Deep Learning Extensions

Lyle Ungar

University of Pennsylvania

Learning objectives

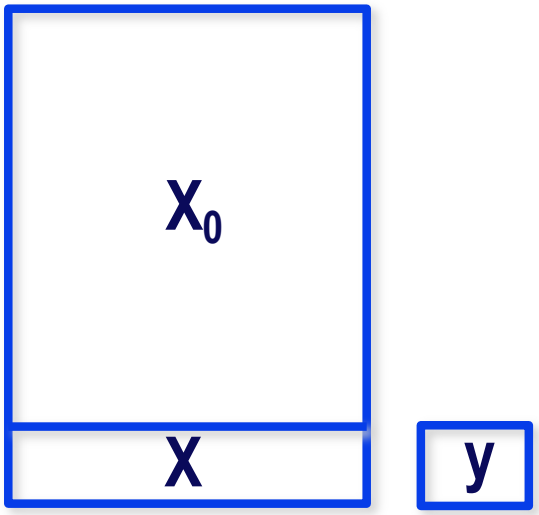
GANS

KL-divergence

CNNs for NLP

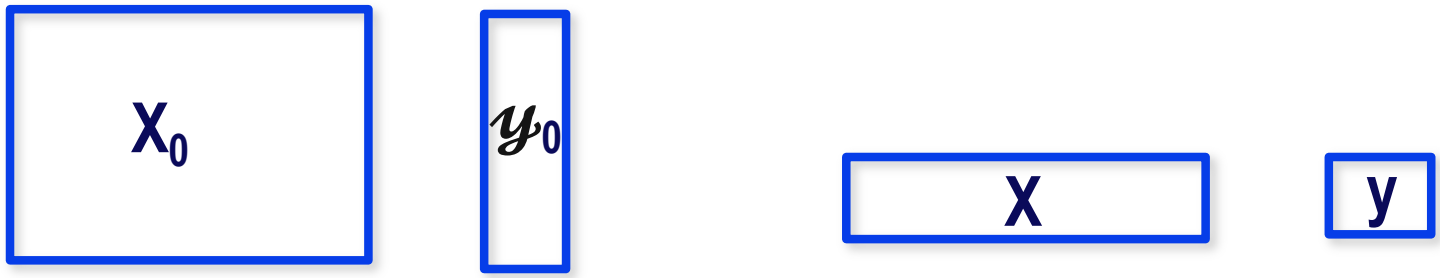
Semi-supervised learning

- Use unlabeled data X_0 to derive new features $z=\phi(\mathbf{x})$
- Train $y = f(\phi(\mathbf{x}), \mathbf{w})$



Transfer learning

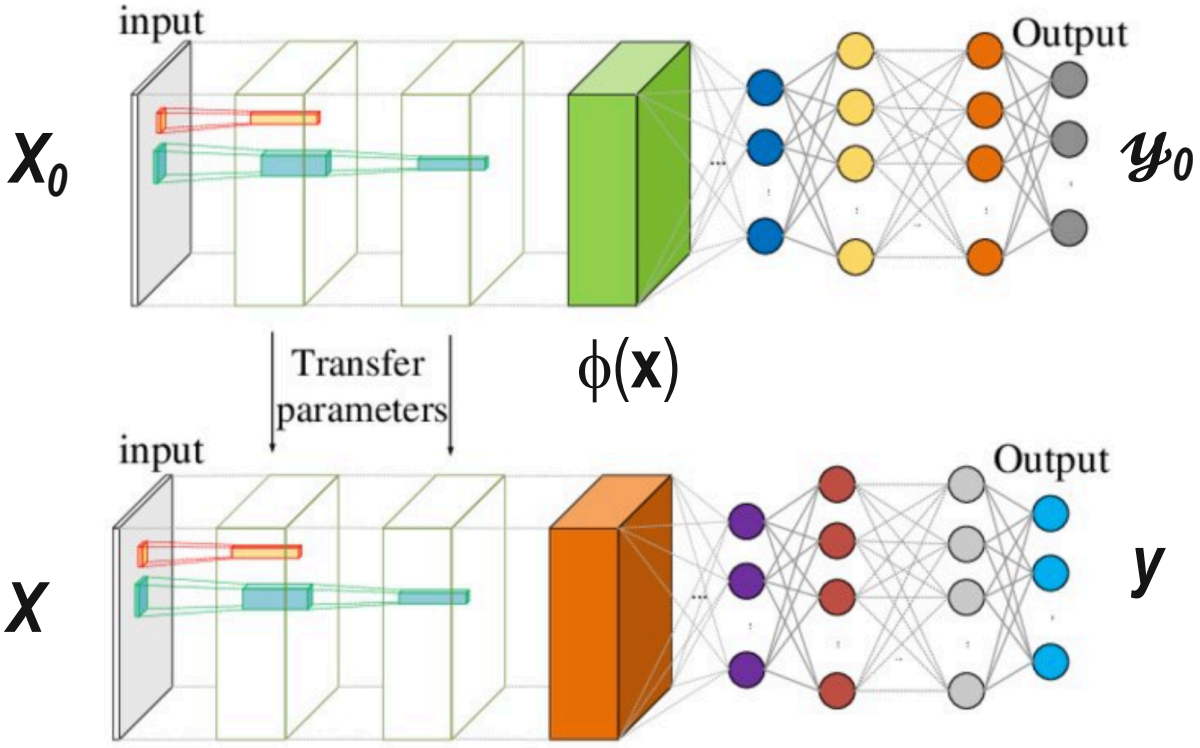
- Use one data set (X_0, y_0) to train a model
- Find feature transformations $\phi(x)$
- Use those transformations $\phi(x)$ on data from data set with a different label, y.



Data set for learning $\phi(x)$

Data set with target y.

Transfer learning for NNets

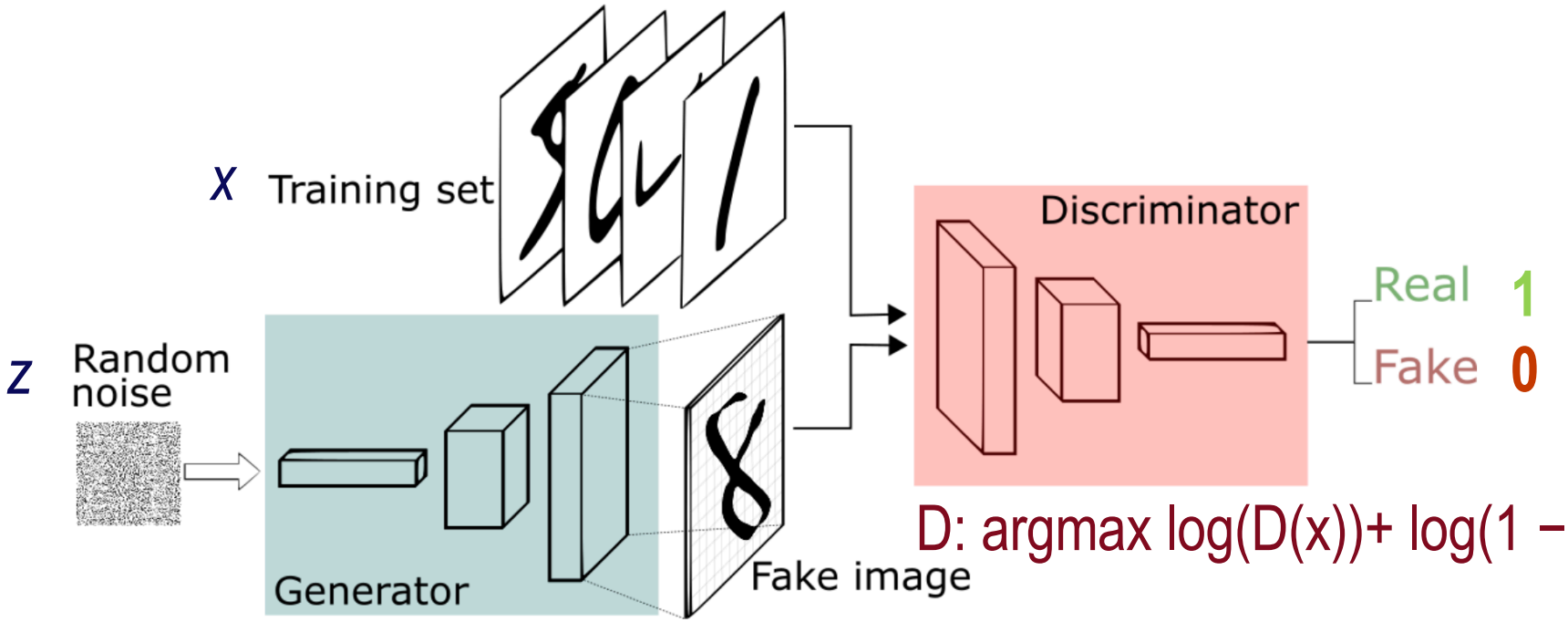


Data set for learning $\phi(x)$

Data set with target y.

Digression: KL-divergence

Generative Adversarial Networks: GANS



D: argmax log(D(x))+ log(1 − D(G(z)))

G: argmin log(1 − D(G(z)))

https://medium.freecodecamp.org/an-intuitive-introduction-to-

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

GANs

- **Make generative model, *G*, close to data PDF**

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}\left(p_{\mathrm{data}}(\boldsymbol{x}) \| p_{\mathrm{model}}(\boldsymbol{x}; \boldsymbol{\theta})\right) \quad = \quad \mathrm{E}\left(\sum p_{\mathrm{data}} \log\left(p_{\mathrm{model}}/p_{\mathrm{data}}\right)\right.$$

- **For actual data this is the MLE:**

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\mathrm{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)$$

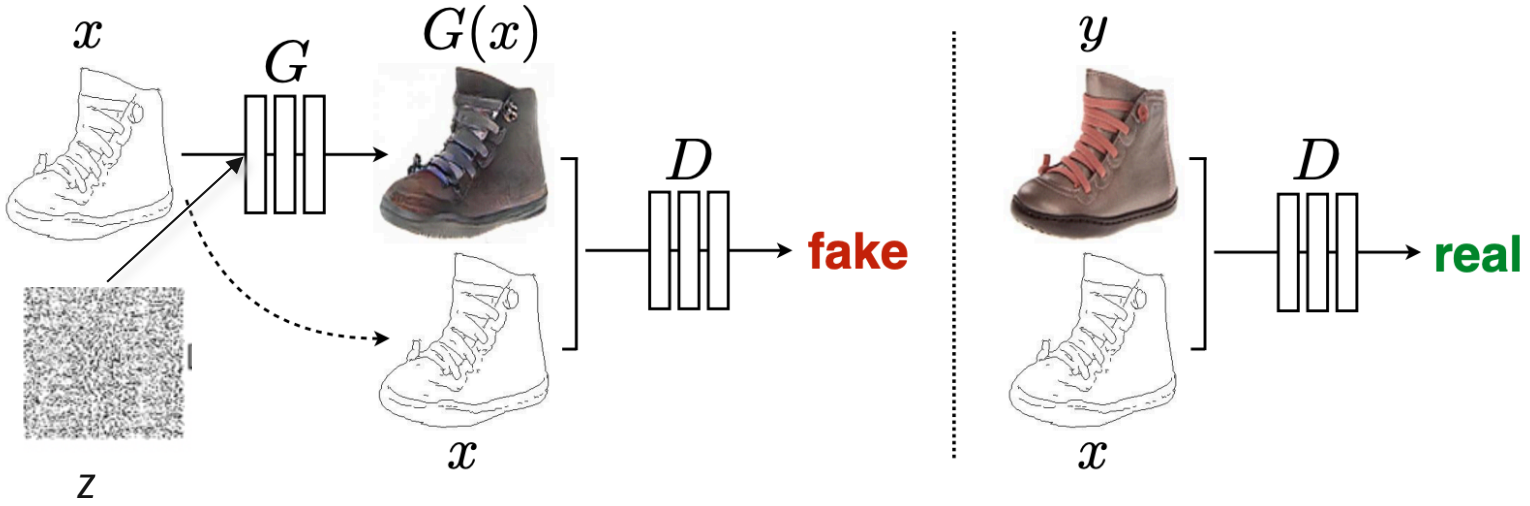
$$= \arg\max_{\boldsymbol{\theta}} \log \prod_{i=1}^{m} p_{\mathrm{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\mathrm{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right) \qquad \mathrm{E}\left(\sum p_{\mathrm{data}} \log\left(p_{\mathrm{model}}/p_{\mathrm{data}}\right)\right.$$

GANs Training

- **Training can be slow and unstable**
  - Better convergence if you alternately train discriminator and generator

Conditional GANS for Image Translation (outline to photo)



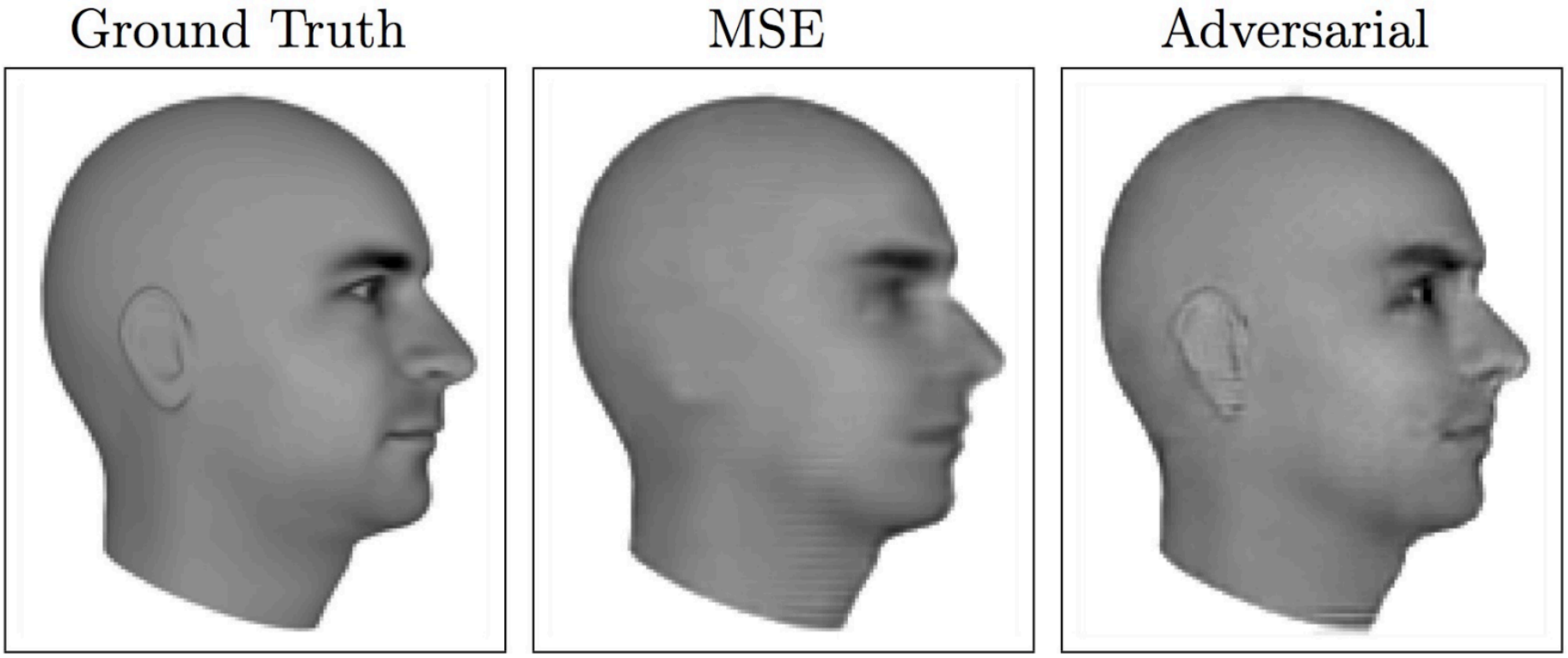
$$\min_G \max_D V(D,G) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))]$$

https://arxiv.org/pdf/1611.07004.pdf

GANs: Image translation

https://arxiv.org/pdf/1701.00160.pdf
Isola et al. (2016)

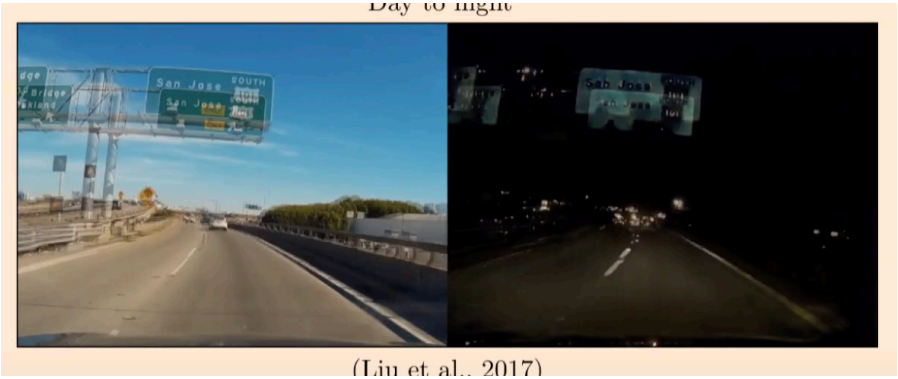
GANs: Predict next video frame



https://arxiv.org/pdf/1701.00160.pdf
Lotter et al. 2015

GANs: Image translation

- Summer to winter
- Sketch to photo
- low resolution to high resolution photo
- Young to old
- Photo to impressionist painting



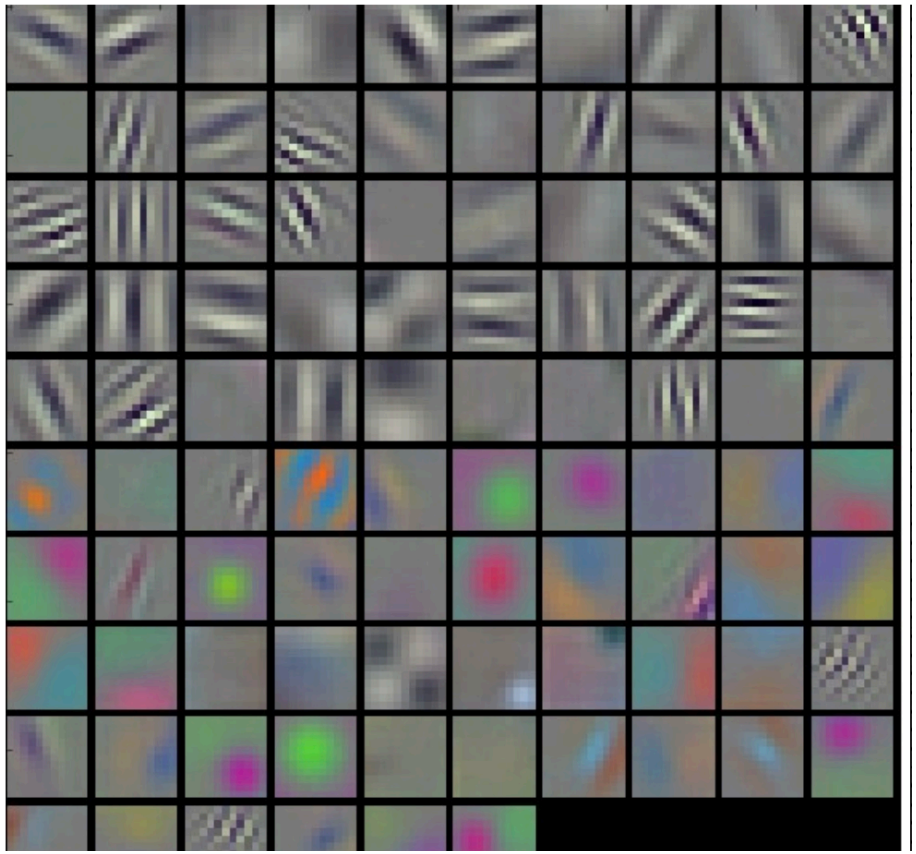
https://www.youtube.com/watch?time_continue=3704&v=AJJRWFVfNPg

Visualizing networks

- **Display pattern of hidden unit activations**
  - Just shows they are sparse
- **Show input that maximizes a node's output**
  - Over all inputs in the training set
  - Over the entire range of possible inputs
  - Early layers do feature detection
  - Later layers do object detection
- **Show how occluding parts of an image affect classification accuracy**

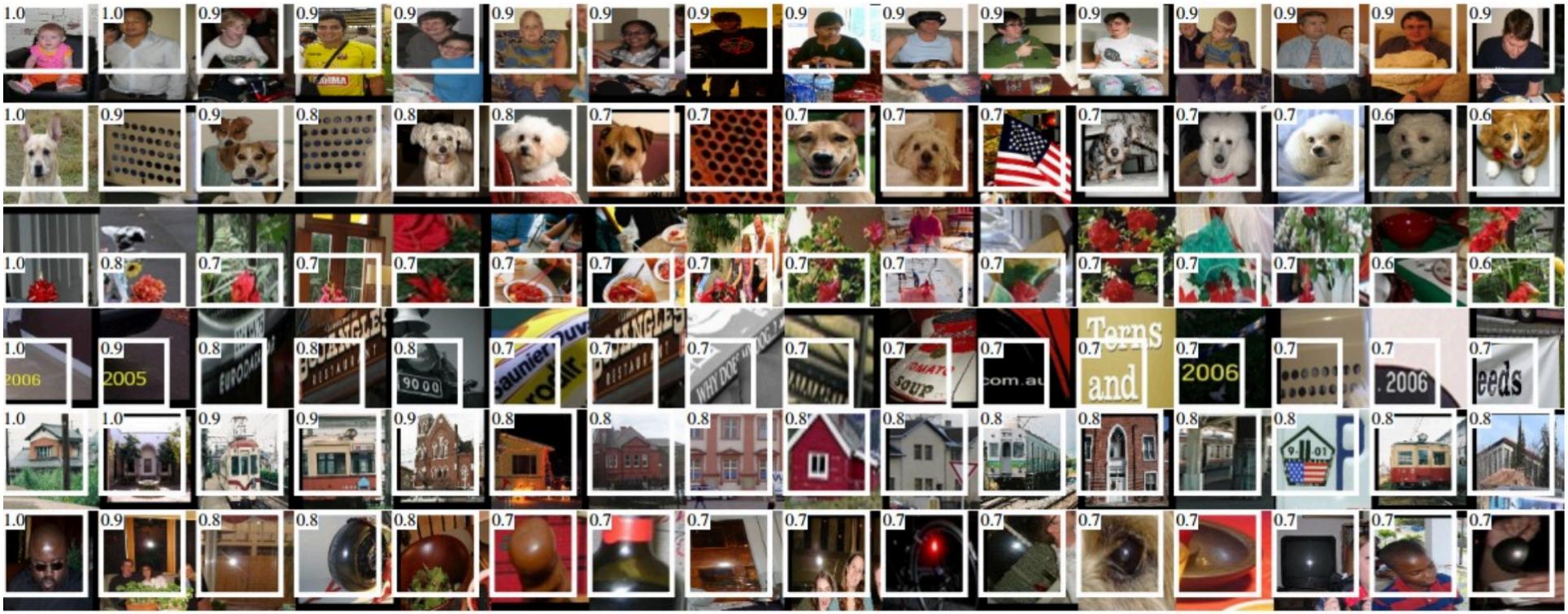
http://cs231n.github.io/understanding-cnn/

Maximally activating inputs for the first CONV layer of an AlexNet

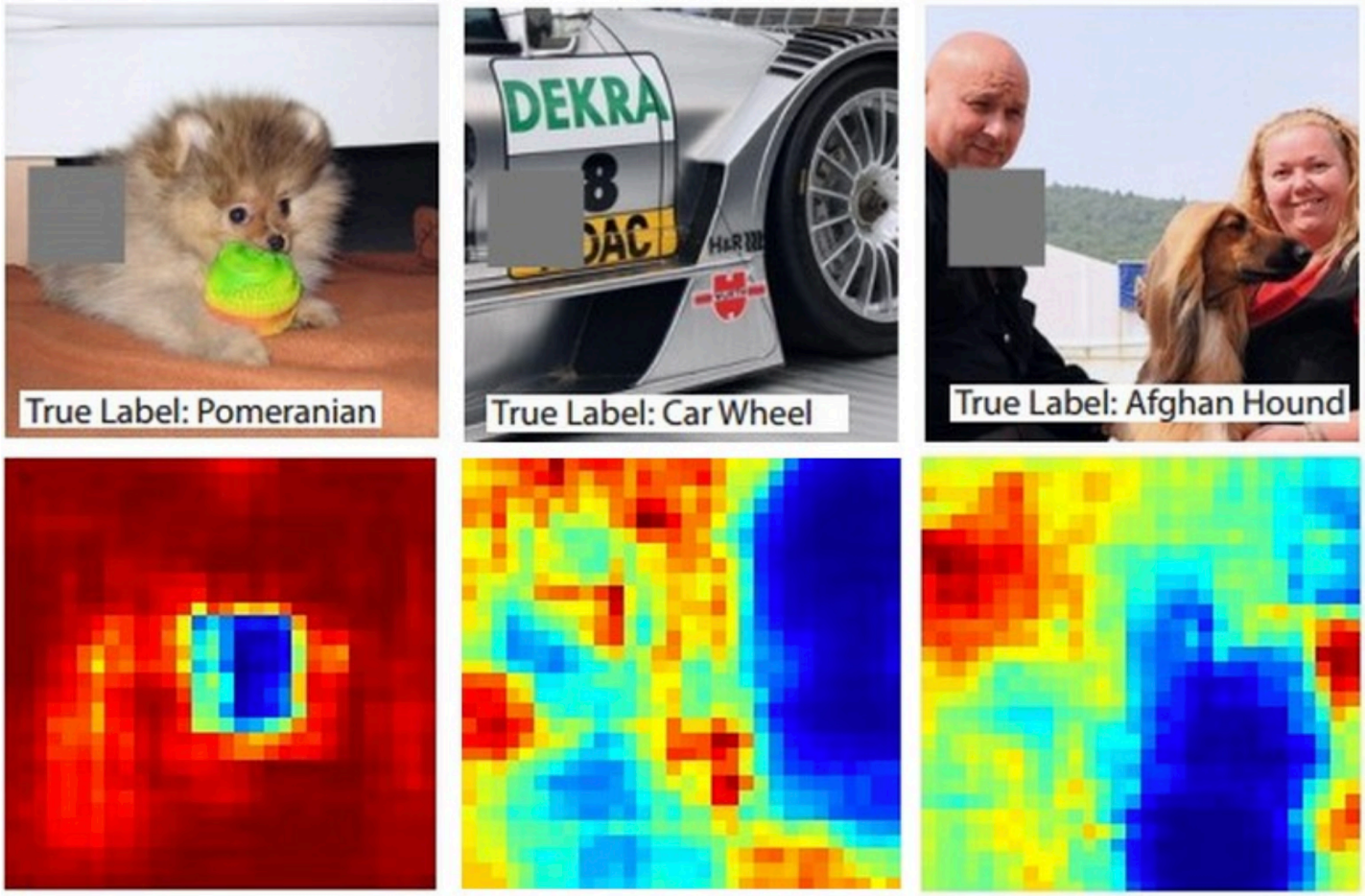


http://cs231n.github.io/understanding-cnn/

Maximally activating images for some 5th maxpool layer neurons of an AlexNet.



P(correct label) after occlusion



Matthew Zeiler's Visualizing and Understanding Convolutional Networks: