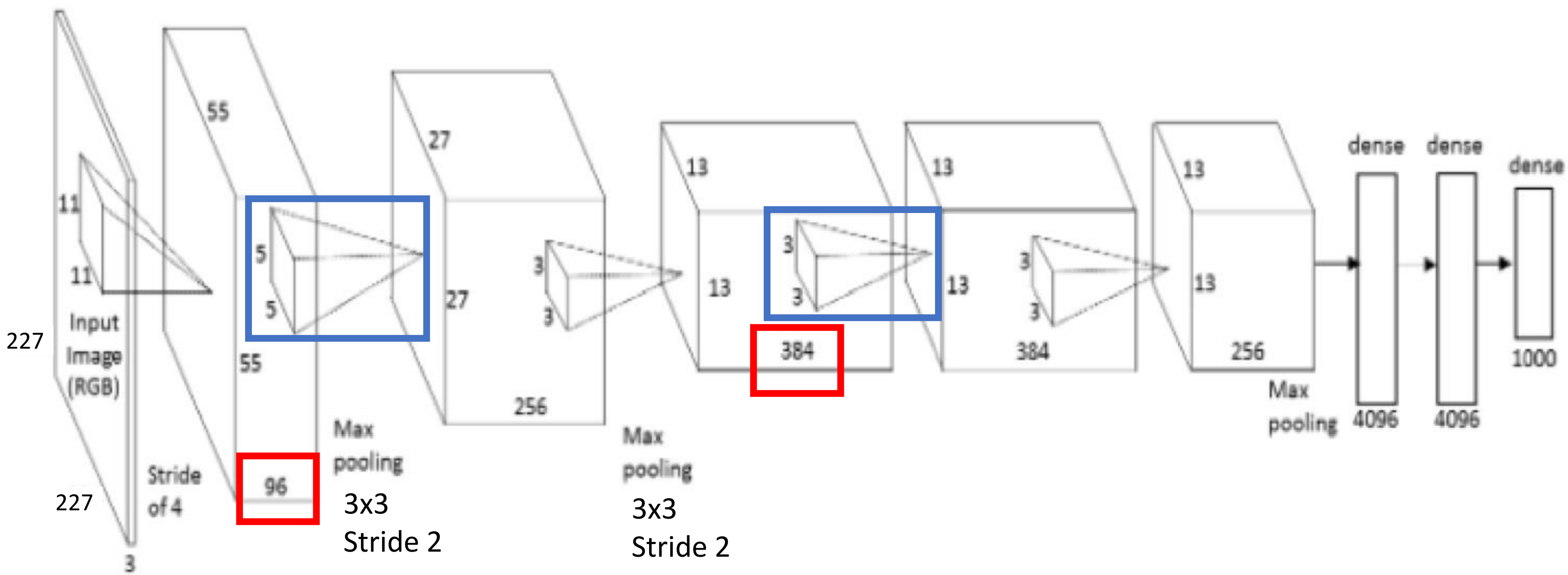


[https://whitecollar.thenewinquiry.com/]

[Krizhevsky et al. 2012]

AlexNet diagram (simplified)

Input size
227 x 227 x 3



Conv 1
11 x 11 x 3
Stride 4
96 filters

Conv 2 Eh?
5 x 5 x 48
Stride 1
256 filters

Conv 3
3 x 3 x 256
Stride 1
384 filters

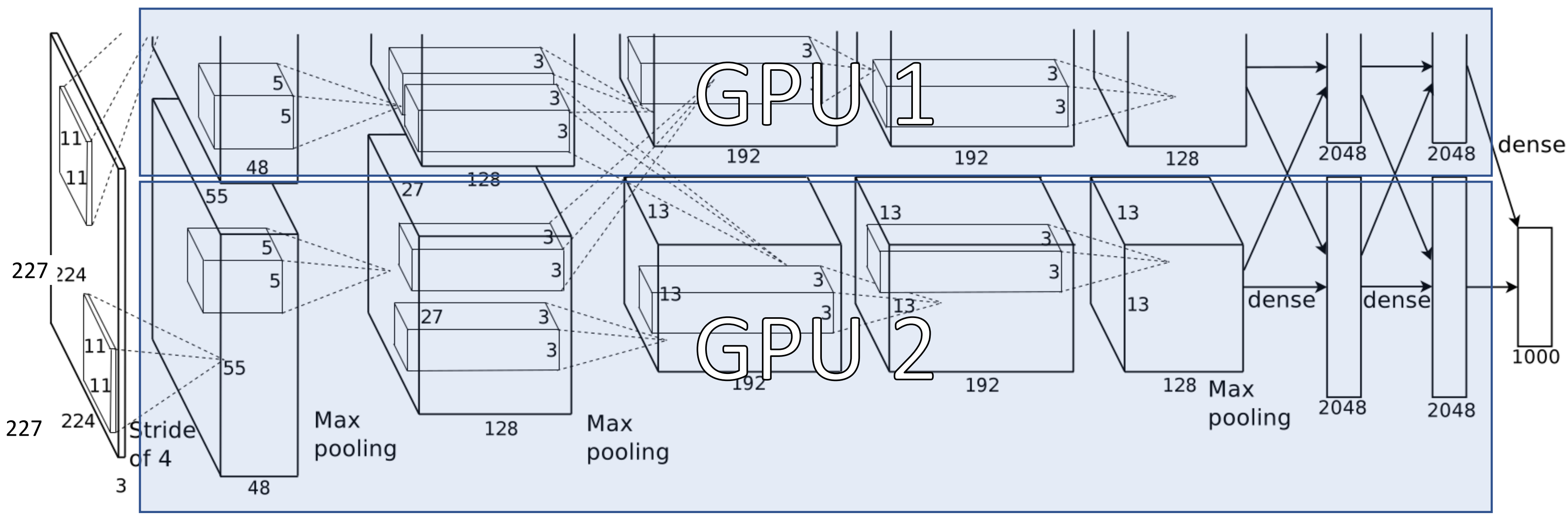
Conv 4
3 x 3 x 192
Stride 1
384 filters

Conv 4
3 x 3 x 192
Stride 1
256 filters

Eh? Shouldn't these be equal?

AlexNet diagram (unsimplified)

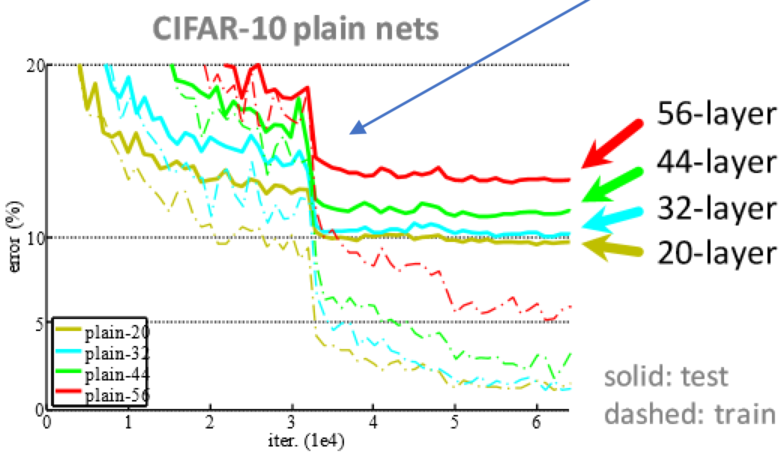
Not enough memory for all the weights – use two GPUs!

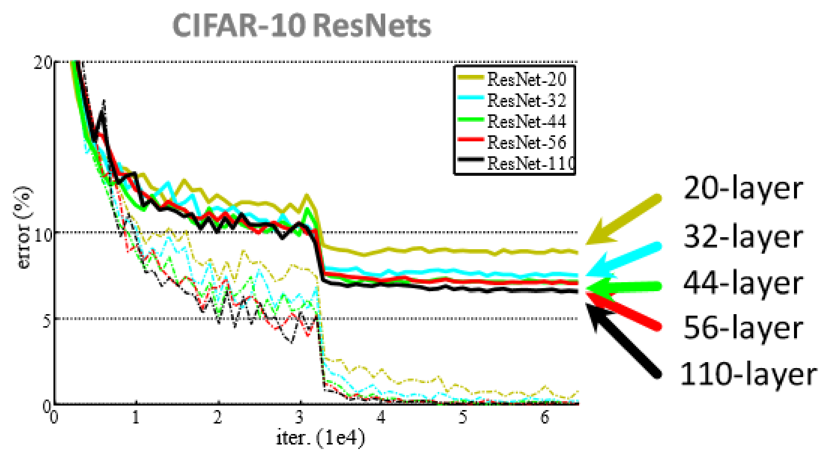


Convergence cliff

Why so steep?

CIFAR-10 experiments

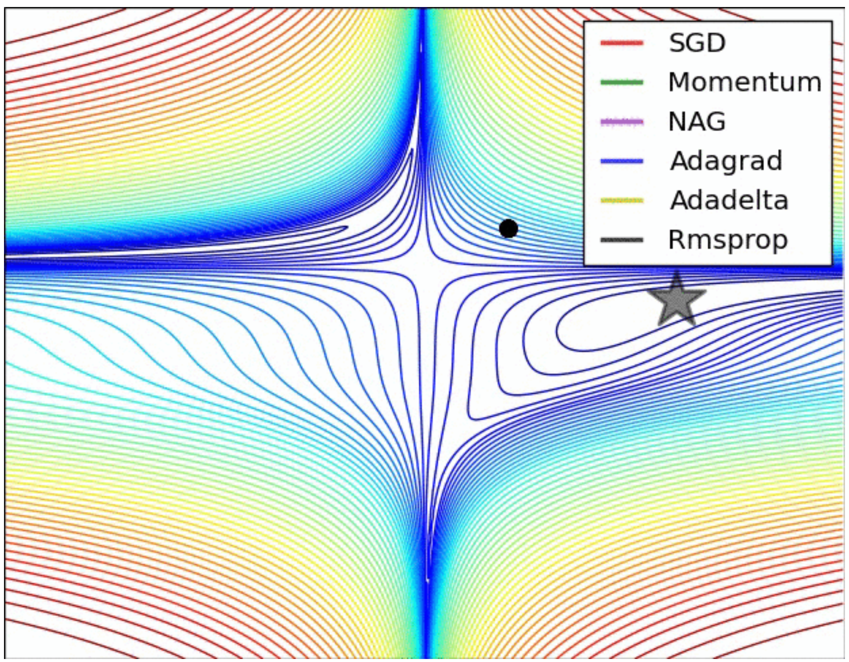


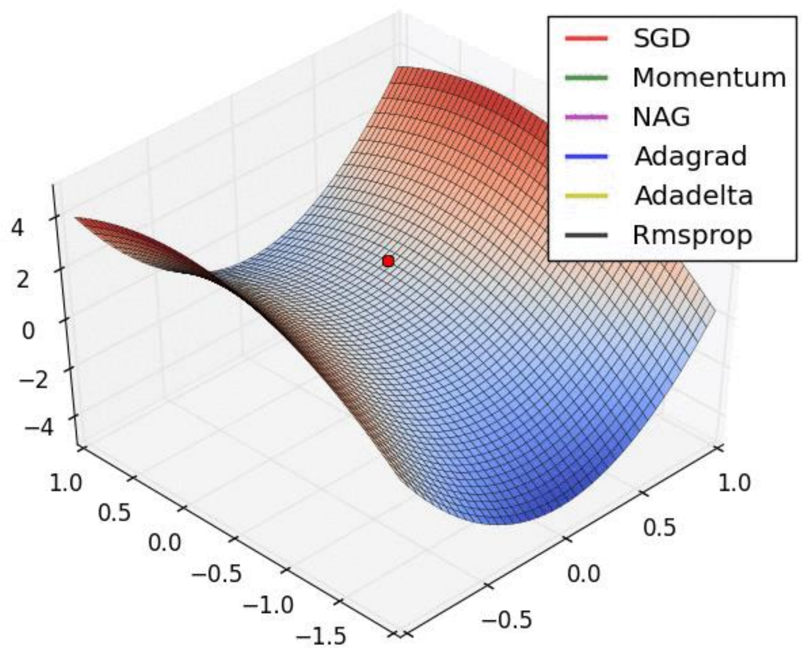


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

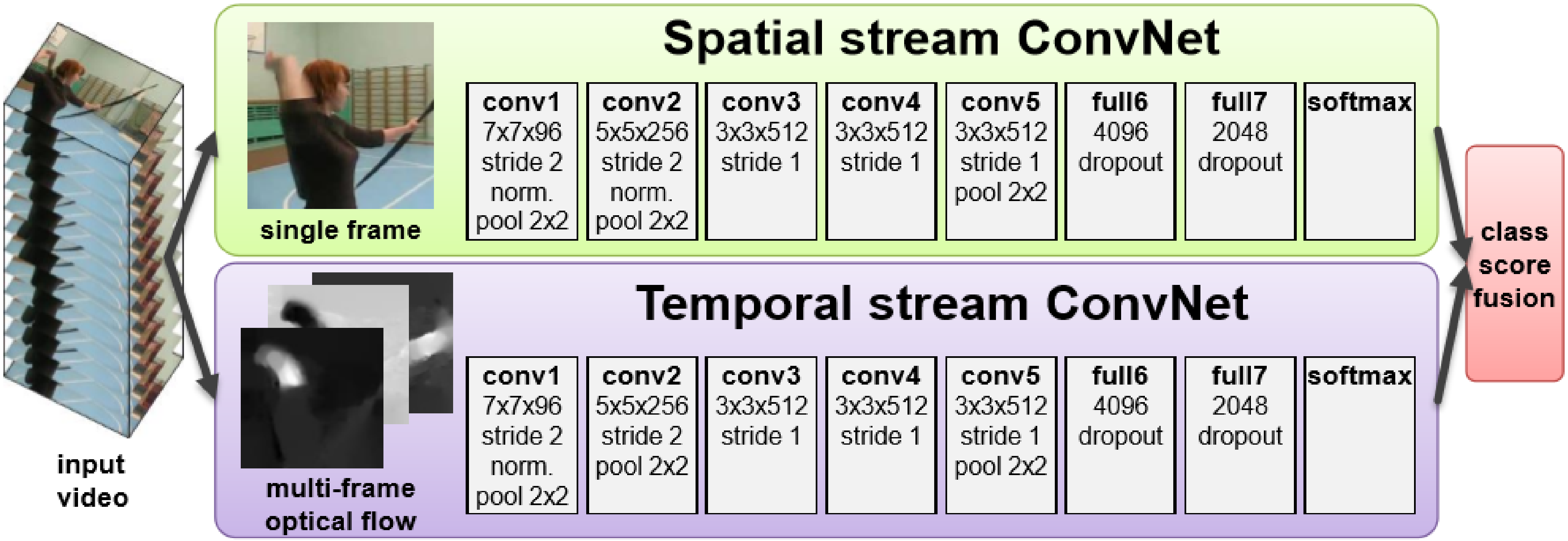
Flat regions in energy landscape



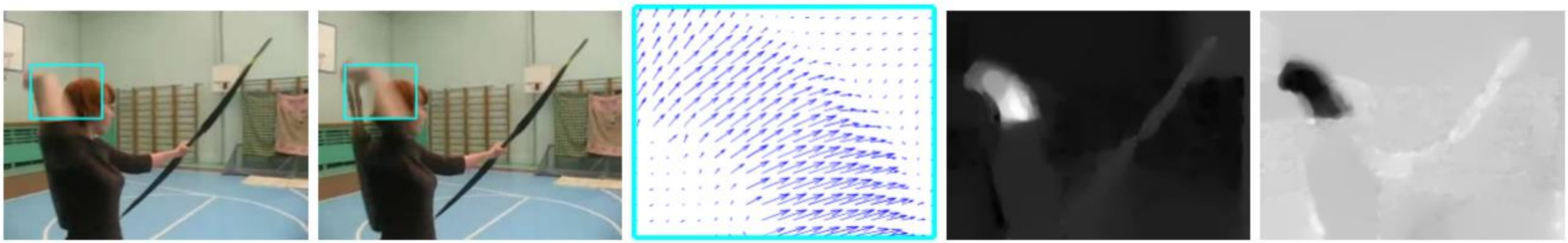


What about learning across ‘domains’?

Two-stream networks – *action recognition*



[Simonyan et al. 2014]



(a) (b) (c) (d) (e)

[Simonyan et al. 2014]

Learning Deep Representations For Ground-to-Aerial Geolocalization

Tsung-Yi Lin, Yin Cui, Serge Belongie, James Hays

CORNELL NYCTECH

Georgia Tech

CVPR 2015

View From Your Window Contest

June 9, 2010 – Feb. 4, 2015

Where was the photo taken?

Ans:
Milano, Italy

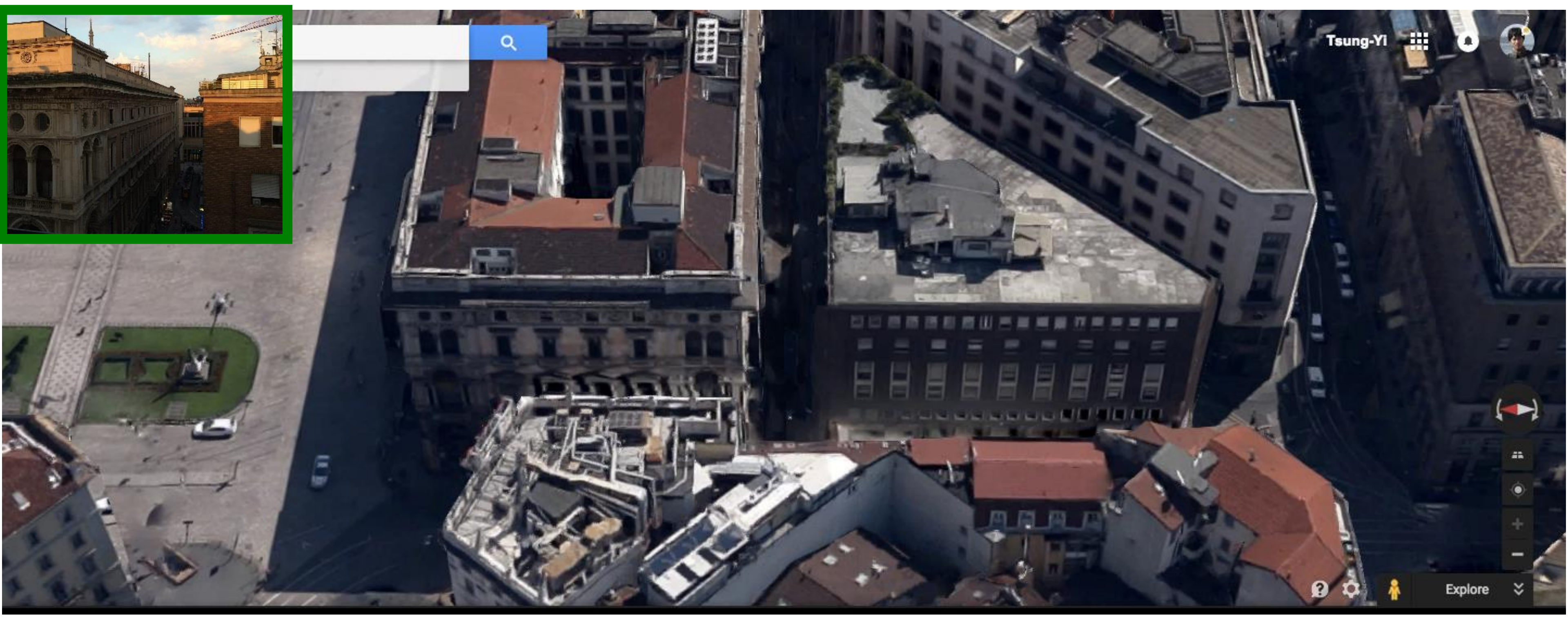
To Geolocalize a Photo

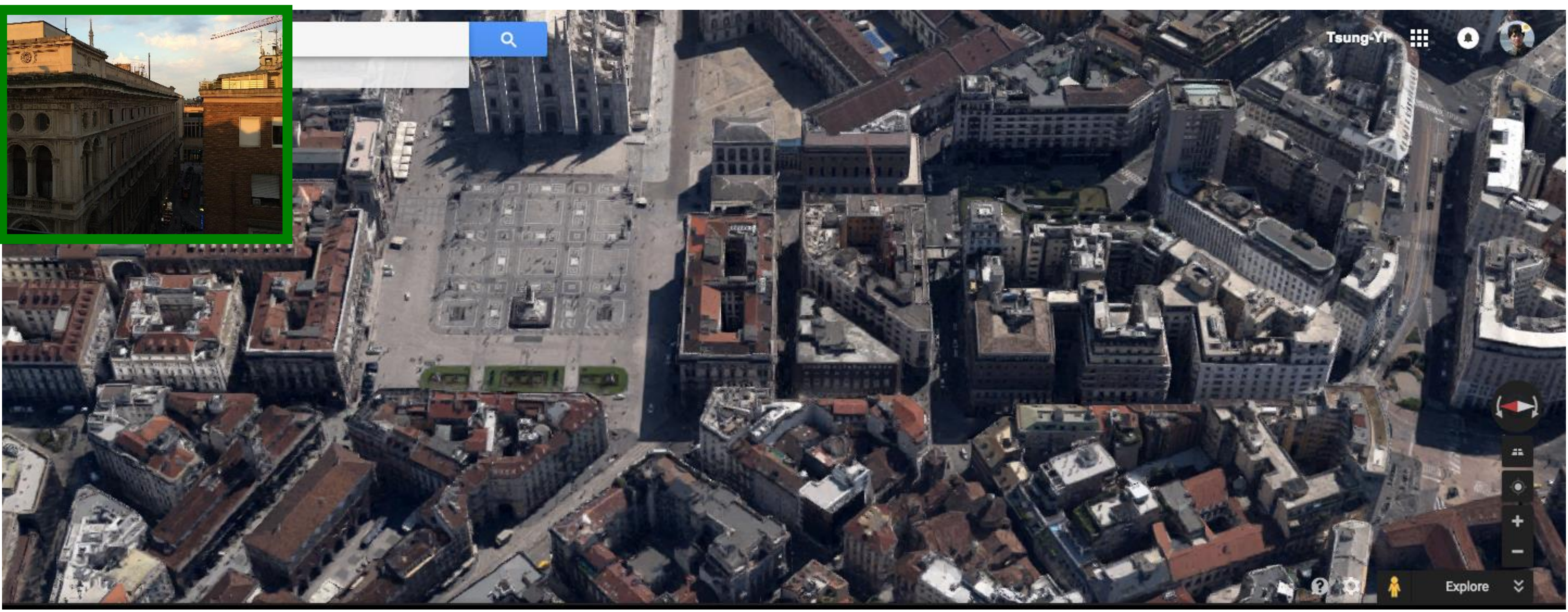
- One can capture every corner on the earth




...

To Geolocalize a Photo




Tsung-Yi
Explore


Tsung-Yi
Explore

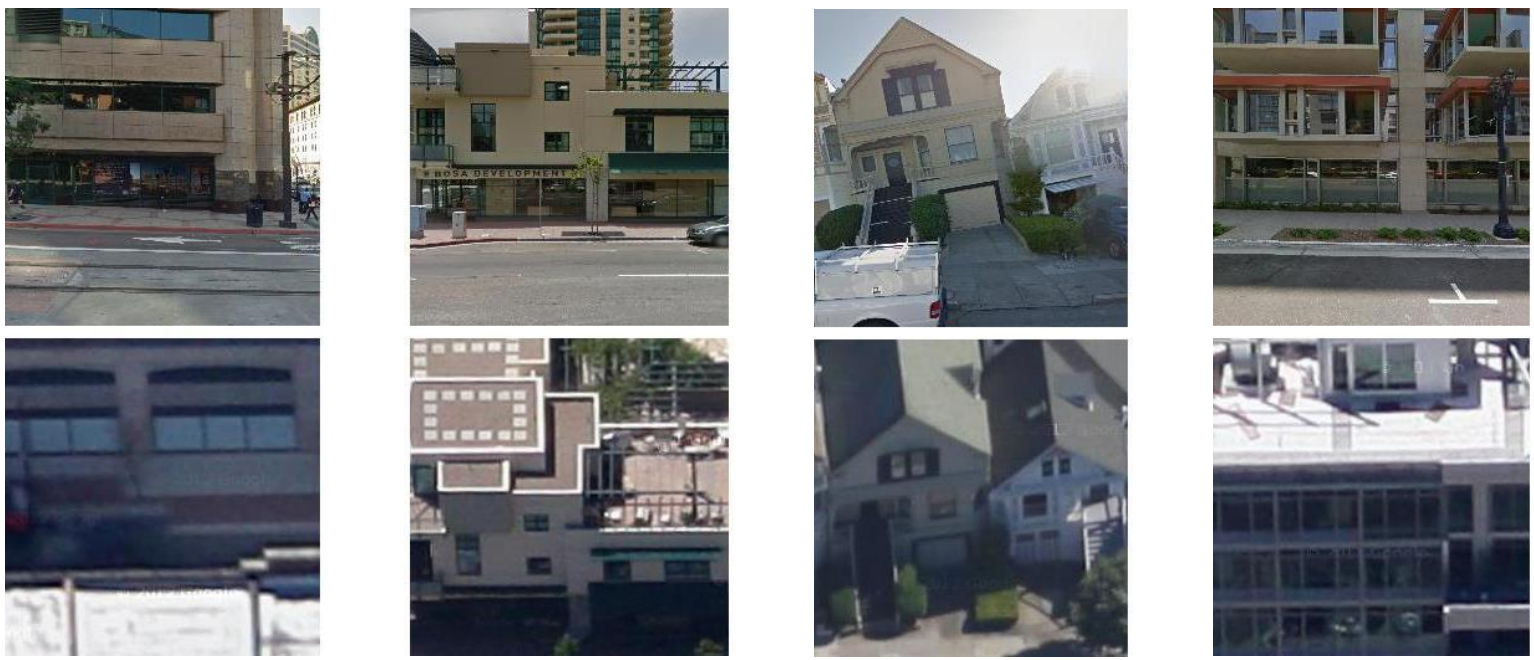
How To Match Ground-to-Aerial?

Shan et al., Accurate Geo-registration by Ground-to-Aerial Image Matching, 3DV'14
Bansal et al., Ultra-wide baseline façade matching for geo-localization, ECCV workshop'12

Are these the same location?

Ground

Aerial



Are these the same location?

Ground

Aerial

Why Don’t You Just…

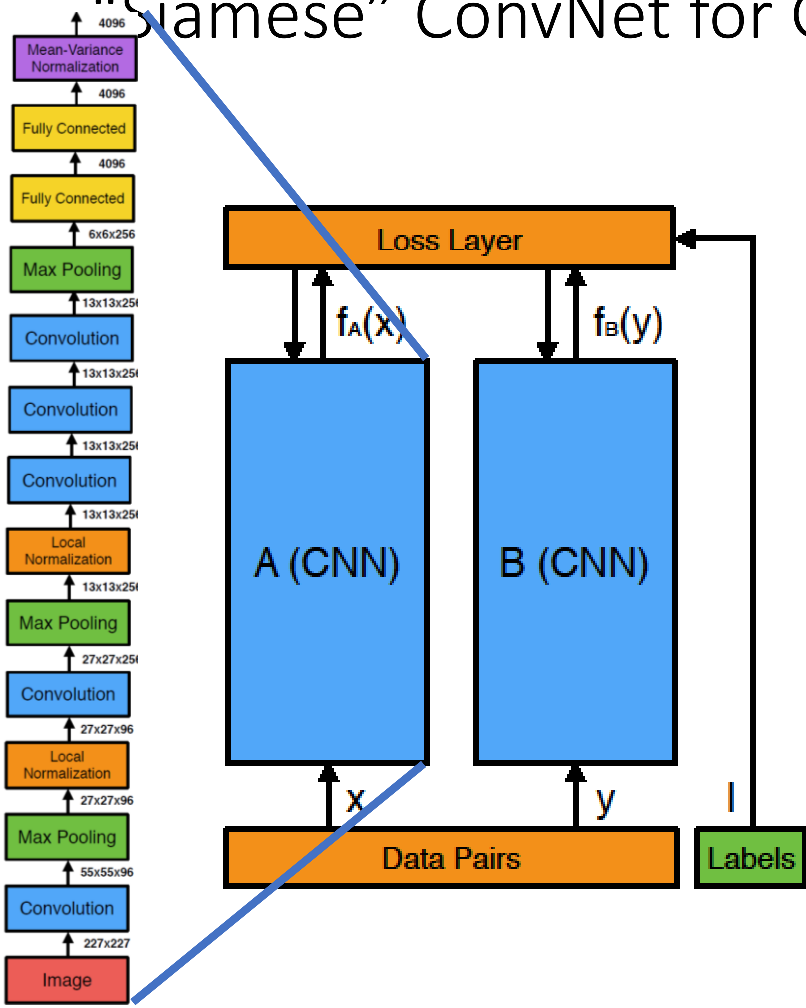
- Sparse Keypoint Matching + RANSAC

Cross-view Pairs

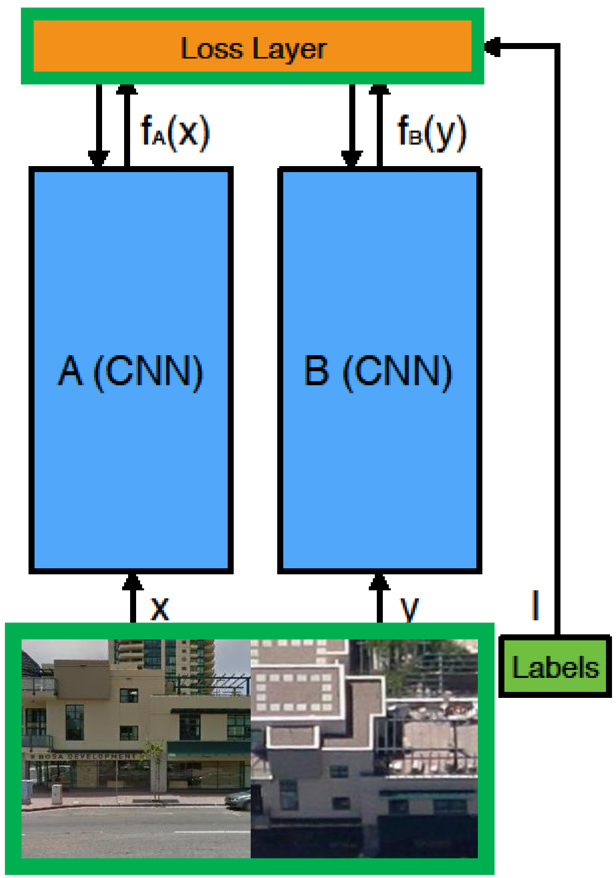
Ground

Aerial

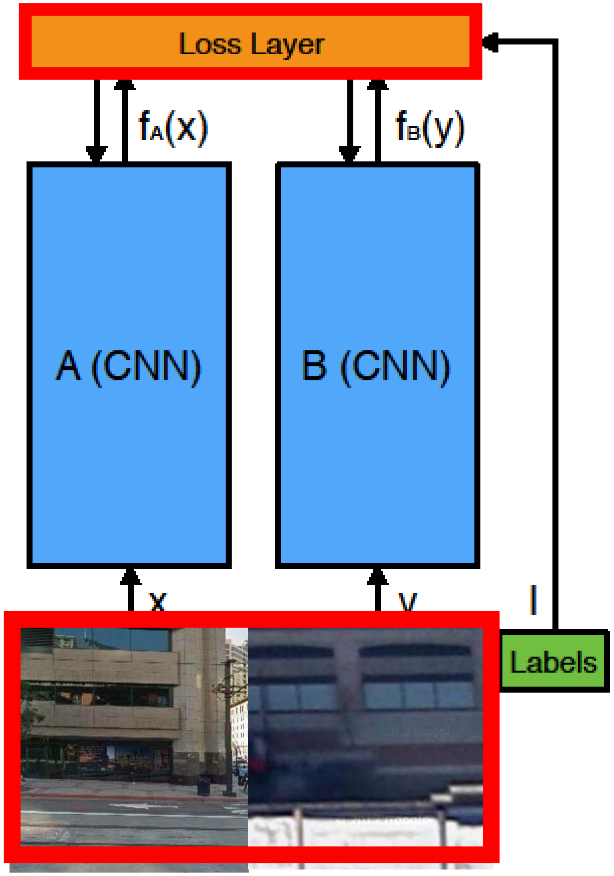
“Siamese” ConvNet for Ground-to-Aerial Matching

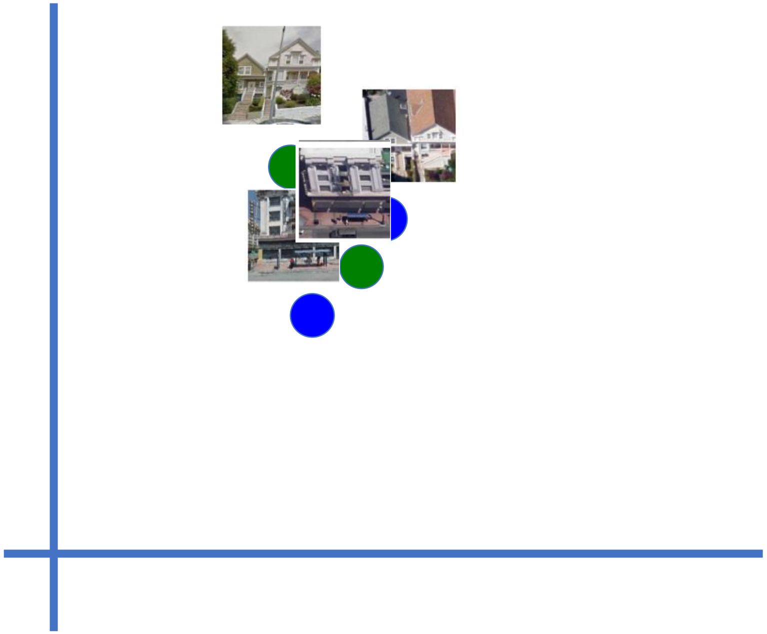


“Siamese” ConvNet for Ground-to-Aerial Matching



“Siamese” ConvNet for Ground-to-Aerial Matching





Contrastive Loss

Loss Function:

- **For similar pairs:**

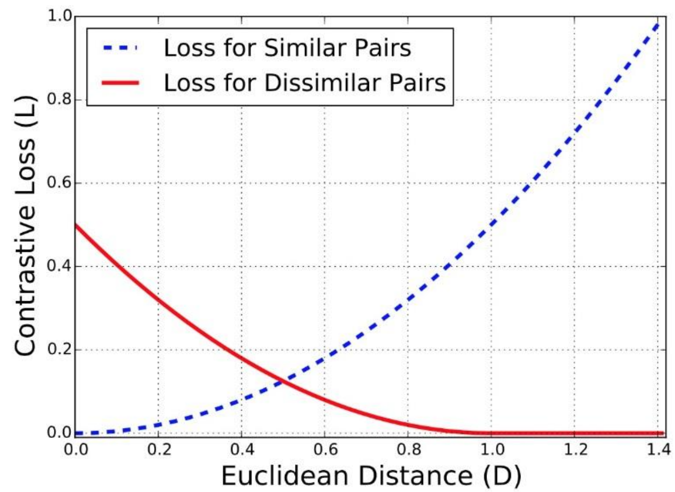
||f()-f() || ^{2}

- **For dissimilar pairs**

max(0, m- ||f()-f() ||) ^{2}

red: similar pairs

blue: dissimilar pairs

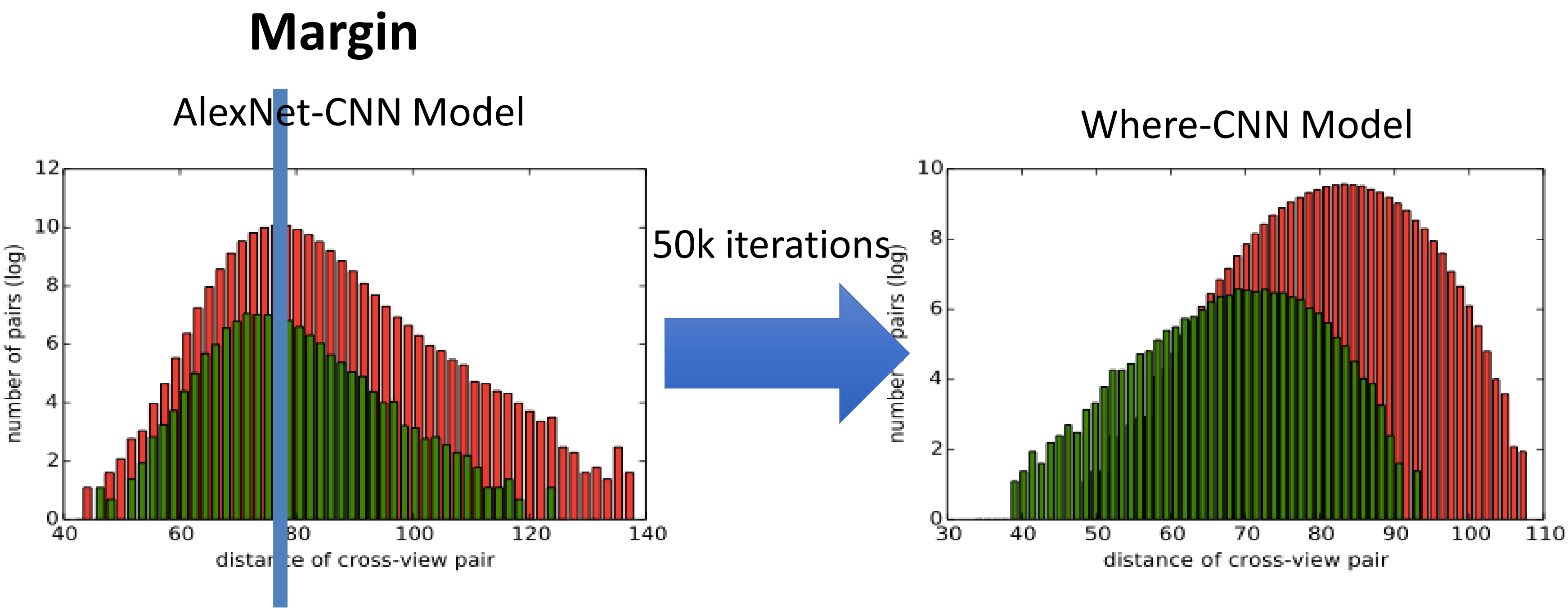


m = level from which to invert

Hadsell, Chopra, Yann LeCun, **Dimensionality Reduction by Learning an Invariant Mapping**, CVPR06

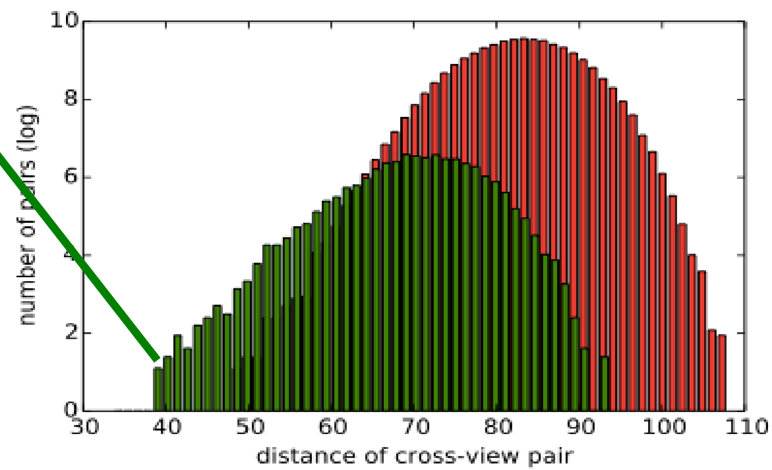
Pair Distance Distribution

Green: positive pairs
Red: negative pairs

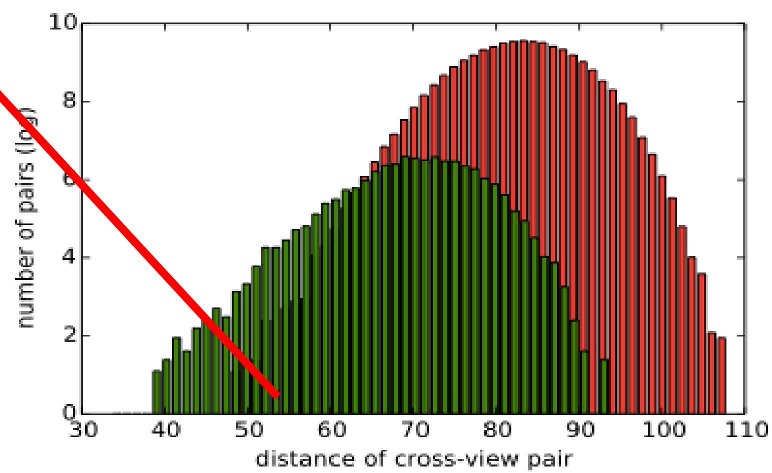


Distribution of distances
between positive/negative pairs.

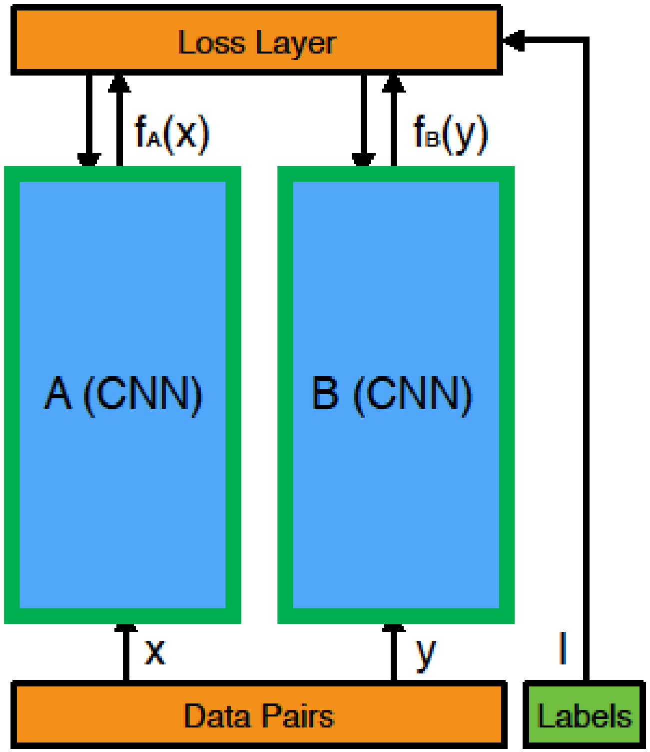
(a) Easy positive pairs.



(b) Hard negative pairs.

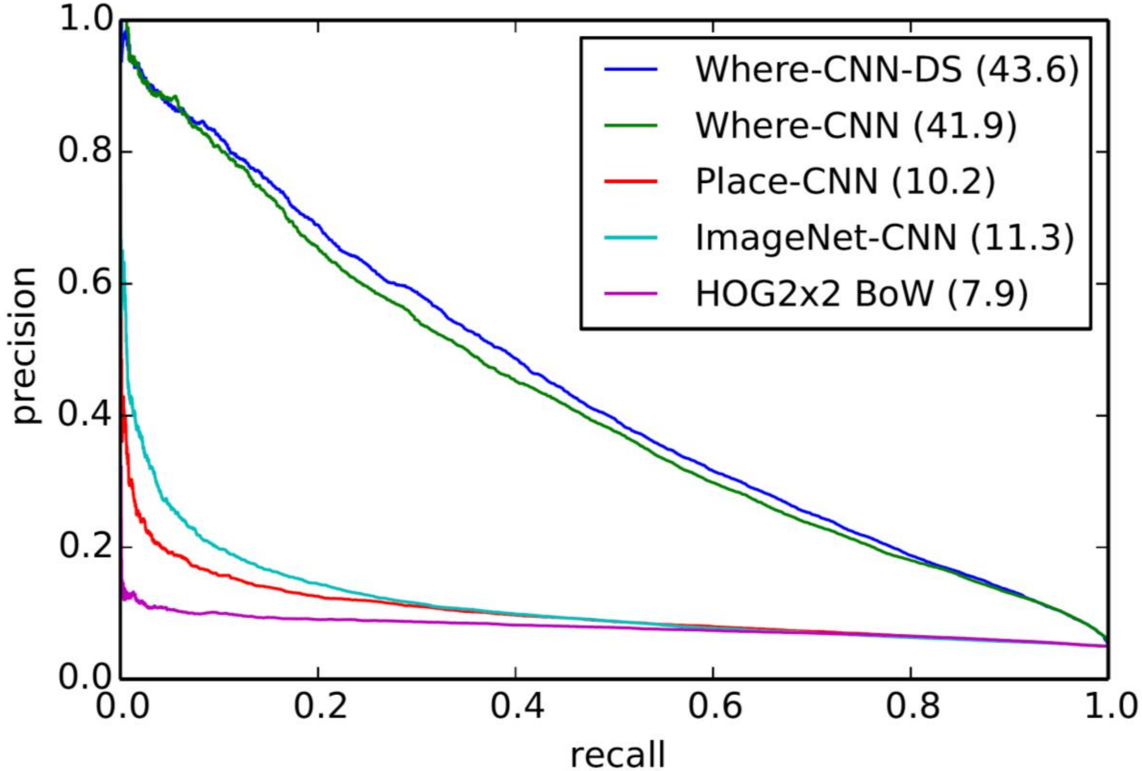


Share The Same Parameters?



For ground-aerial image pairs, should A, B share parameters?

Quantitative Evaluation

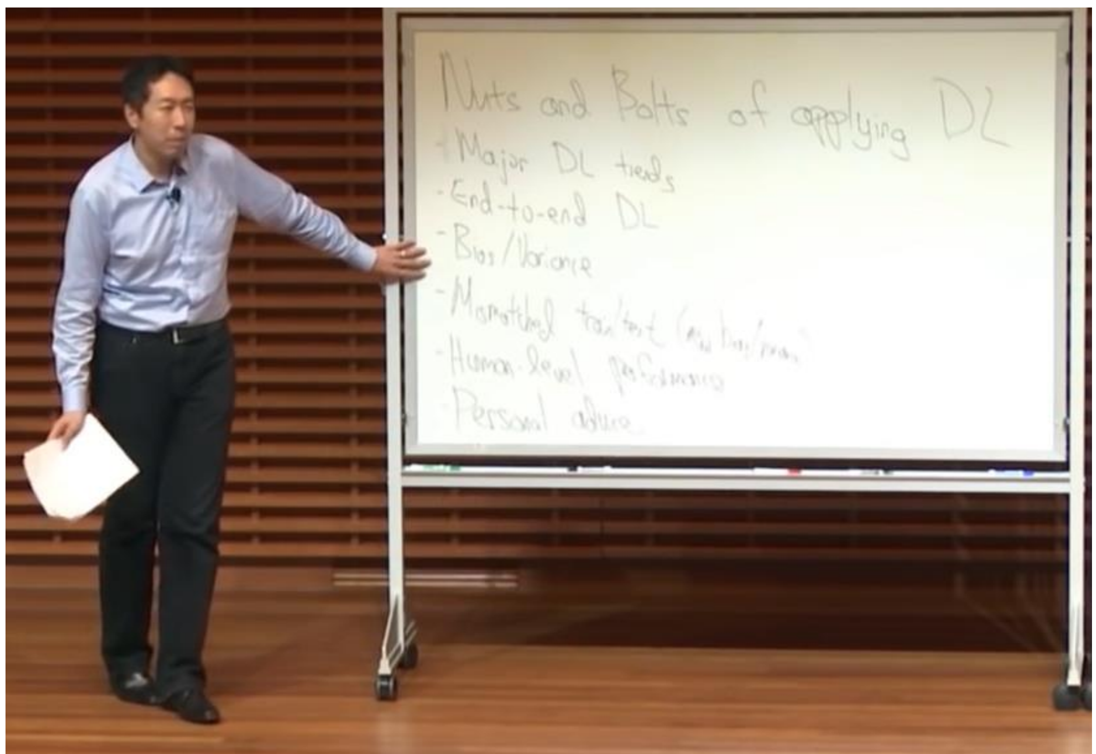


When something is not working…

…how do I know what to do next?

The Nuts and Bolts of Building Applications using Deep Learning

- Andrew Ng - NIPS 2016
- https://youtu.be/F1ka6a13S9I



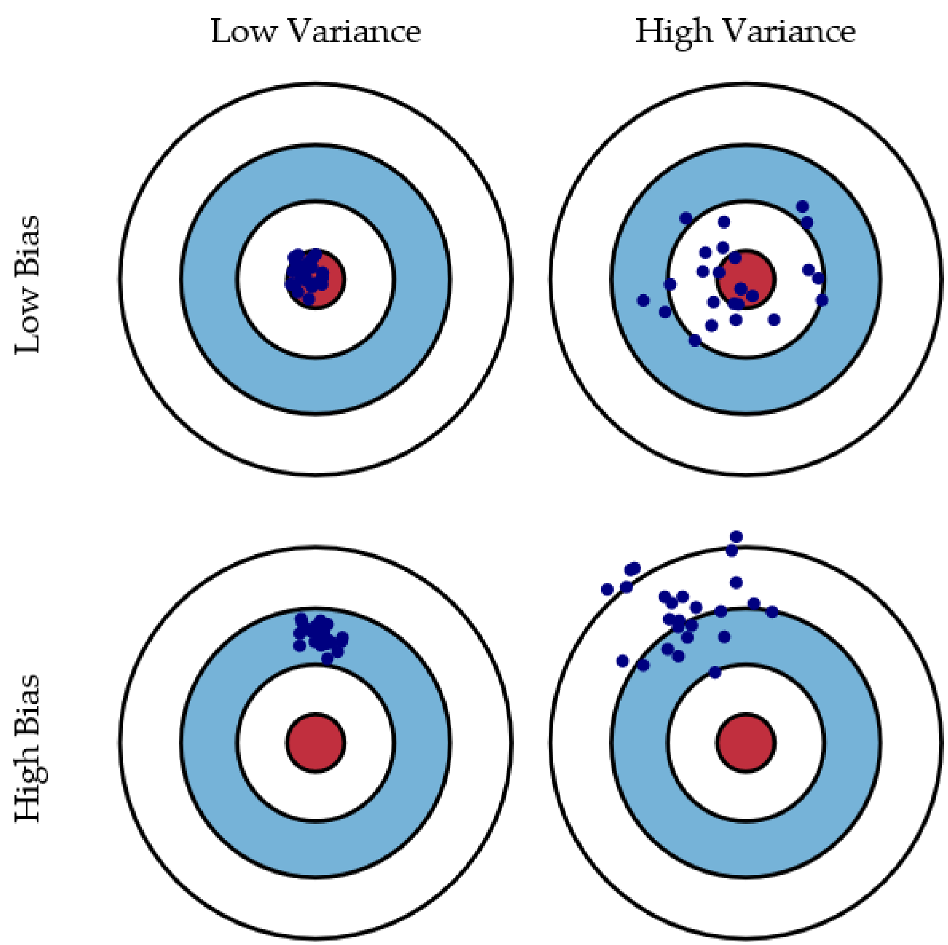
Go collect a dataset

- Most important thing:
  - Training data must represent target application!

- Take all your data
  - 60% training
  - 40% testing
    - 20% testing
    - 20% validation (or ‘development’)

Bias/variance trade-off

"It takes surprisingly long time to grok bias and variance deeply, but people that understand bias and variance deeply are often able to drive very rapid progress." *--Andrew Ng*



Bias = accuracy
Variance = precision

Scott Fortmann-Roe

Properties

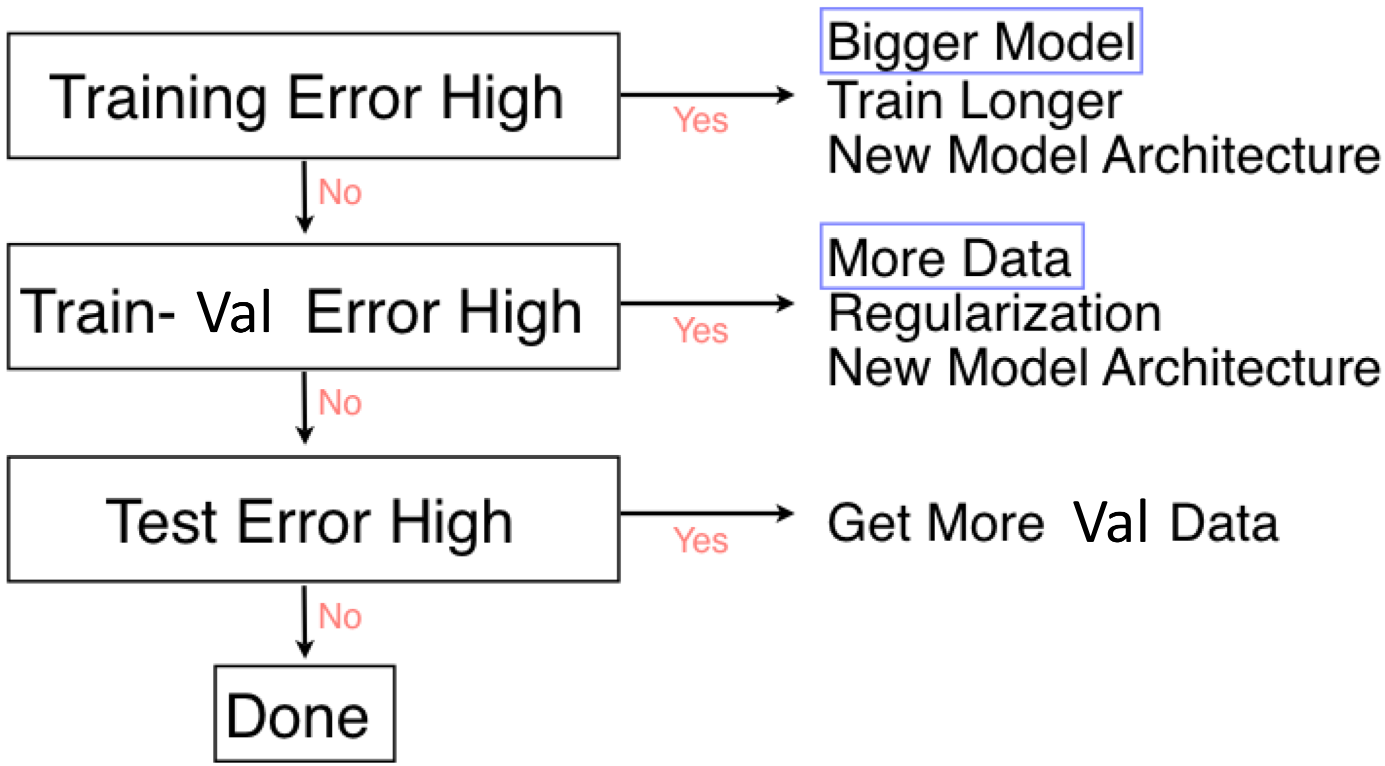
- Human level error = 1%
- Training set error = 10%
- Validation error = 10.2%
- Test error = 10.4%

"Bias"

"Variance"

Overfitting to validation

The Nuts and Bolts of Building Applications Using Deep Learning

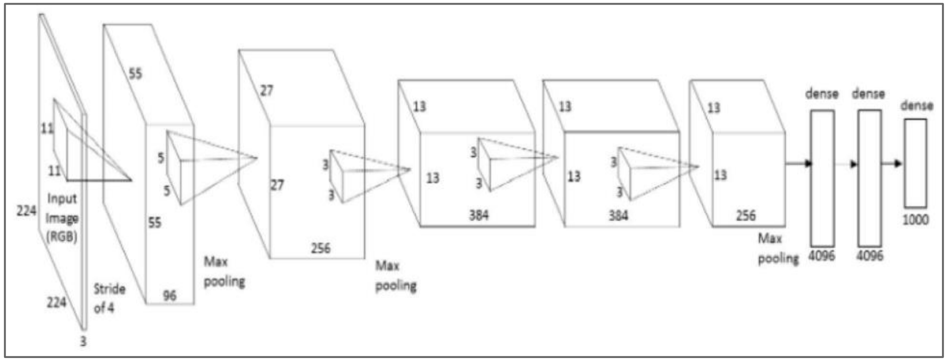


[Andrew Ng]

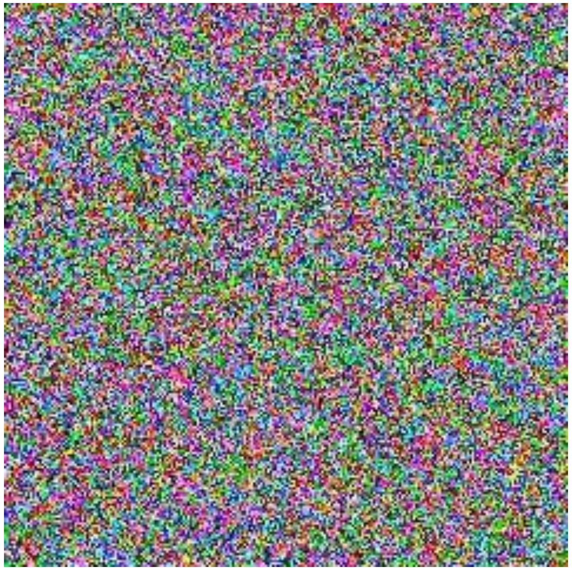
Interesting CNN properties

http://yosinski.com/deepvis

What input to a neuron maximizes a class score?



To visualize the function of a specific unit in a neural network, we synthesize an input to that unit which causes high activation.

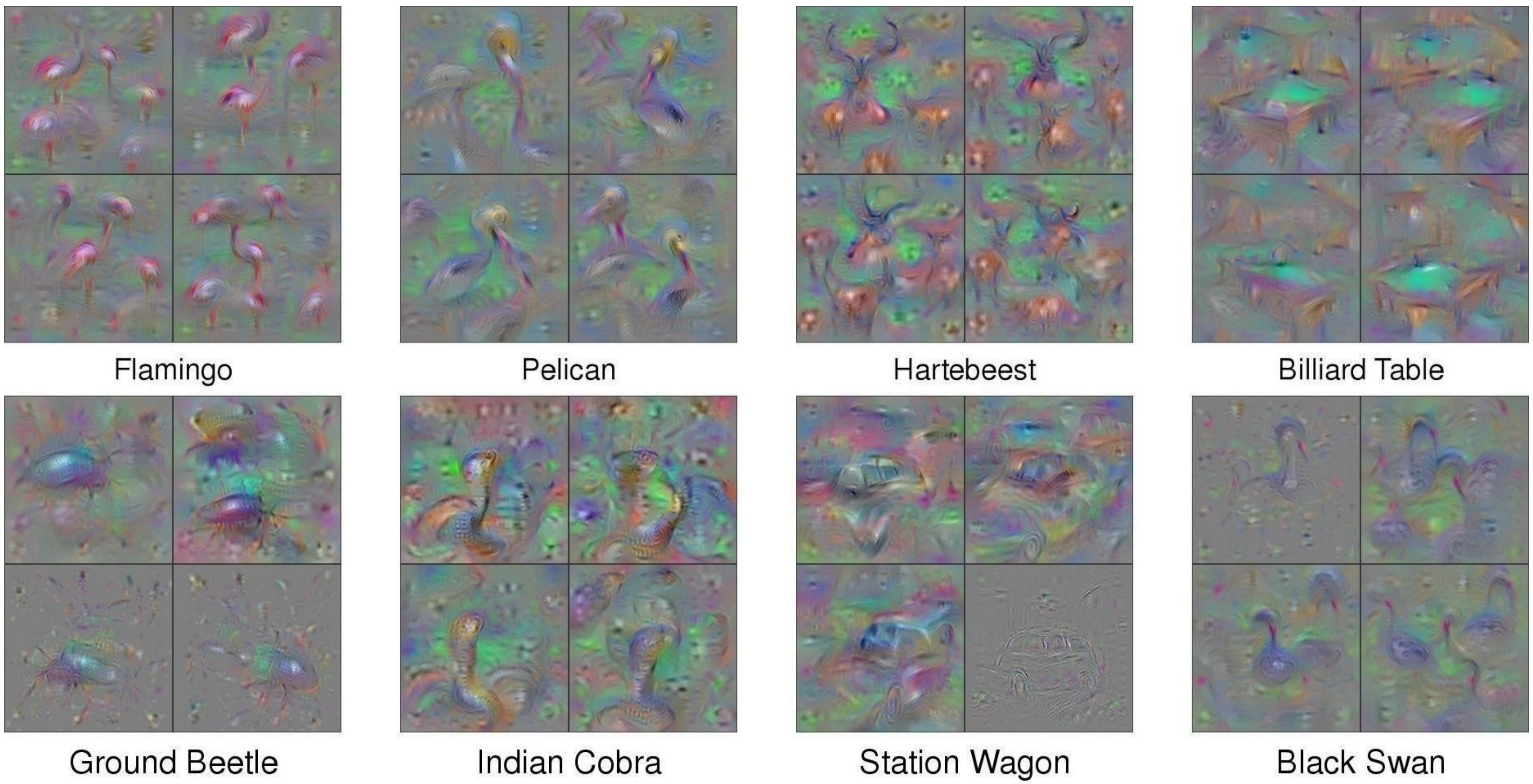


Neuron of choice *i*
An image of random noise *x*.

Repeat:

1. Forward propagate: compute activation $a_i(x)$
2. Back propagate: compute gradient at neuron $\partial a_i(x) / \partial x$
3. Add small amount of gradient back to noisy image.

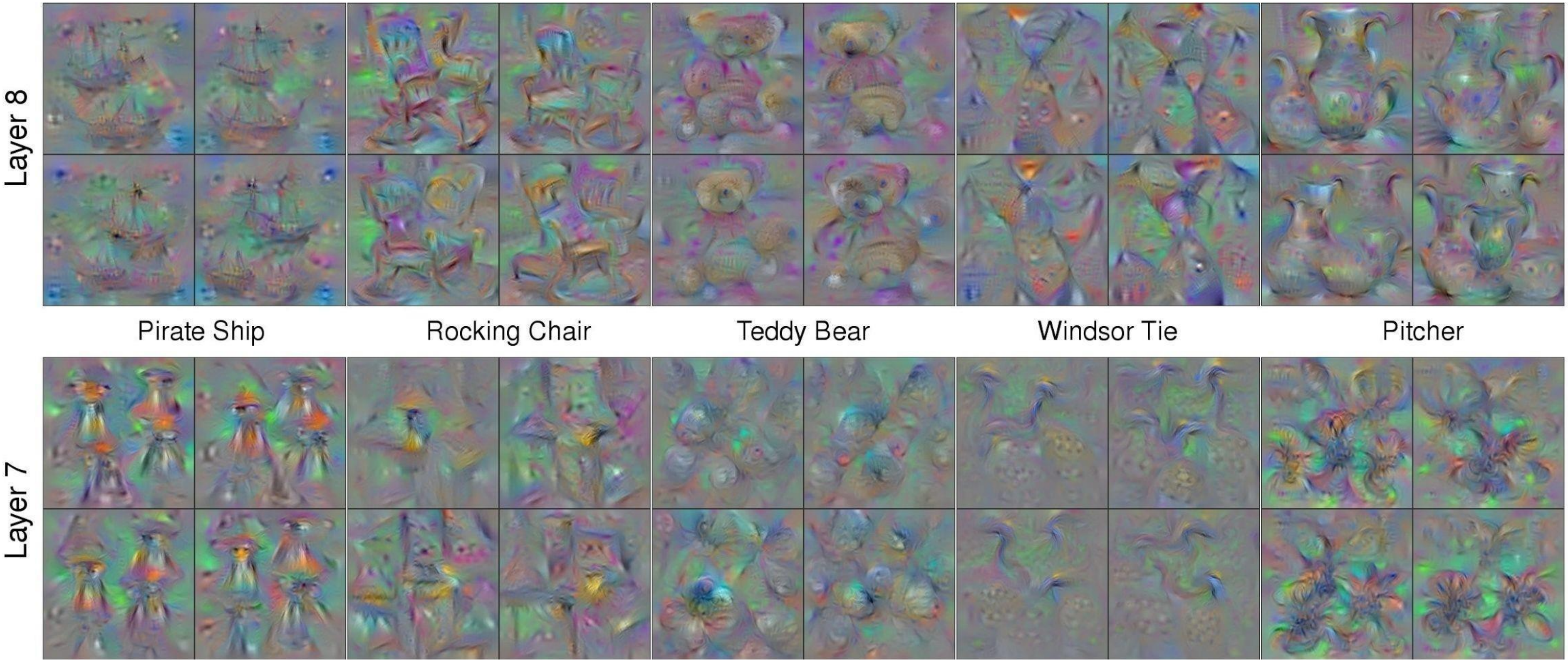
What image maximizes a class score?



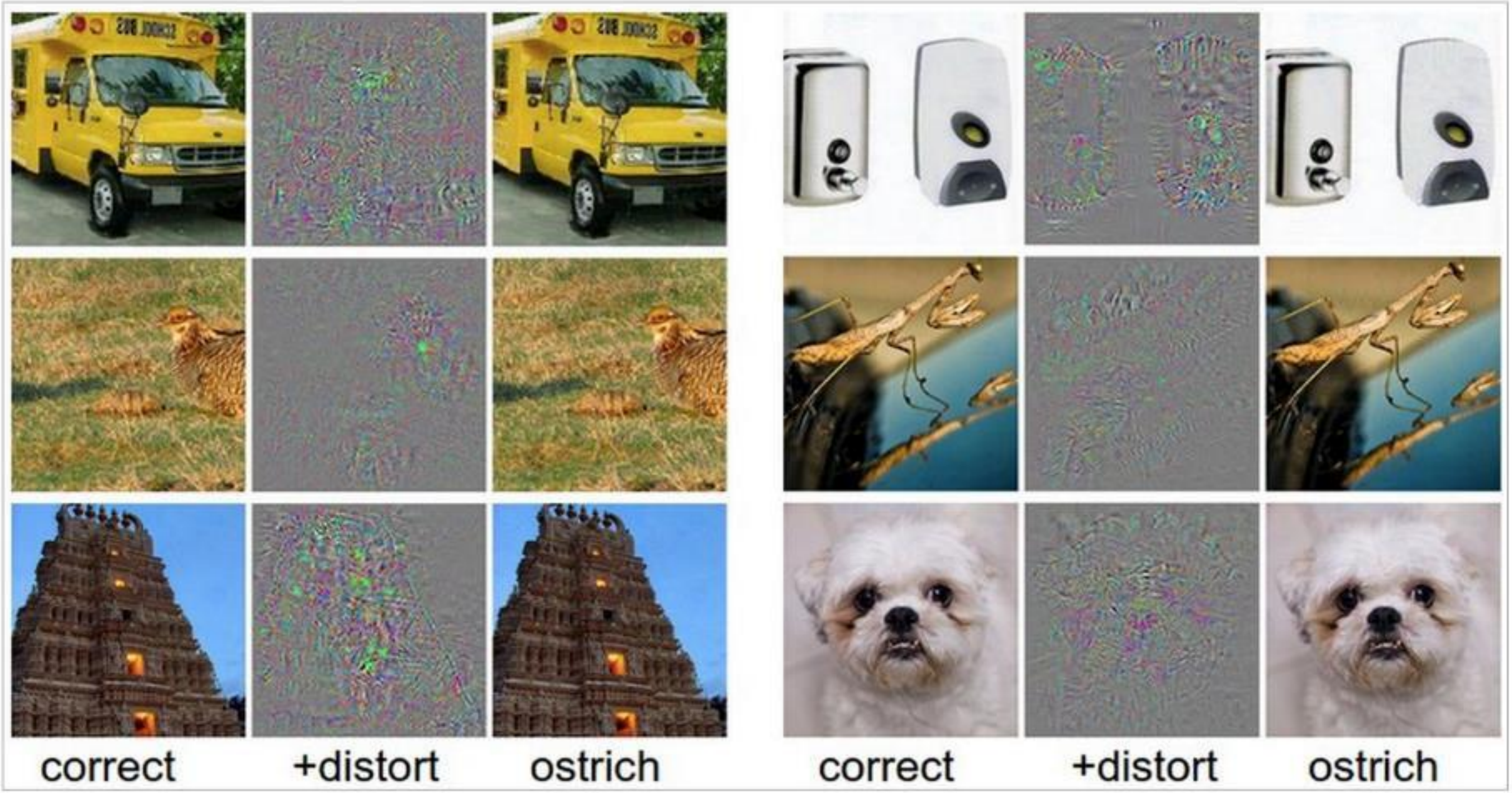
[Understanding Neural Networks Through Deep Visualization, Yosinski et al. , 2015]

http://yosinski.com/deepvis

What image maximizes a class score?



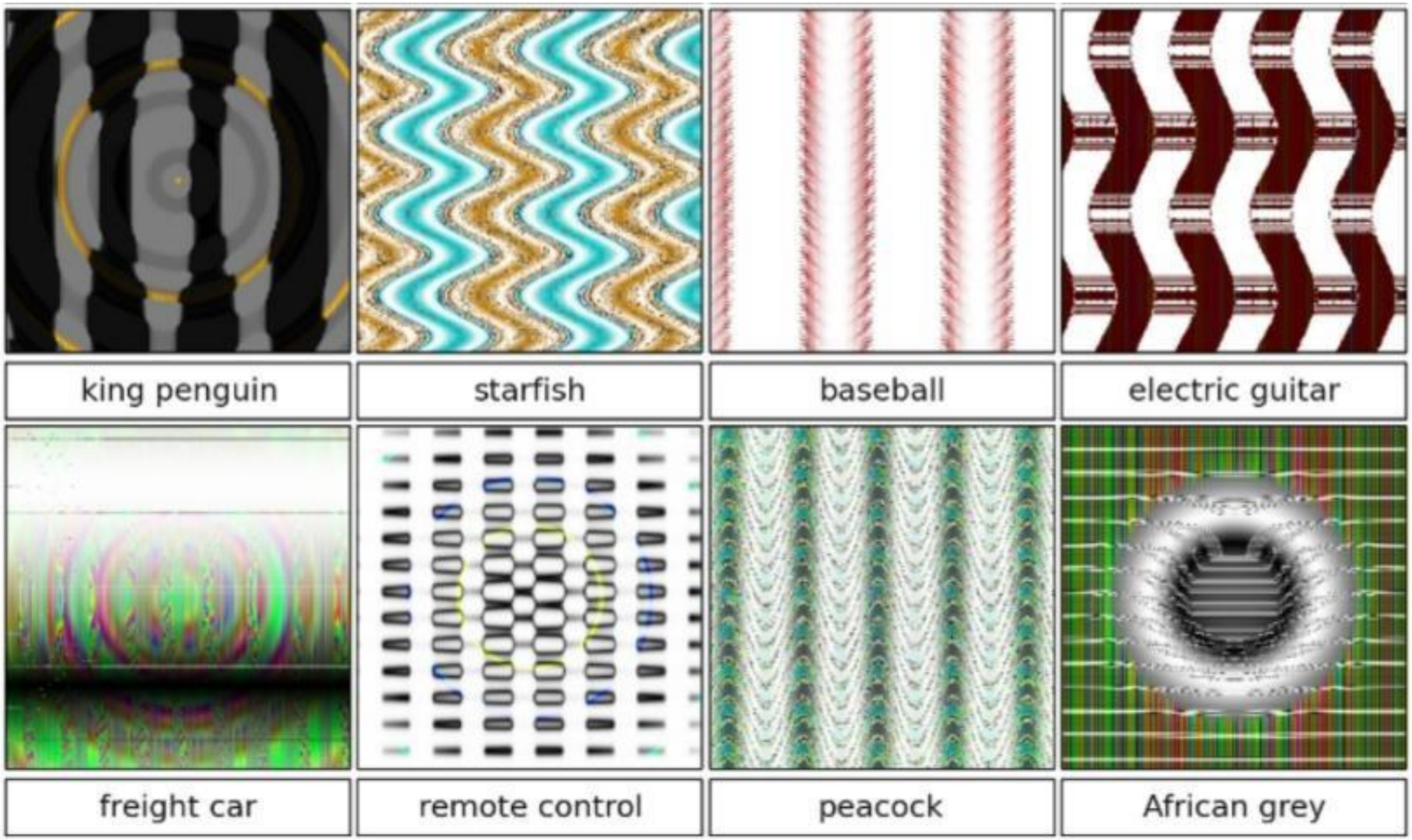
Breaking CNNs



Take a correctly classified image (left image in both columns), and add a tiny distortion (middle) to fool the ConvNet with the resulting image (right).

Intriguing properties of neural networks [Szegedy ICLR 2014]

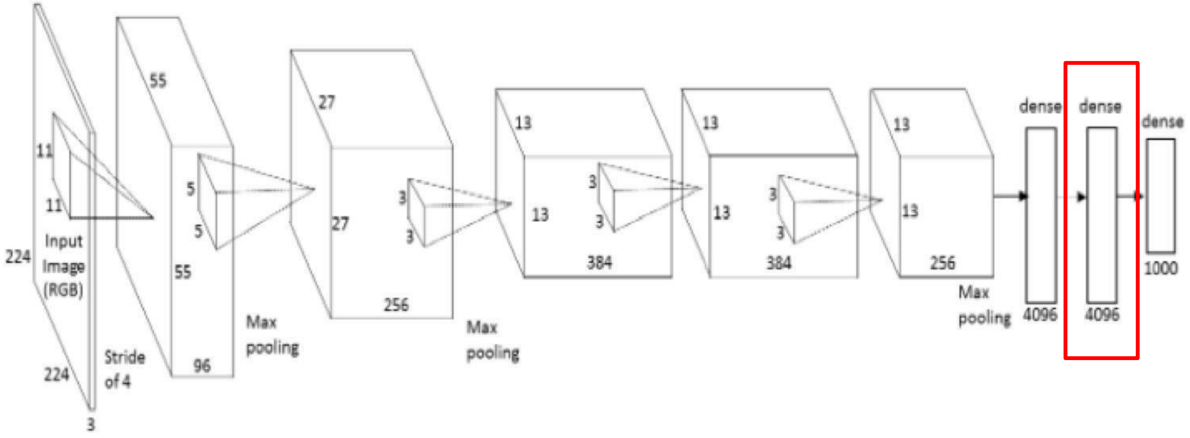
Breaking CNNs



Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

Reconstructing images

Question: Given a CNN code, is it possible to reconstruct the original image?



Andrej Karpathy

Reconstructing images

Find an image such that:

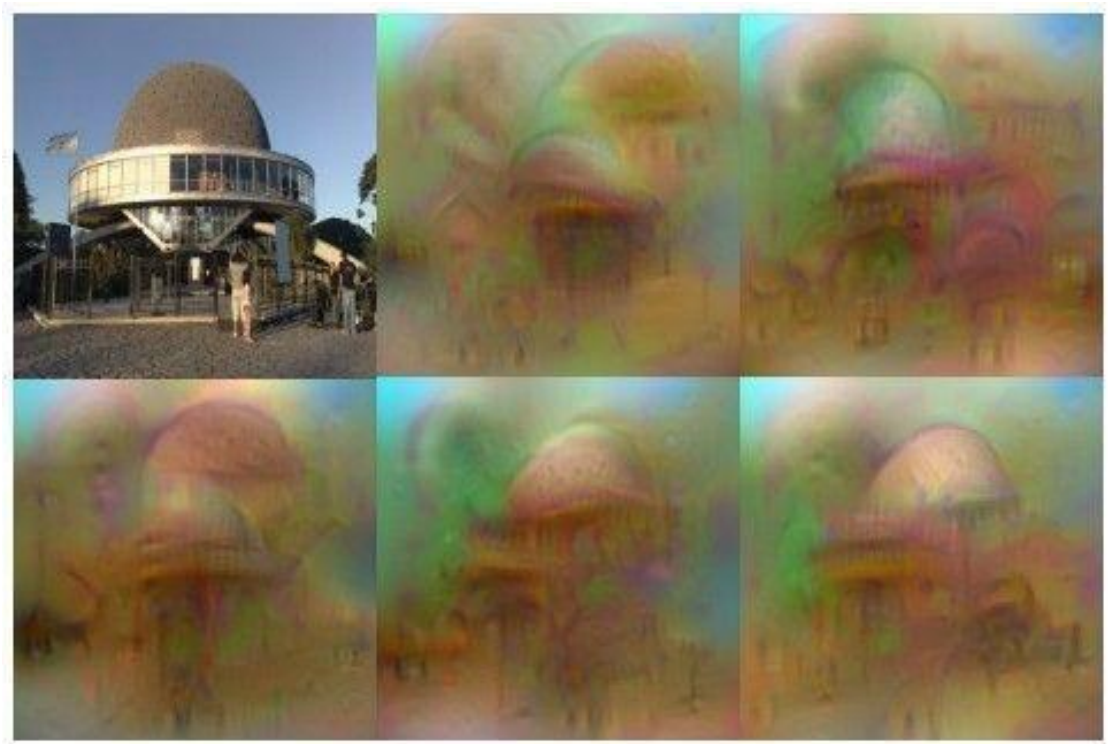
- Its code is similar to a given code
- It "looks natural"
    - Neighboring pixels should look similar

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^{H \times W \times C}} \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda \mathcal{R}(\mathbf{x})$$

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

Reconstructing images

original image



Reconstructions from the 1000 log probabilities for ImageNet (ILSVRC) classes

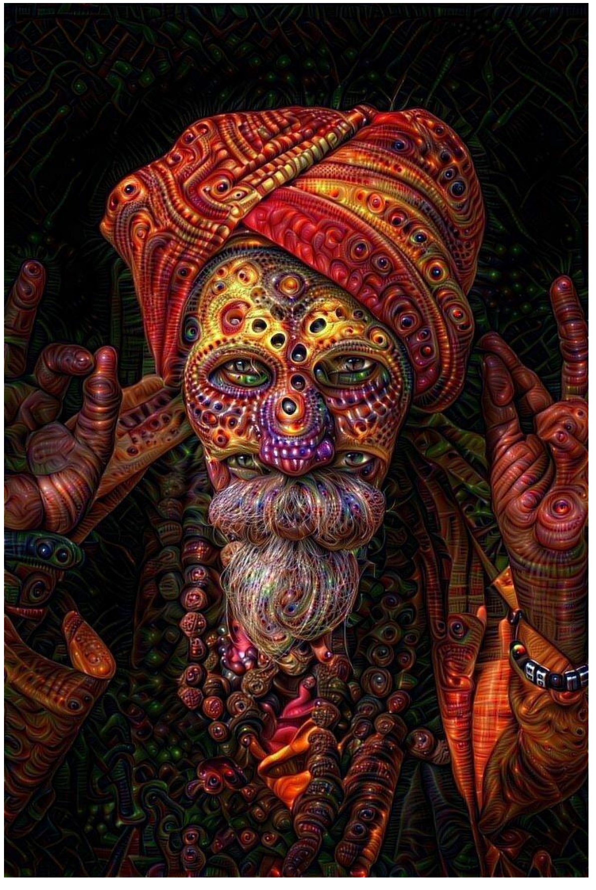
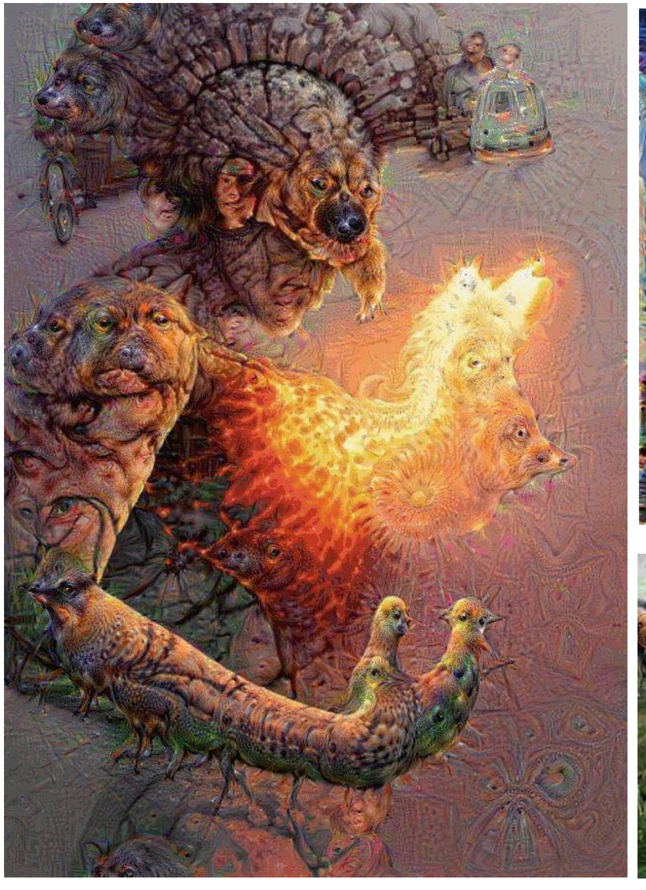
Understanding Deep Image Representations by Inverting Them
[Mahendran and Vedaldi, 2014]

Reconstructing images

Reconstructions from the representation after last last pooling layer (immediately before the first Fully Connected layer)

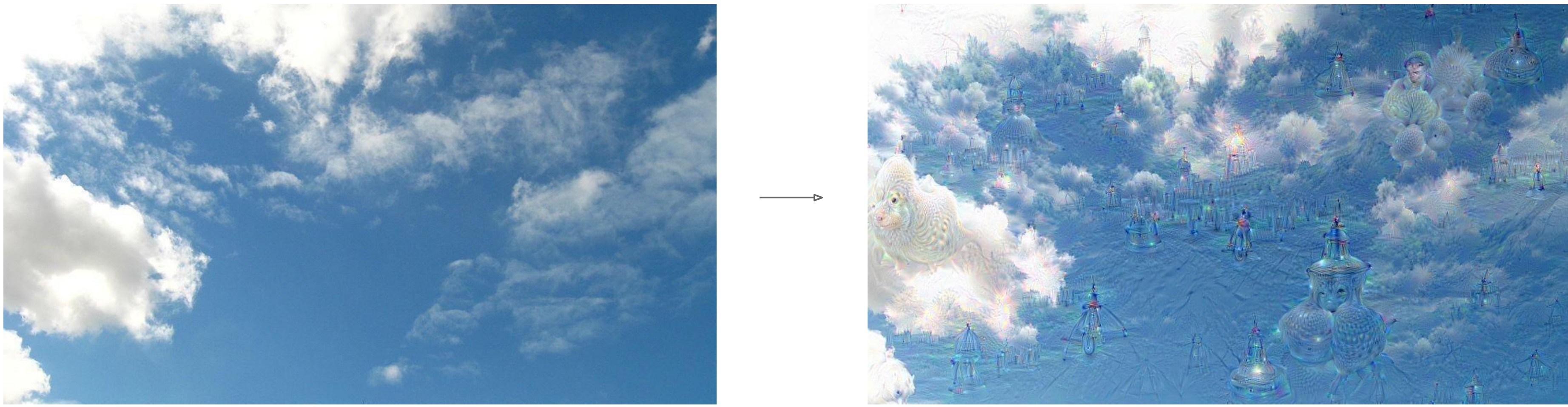
Andrej Karpathy

DeepDream

DeepDream https://github.com/google/deepdream

DeepDream



DeepDream modifies the image in a way that “boosts” all activations, at any layer

this creates a feedback loop: e.g. any slightly detected dog face will be made more and more dog like over time

"Admiral Dog!" "The Pig-Snail" "The Camel-Bird" "The Dog-Fish"

DeepDream

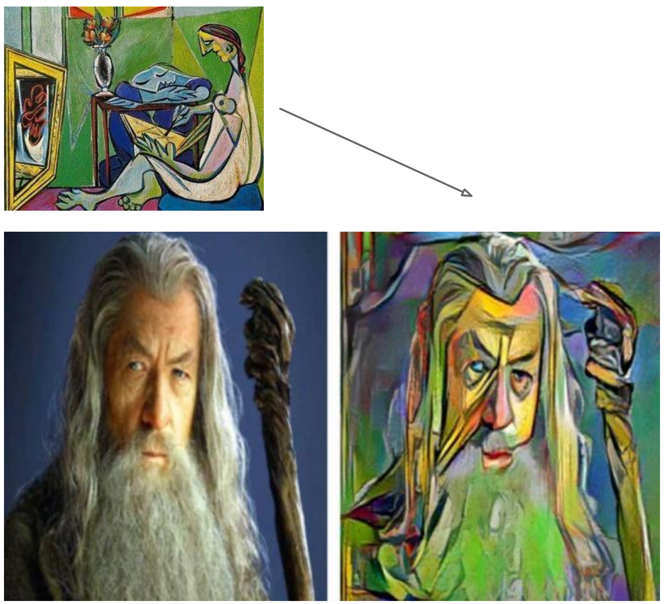
Deep Dream Grocery Trip
https://www.youtube.com/watch?v=DgPaCWJL7XI

Deep Dreaming Fear & Loathing in Las Vegas: the Great San Francisco Acid Wave
https://www.youtube.com/watch?v=oyxSerkkP4o

Synthesis / style transfer

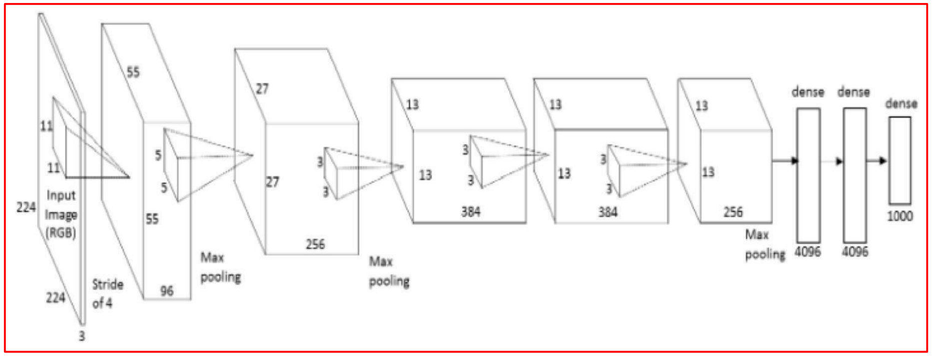
Neural Style

[A Neural Algorithm of Artistic Style by Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge, 2015]
good implementation by Justin Johnson in Torch:
https://github.com/jcjohnson/neural-style



Neural Style

Step 1: Extract **content targets** (ConvNet activations of all layers for the given content image)

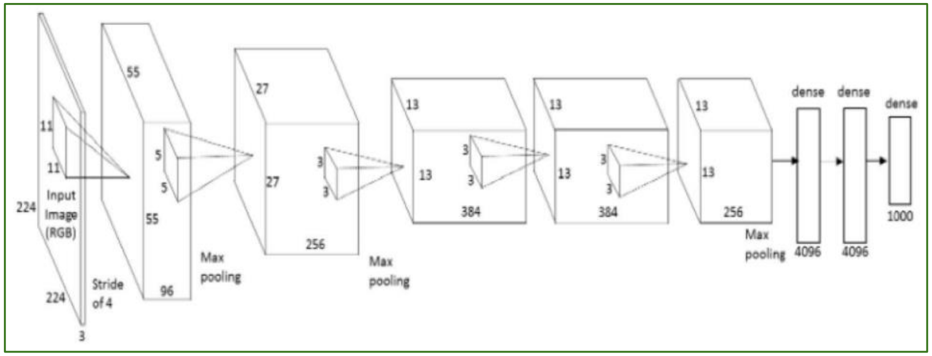


content activations

e.g.
at CONV5_1 layer we would have a [14x14x512] array of target activations

Neural Style

Step 2: Extract **style targets** (Gram matrices of ConvNet activations of all layers for the given style image)



style gram matrices

$$G = V^{\mathrm{T}} V$$

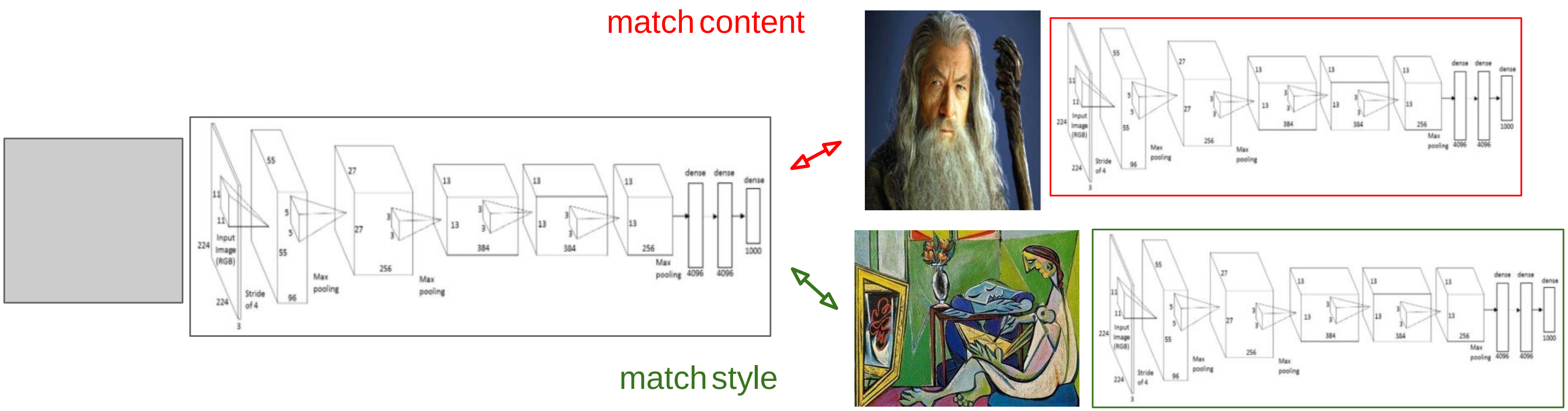
e.g.
at CONV1 layer (with [224x224x64] activations) would give a [64x64] Gram matrix of all pairwise activation covariances (summed across spatial locations)

Neural Style

Step 3: Optimize over image to have:

- The **content** of the content image (activations match content)
- The **style** of the style image (Gram matrices of activations match style)

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$



Adapted from Andrej Karpathy

Neural Style

make your own easily on deepart.io