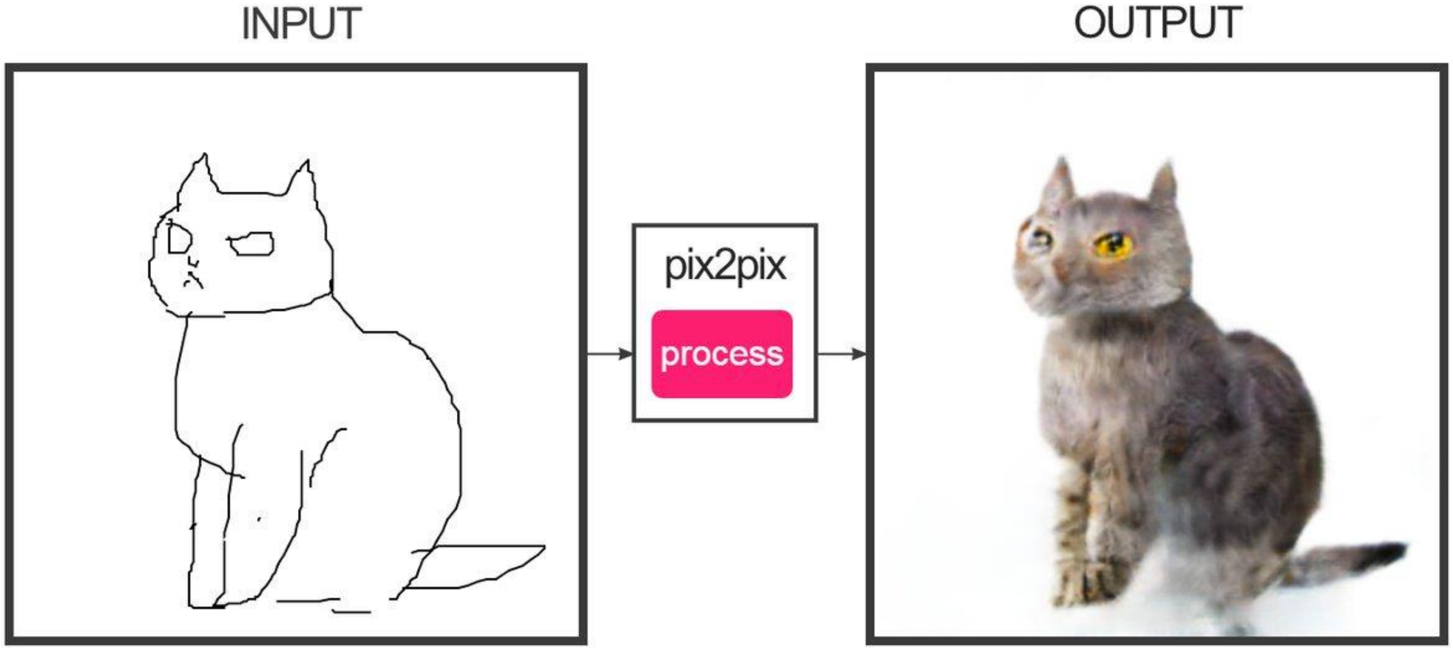
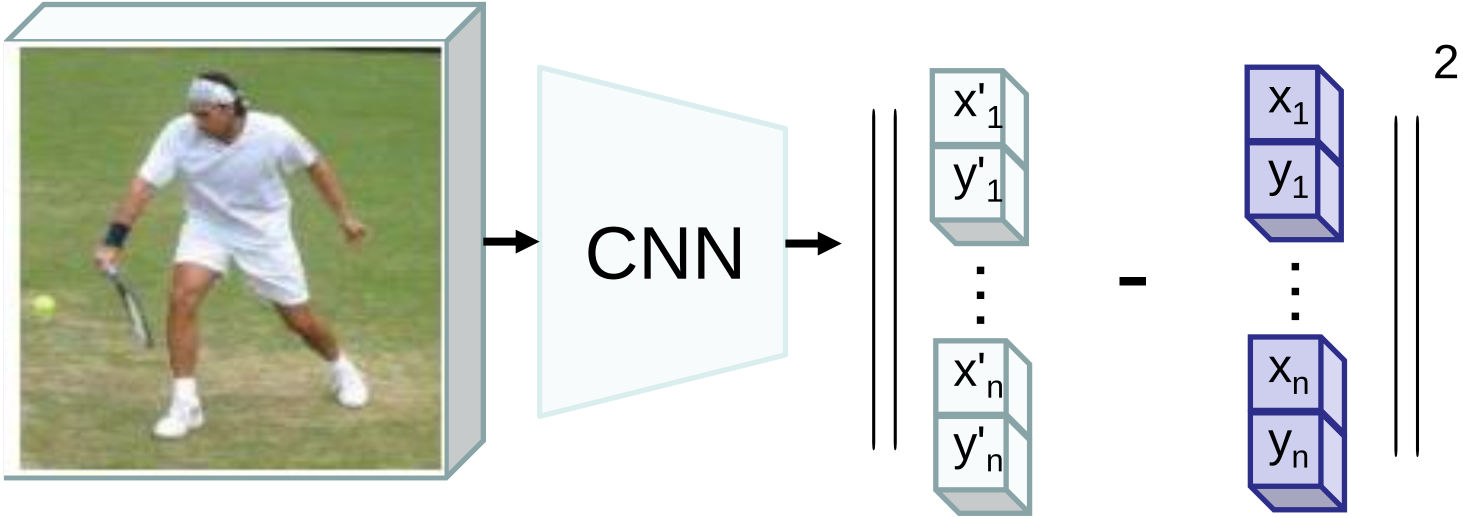
ConvNet for image-to-image tasks



Many slides from
Herr Prof. David Fouhey

CS194: Computer Vision and Comp. Photo
Alexei Efros, UC Berkeley, Fall 2022

Regression Objective

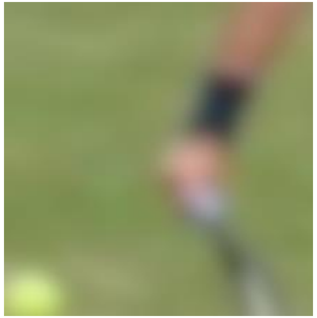


This is what you implement in part 1. But this is not what the SoTA models do in practice

Downsides of regression objective

Locally a lot of things look similar!!

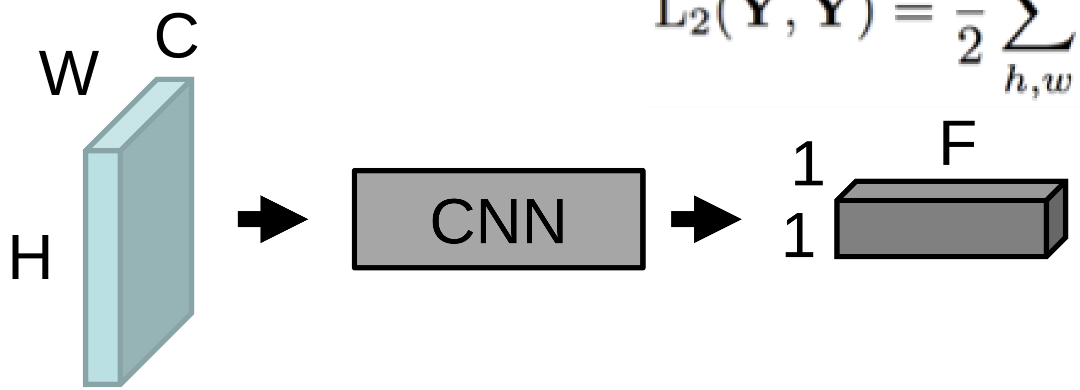


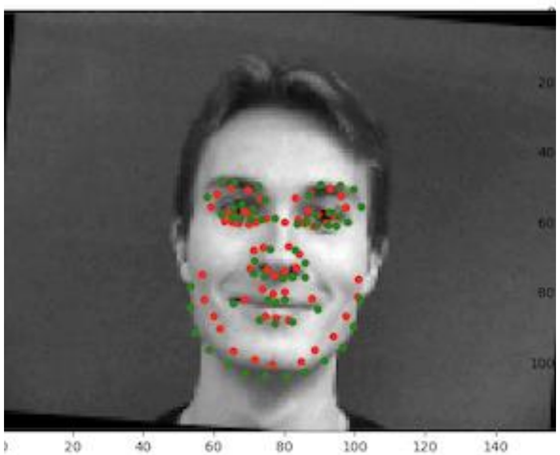


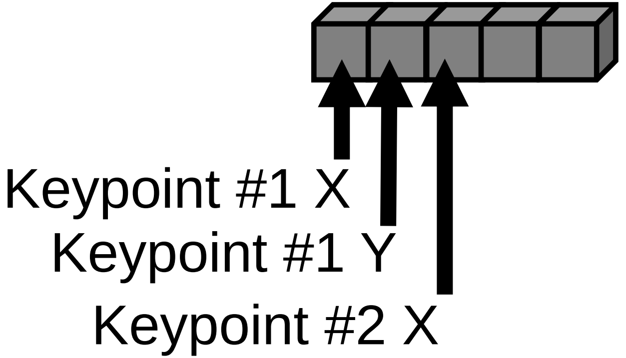
With regression objective you have to commit to ONE location and only get one training signal on how correct that location was.

Regression Loss (e.g. Part 1)

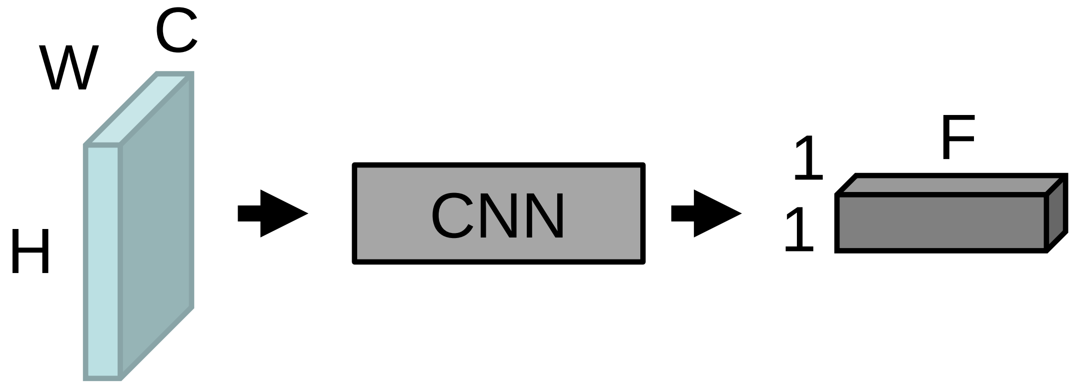
$$L_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

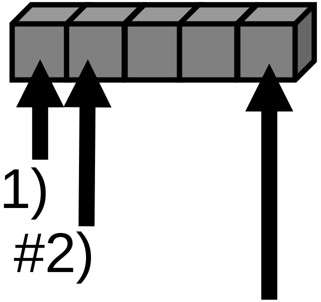






Classification Loss (e.g. ImageNet)

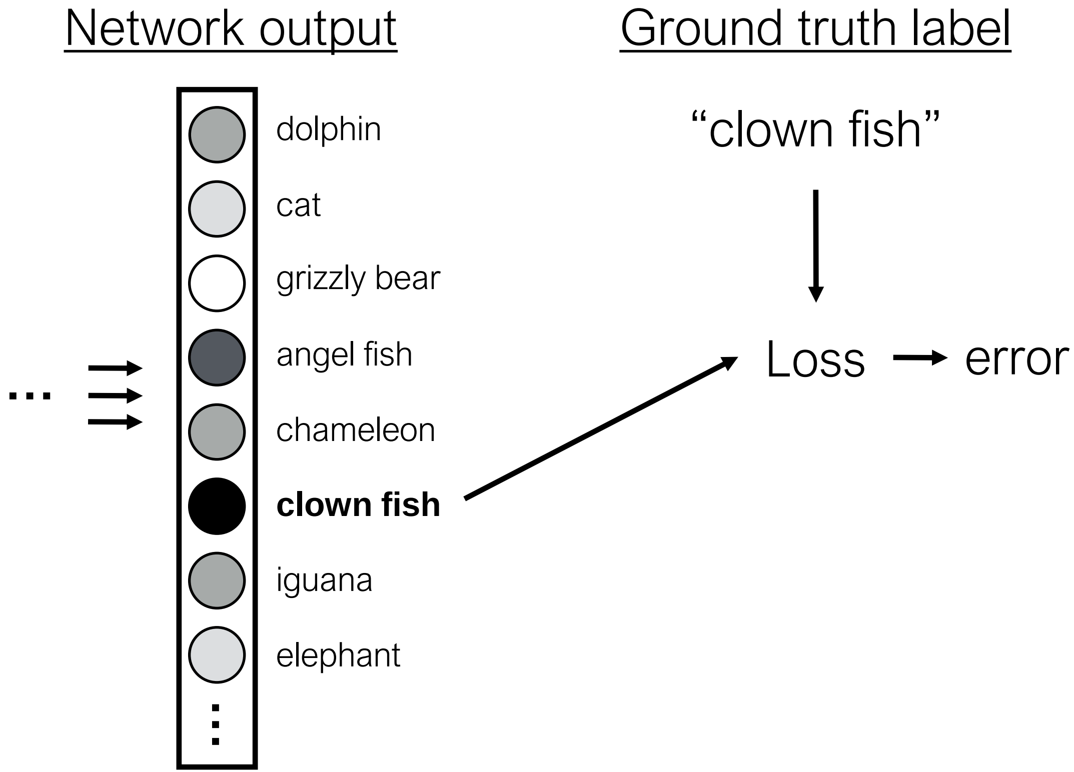




P(image is class #1)
P(image is class #2)
P(image is class #F)

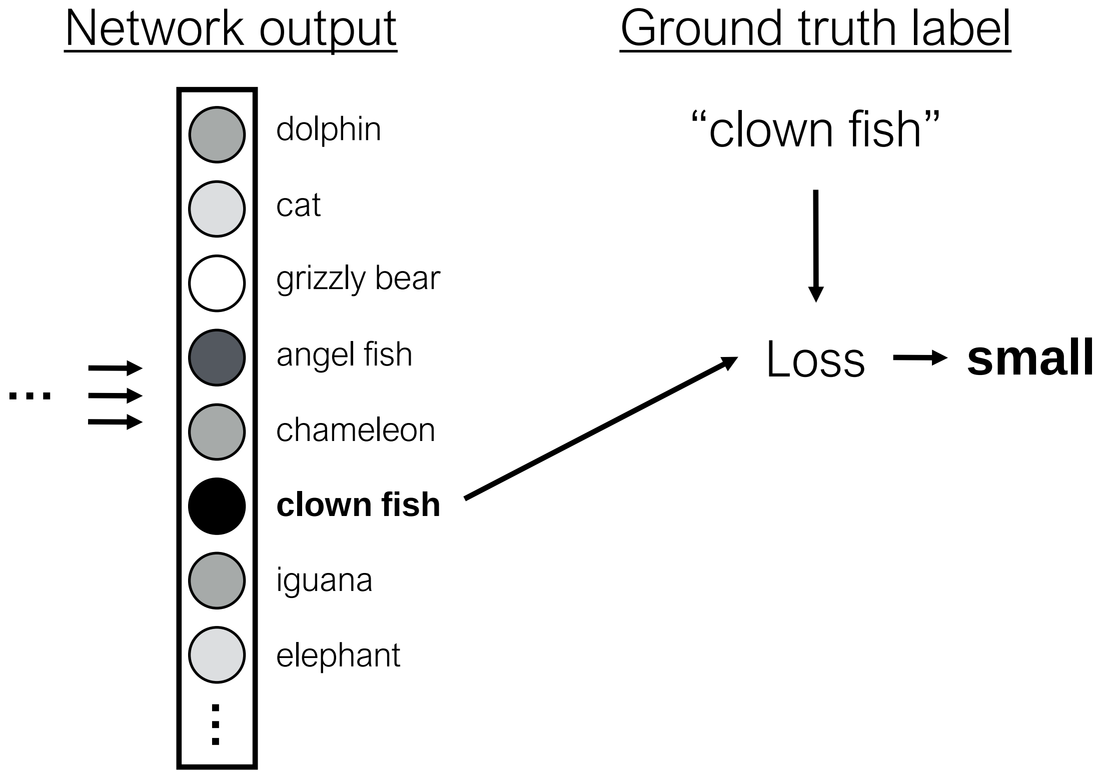
Slide by David Fouhey

Loss function for classification



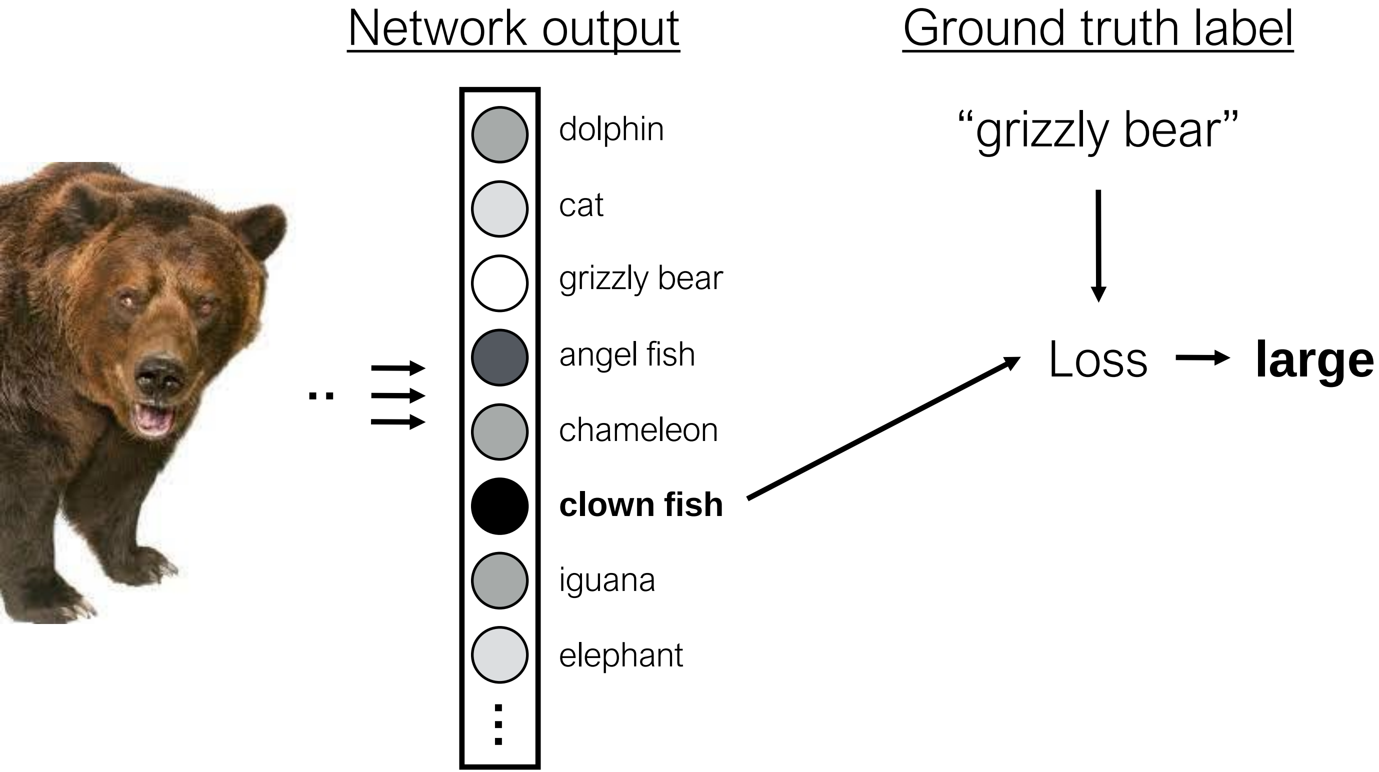
Slide by Philip Isola

Loss function for classification



Slide by Philip Isola

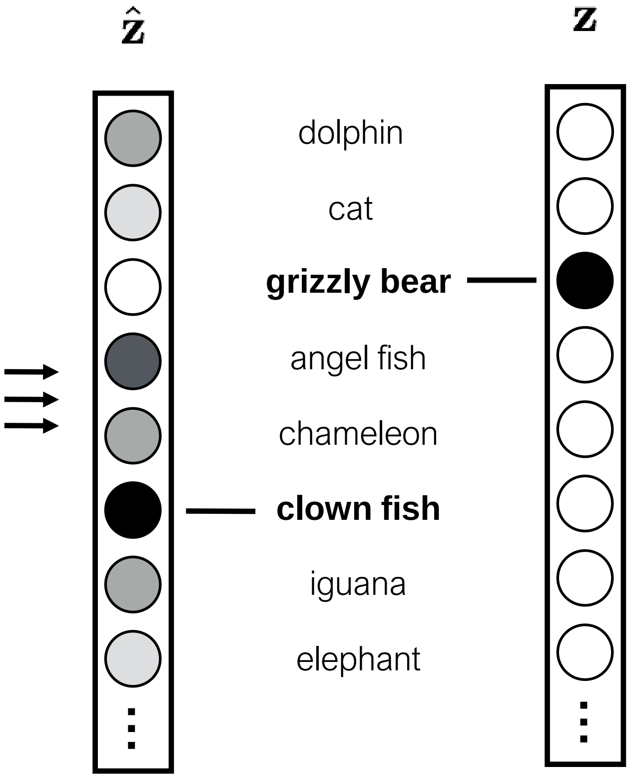
Loss function for classification



Slide by Philip Isola

Loss function for classification

<u>Network output</u> <u>Ground truth label</u>



Cross-entropy Loss: Probability of the observed data under the model

$$H(\hat{\mathbf{z}}, \mathbf{z}) = -\sum_c \hat{\mathbf{z}}_c \log \mathbf{z}_c$$

Results in learning a probability model $p(c|\mathbf{x})$

Slide by Philip Isola

Belief/Confidence map formulation

[Thompson et al. NeurIPS 2014, CVPR 2015, Convolutional Pose Machine, Wei et al. CVPR 2016, (figure credit: Ning et al TMM 2017)…]

For K keypoints, train model to predict K many sheets (h x w) of scores of how likely the pixel is k-th keypoint

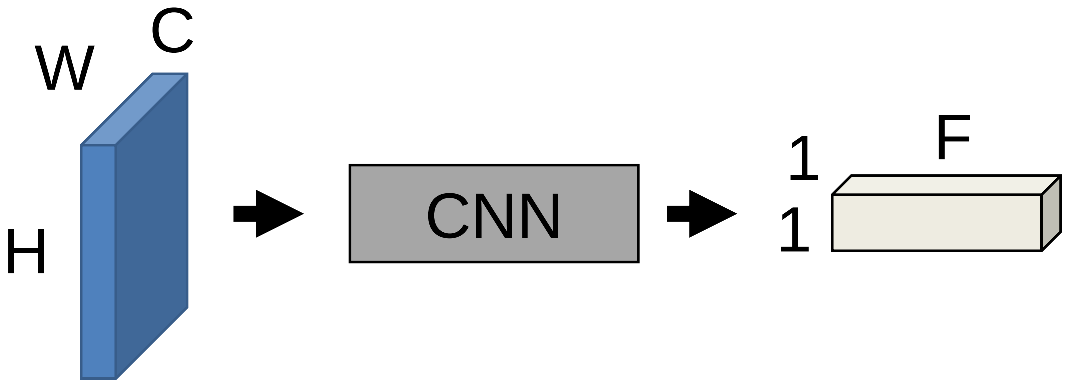
Problem

So far, we've only seen examples that output a vector representation out of an image.

How do we do dense (per-pixel) predictions?

Slide by David Fouhey

So Far…

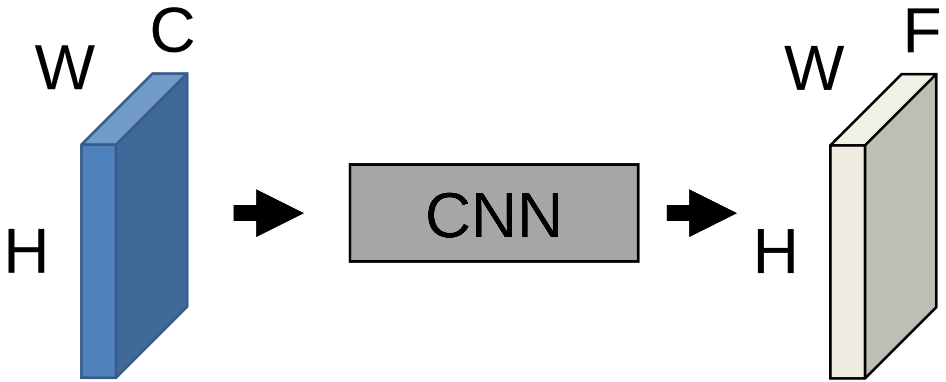


Convert HxW image into a F-dimensional vector

Is this image a cat?
At what distance was this photo taken?
Is this image fake?

Slide by David Fouhey

Pixel Labeling



Convert HxW image into a F-dimensional vector

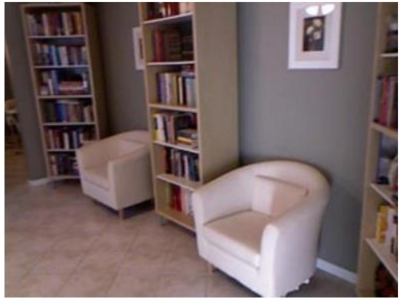
Which pixels in this image are a cat?
How far is each pixel away from the camera?
Which pixels of this image are fake?

Slide by David Fouhey

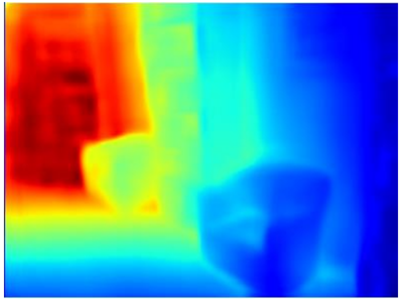
e.g. Depth Prediction

Instead: give label of depthmap, train network to do regression (e.g., $||z_i - \hat{z_i}||$ where z_i is the ground-truth and $\hat{z_i}$ the prediction of the network at pixel i).

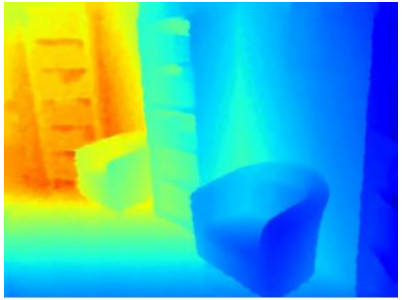
Input HxWx3
RGB Image



Output HxWx1
Depth Image



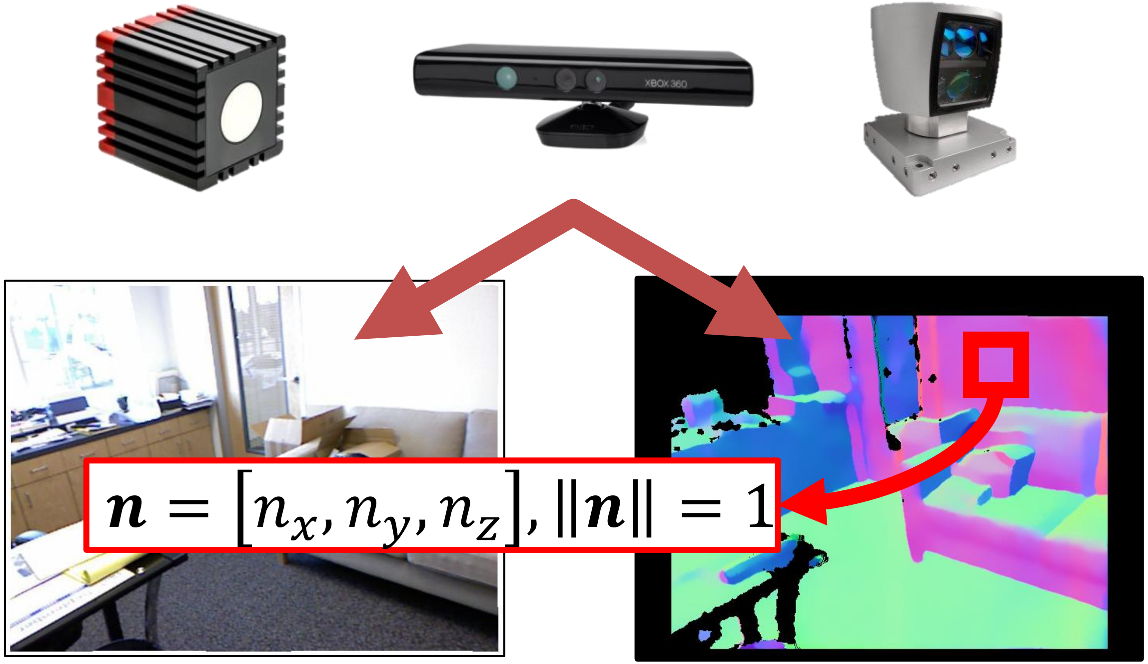
True HxWx1
Depth Image



Result credit: Eigen and Fergus, ICCV 2015

Slide by David Fouhey

Surface Normals

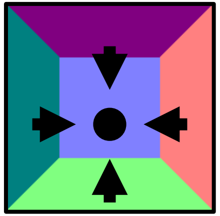


$$\boldsymbol{n} = [n_x, n_y, n_z], \|\boldsymbol{n}\| = 1$$

Color Image

Normals

Room



Legend

Image credit: NYU Dataset, Silberman et al. ECCV 2012

Slide by David Fouhey

Surface Normals

Instead: train normal network to minimize $||\boldsymbol{n}_i - \widehat{\boldsymbol{n}_i}||$
where $\boldsymbol{n}_i$ is ground-truth and $\widehat{\boldsymbol{n}_i}$ prediction at pixel i.

Input: HxWx3
RGB Image

Output: HxWx3
Normals

Result credit: X. Wang, D. Fouhey, A. Gupta, *Designing Deep Networks for Surface Normal Estimation.* CVPR 2014

Slide by David Fouhey

“Semantic Segmentation”

Each pixel has label, inc. **background**, and unknown

Usually visualized by colors.

Note: don’t distinguish between object *instances*

Input

Label

Input

Label

Image Credit: Everingham et al. Pascal VOC 2012.

Slide by David Fouhey

“Semantic Segmentation”

“Semantic”: a usually meaningless word.
Meant to indicate here that we’re **naming** things.

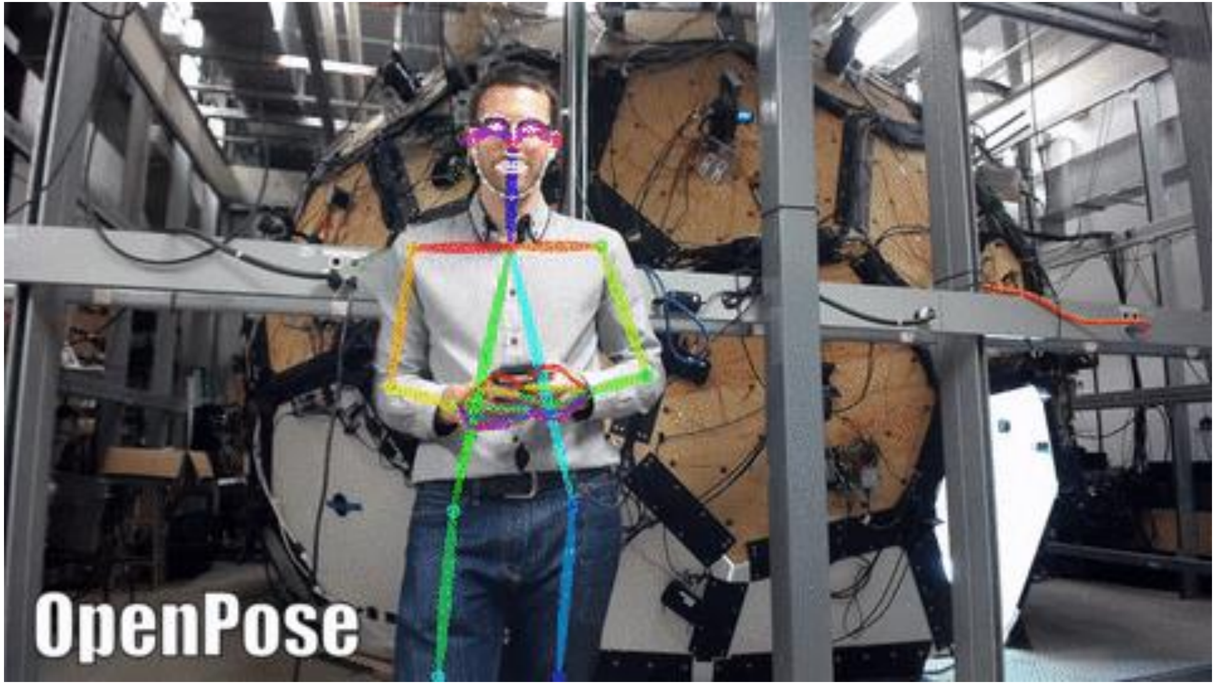
Input | Label | Input | Label




Slide by David Fouhey

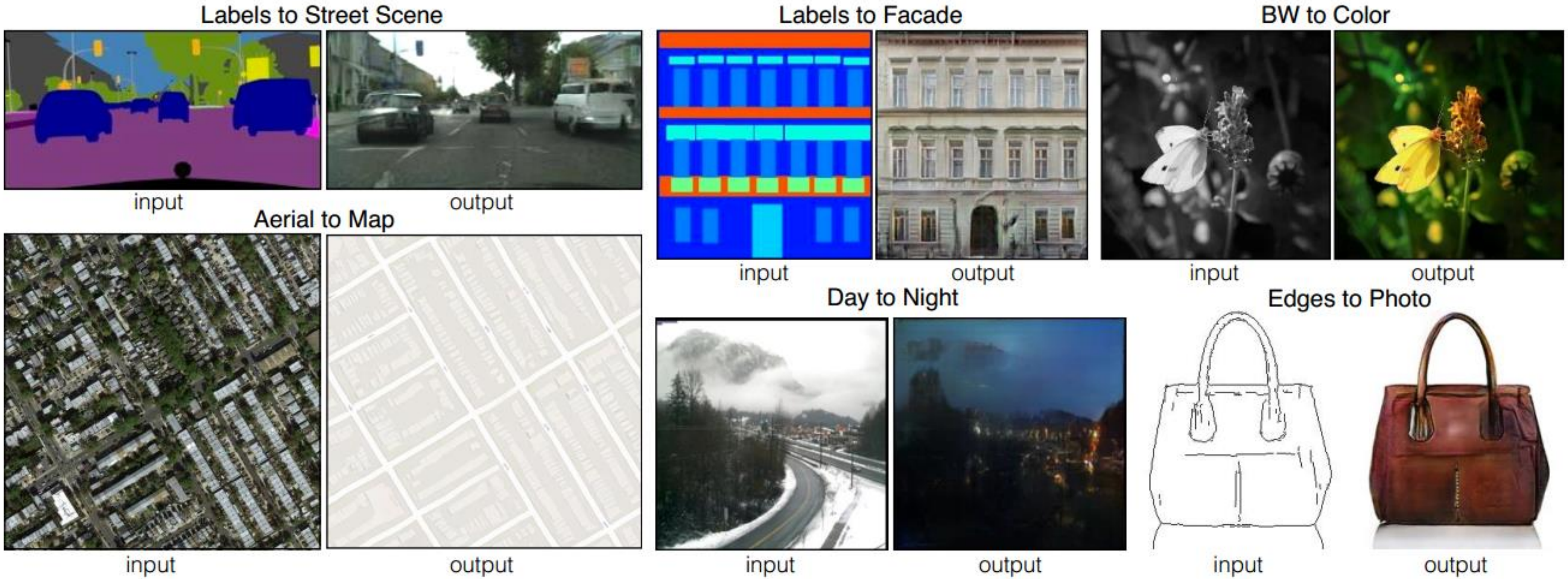
OpenPose

Great opensource tool, builds on convolutional pose machine architecture, adapted to multiple people



Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, Yaser Sheikh ‘16-17

Generic Image-to-Image Translation

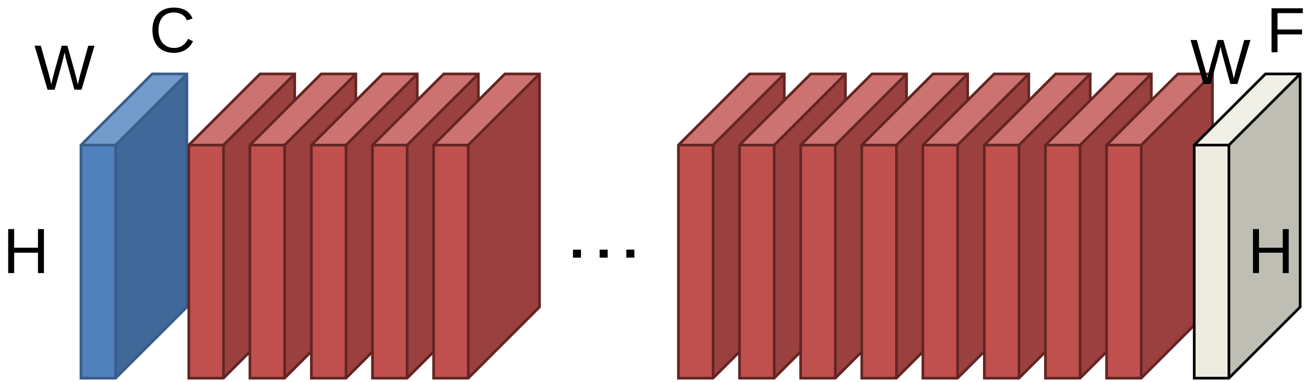


First – Two “Wrong” Ways

- It’s helpful to see two “wrong” ways to do this.

Slide by David Fouhey

Why Not Stack Convolutions?



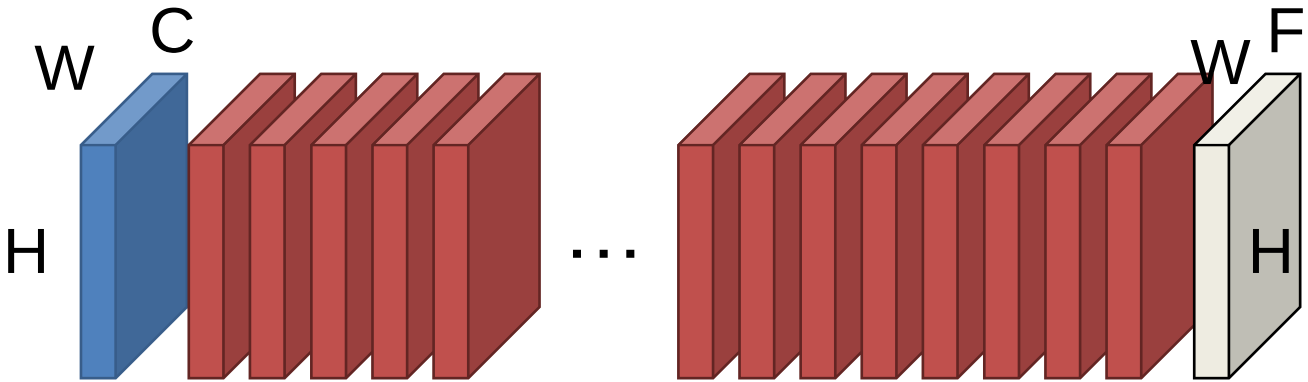
n 3x3 convs have a receptive field of 2n+1 pixels

How many convolutions until >=200 pixels?

100

Slide by David Fouhey

Why Not Stack Convolutions?



Suppose 200 3x3 filters/layer, H=W=400

Storage/layer/image: 200 * 400 * 400 * 4 bytes = 122MB

Uh oh!*

*100 layers, batch size of 20 = 238GB of memory!

Slide by David Fouhey

Idea #2

Crop out every sub-window and predict the label in the middle.

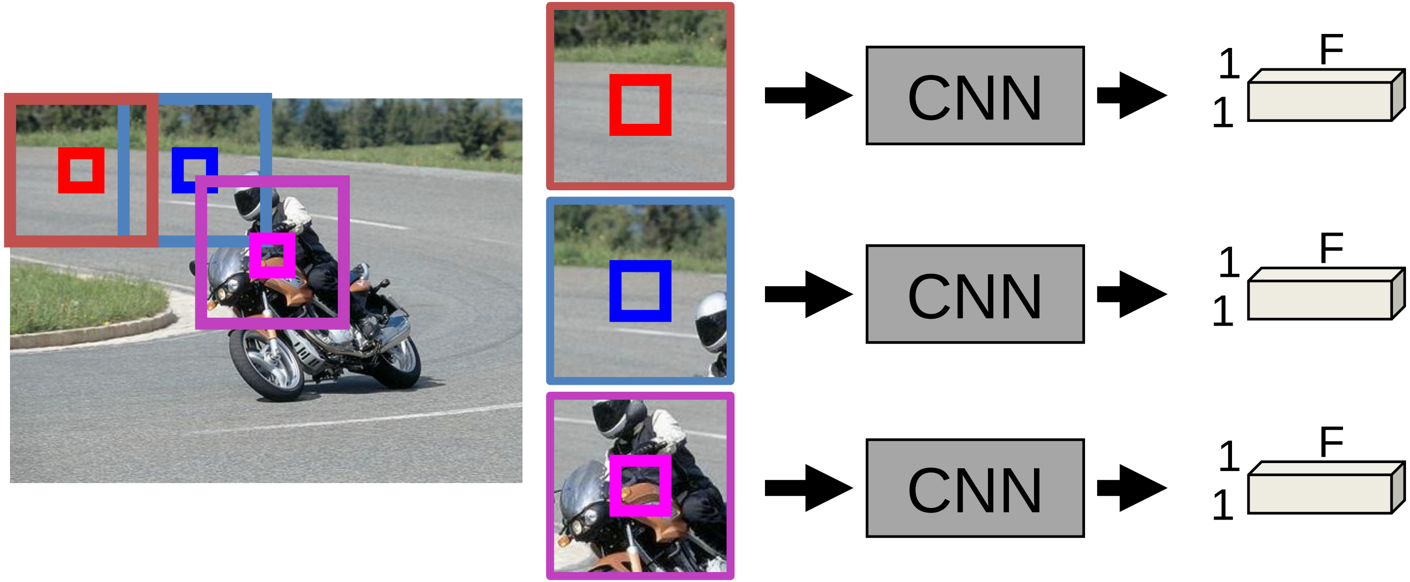


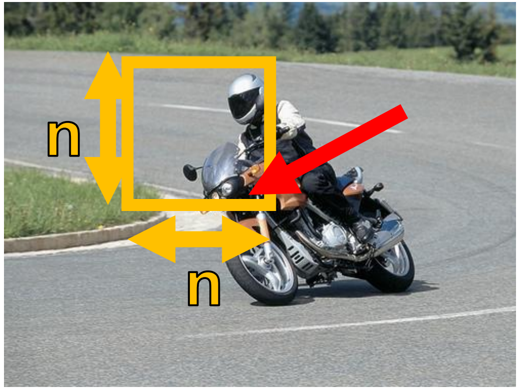
Image credit: PASCAL VOC, Everingham et al.

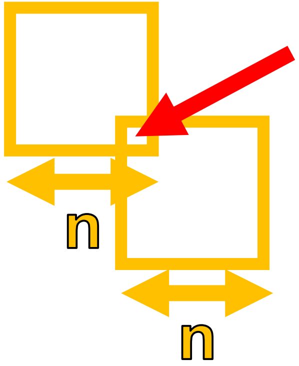
Slide by David Fouhey

Idea #2

Meet “Gabor”. We extract NxN patches and do independent CNNs. **How many times does Gabor filter the red pixel?**

Gabor





Answer:
(2n-1)*(2n-1)

Image credit: PASCAL VOC, Everingham et al.

Slide by David Fouhey

The Big Issue

We need to:

1. Have large receptive fields to figure out what we're looking at
2. Not waste a ton of time or memory while doing so

These two objectives are in total conflict

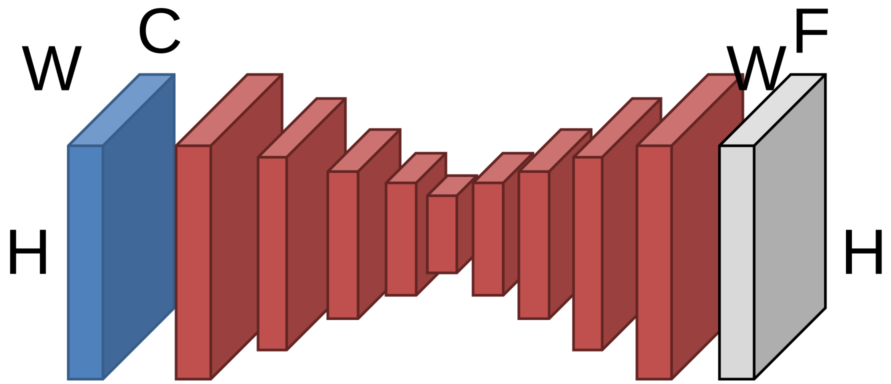
Slide by David Fouhey

Encoder-Decoder

Key idea: First **downsample** towards middle of network. Then **upsample** from middle.

How do we downsample?

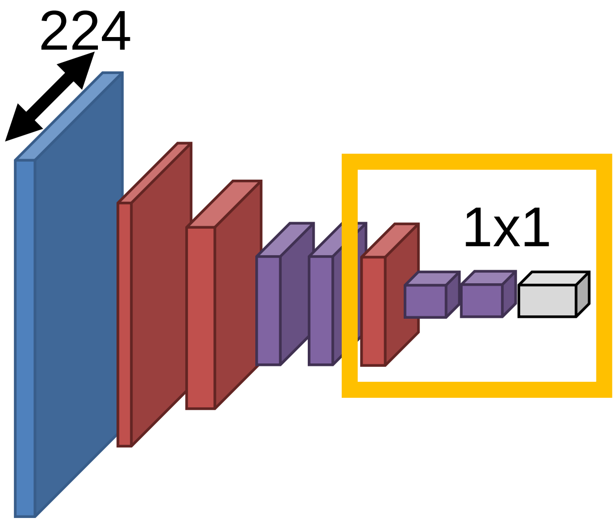
Convolutions, pooling

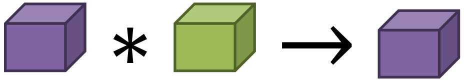


Slide by David Fouhey

Where Do We Get Parameters?

Convnet that maps images to vectors





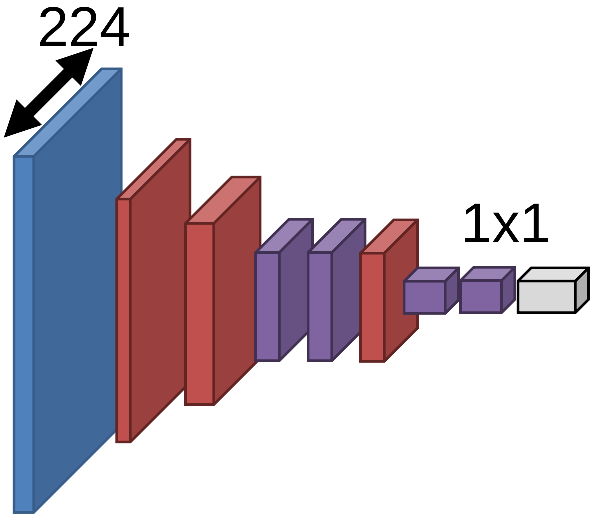
Recall that we can rewrite any vector-vector operations via 1x1 convolutions

Long et al. *Fully Convolutional Networks For Semantic Segmentation*. CVPR 2014

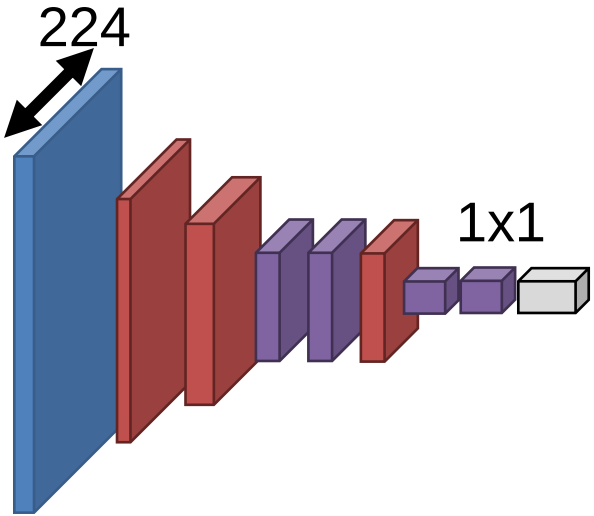
Slide by David Fouhey

Where Do We Get Parameters?

Convnet that maps images to vectors



Convnet that maps images to images

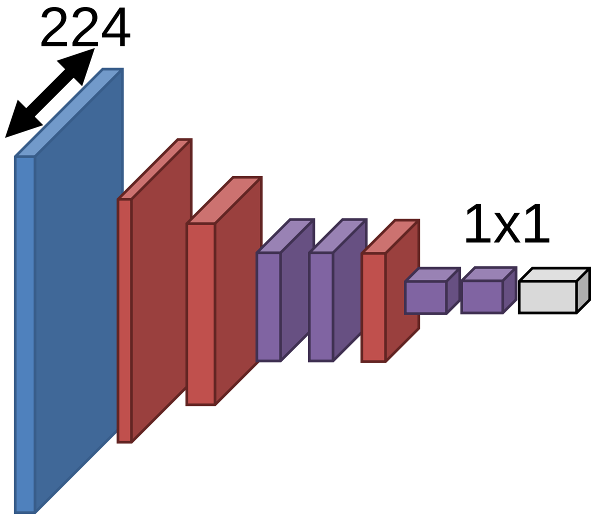


What if we make the input bigger?

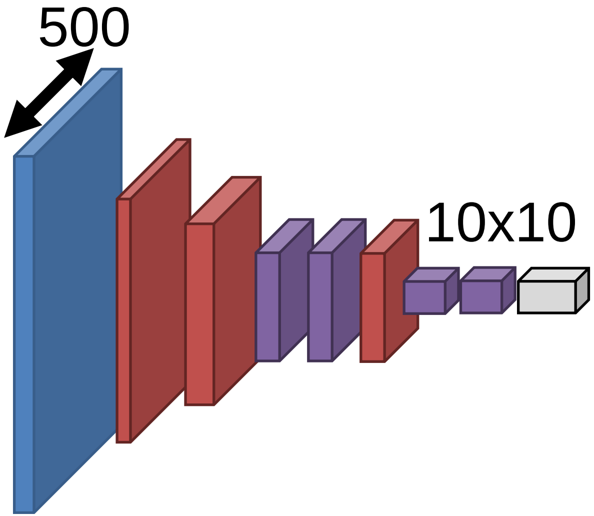
Slide by David Fouhey

Where Do We Get Parameters?

Convnet that maps images to vectors



Convnet that maps images to images

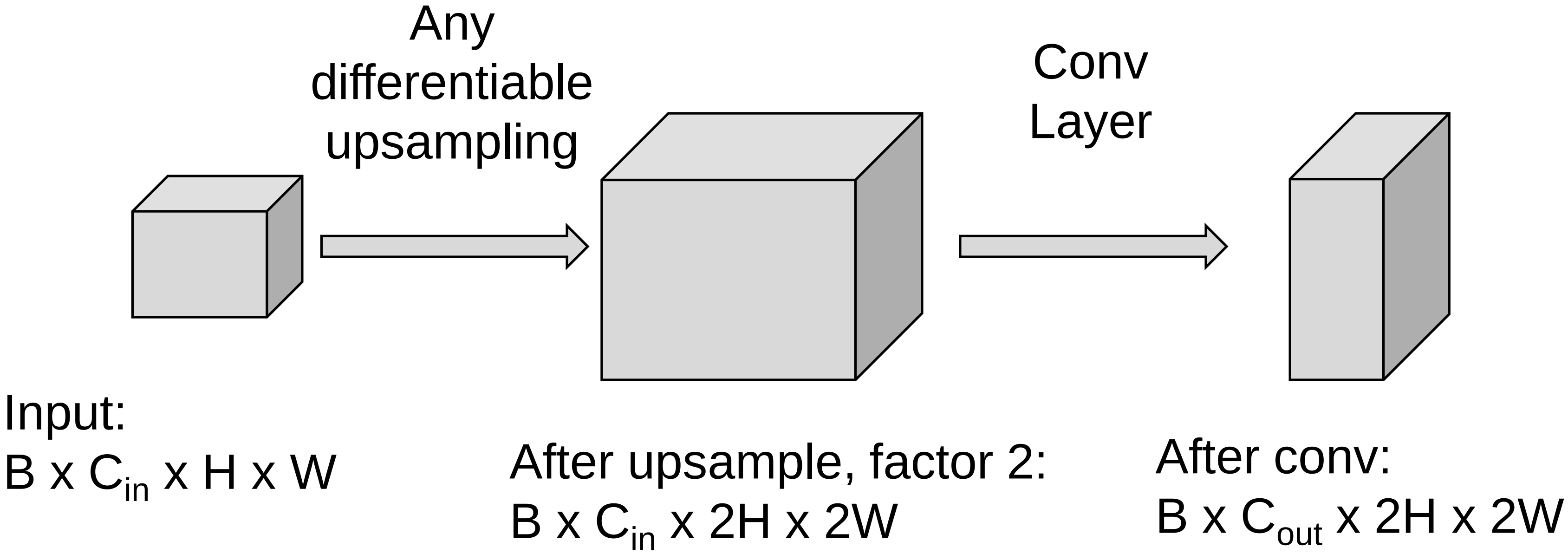


Slide by David Fouhey

How to upsample with convnets?

Simple solution

- Upsample, followed by a regular Convolution



How to Upsample

Bed of Nails

1	2
3	4

Input
C x 2 x 2

1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

Output
C x 4 x 4

Nearest Neighbor

1	2
3	4

Input
C x 2 x 2

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Output
C x 4 x 4

Bilinear Interpolation

1	2
3	4

Input
C x 2 x 2

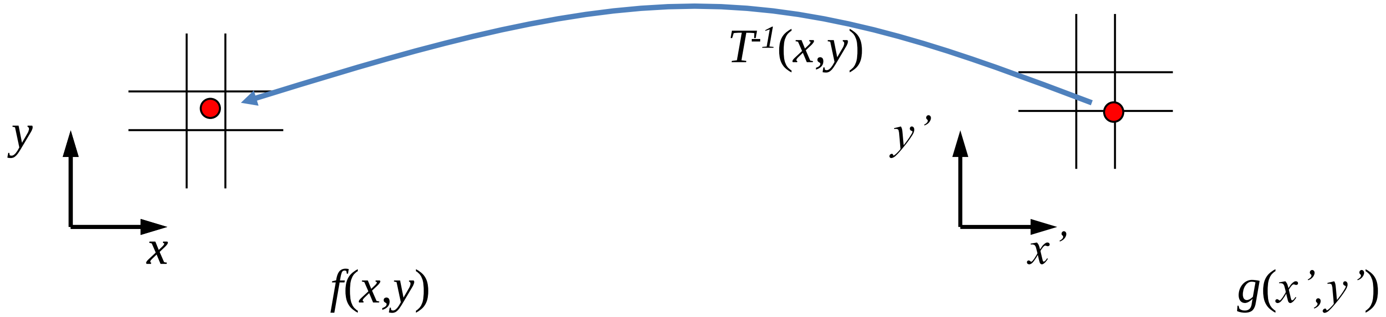
1.00	1.25	1.75	2.00
1.50	1.75	2.25	2.50
2.50	2.75	3.25	3.50
3.00	3.25	3.75	4.00

Output
C x 4 x 4

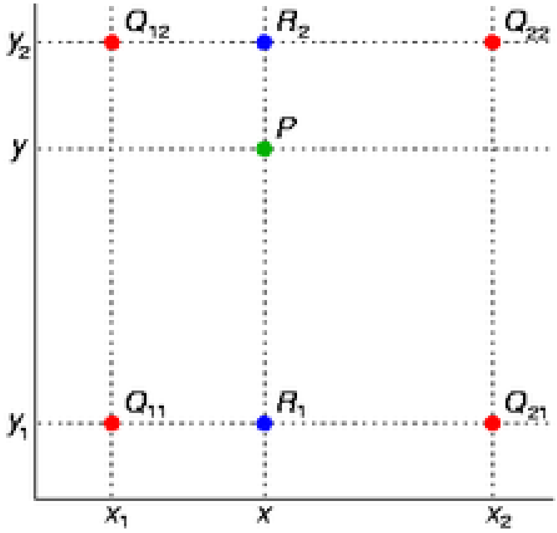
Recall from Morphing Lecture: Inverse warping

Don't splat! Do inverse warping.

You know this from project 3



Recall: Bilinear Interpolation

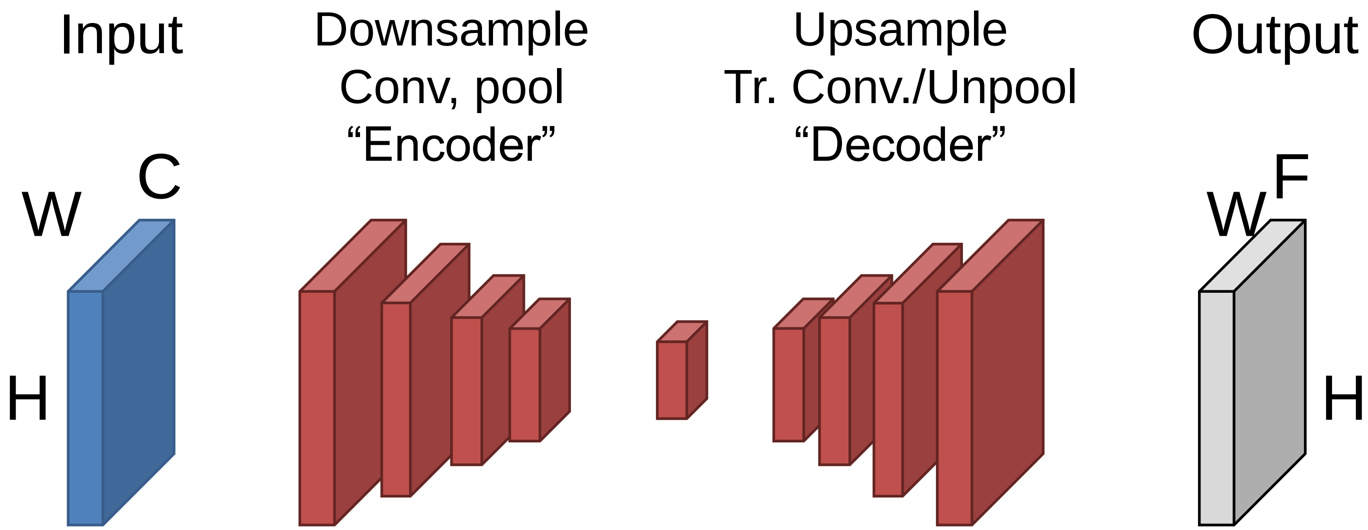


http://en.wikipedia.org/wiki/Bilinear_interpolation

`Help interp2`

Putting it Together

Convolutions + pooling downsample/compress/encode
Transpose convs./unpoolings upsample/uncompress/decode

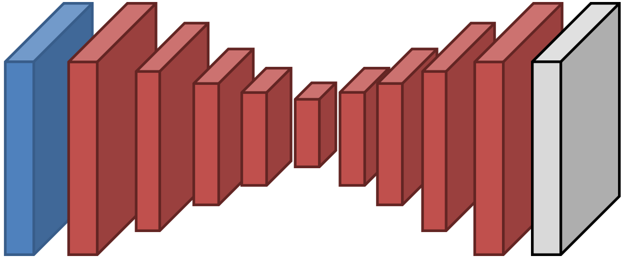


Slide by David Fouhey

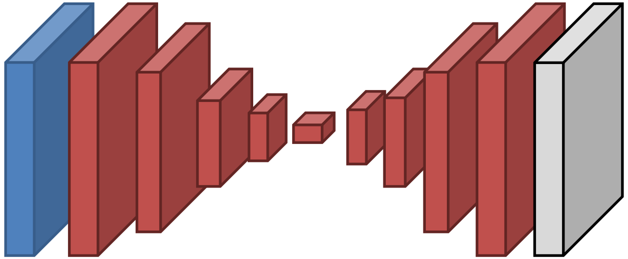
Putting It Together – Block Sizes

- Networks come in lots of forms
- **Don’t take any block sizes literally.**
- Often (not always) keep some spatial resolution

Encode to spatially smaller tensor, then decode.

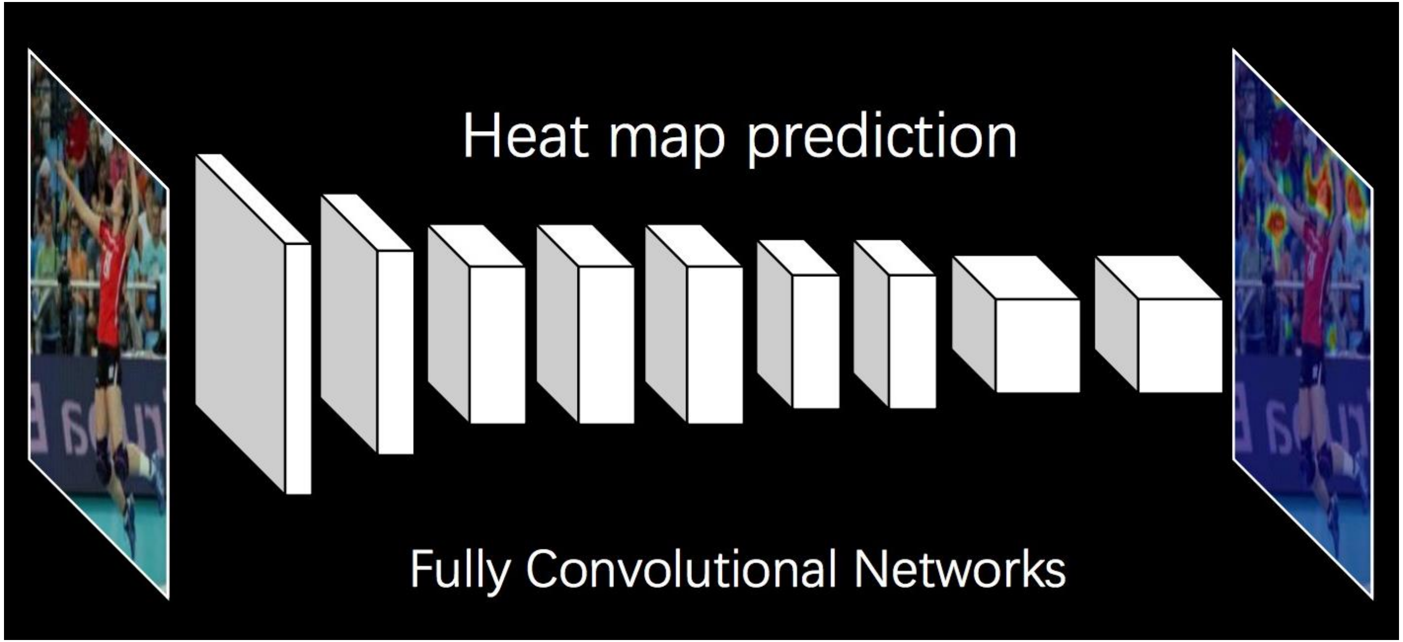


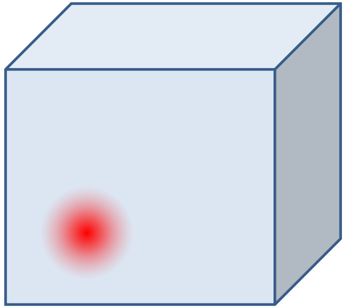
Encode to 1D vector then decode



Slide by David Fouhey

Application to pose detection: Predict heat maps





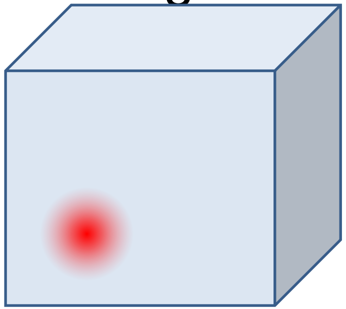
Target: K+1 x H x W Gaussian around (x,y) for k-th keypoint in the k-th channel

K+1 for K parts + background

You will implement this in project 5!

L2 Training Loss

- L2 loss on the target heatmap (peaky gaussian around the gt keypoint)



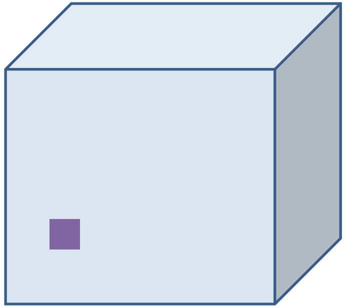
$$L = \sum_{k=1}^{K+1} \sum_{(x,y)} ||b^k(x,y) - b_*^k(x,y)||$$

Target "belief map" :
K+1 x H x W
Gaussian around (x,y) for k-th keypoint in the k-th channel

You will implement this in project 5!

Log Loss Training Loss

- Log loss (or cross entropy loss) on the target heatmap probabilities
- The target must also sum to 1
- Mask RCNN just uses 1 at the target, 0 everywhere else.
- Experiment

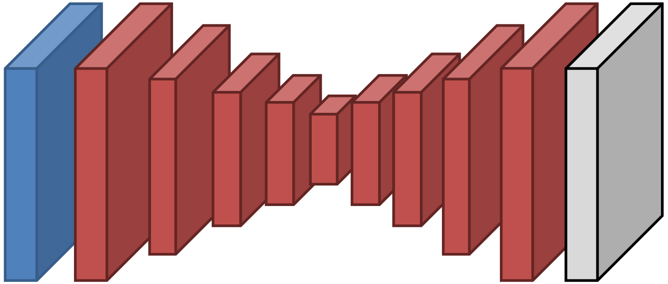


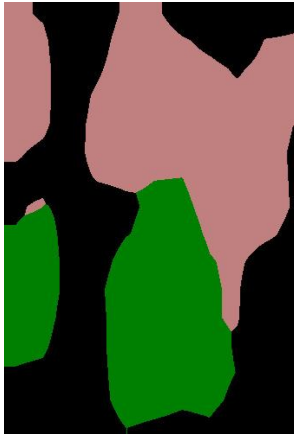
Target “belief map” :
K+1 x H x W
1 at Grount truth location (x,y) for k-th keypoint in the k-th channel

Missing Details

While the output *is* HxW, just upsampling often produces results without details/not aligned with the image.
Why?

Information about details lost when downsampling!



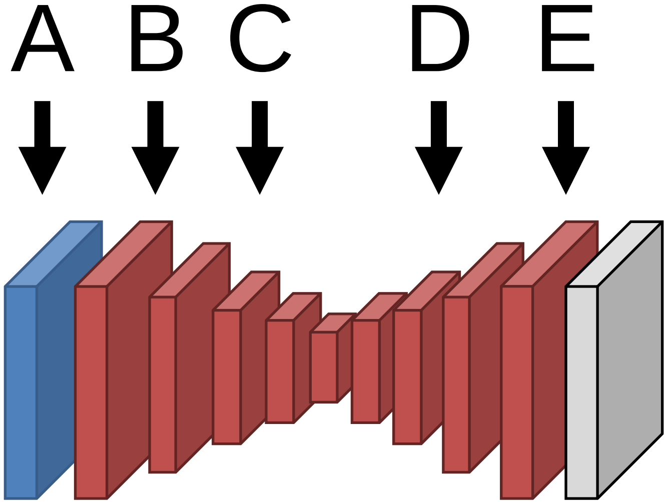


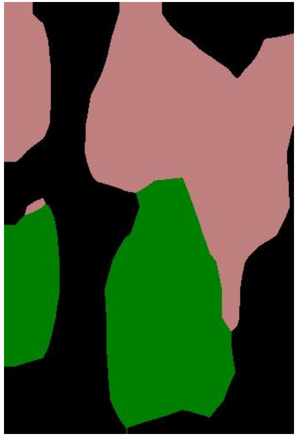
Result from Long et al. *Fully Convolutional Networks For Semantic Segmentation*. CVPR 2014

Slide by David Fouhey

Missing Details

Where is the useful information about the high-frequency details of the image?





Result from Long et al. *Fully Convolutional Networks For Semantic Segmentation*. CVPR 2014

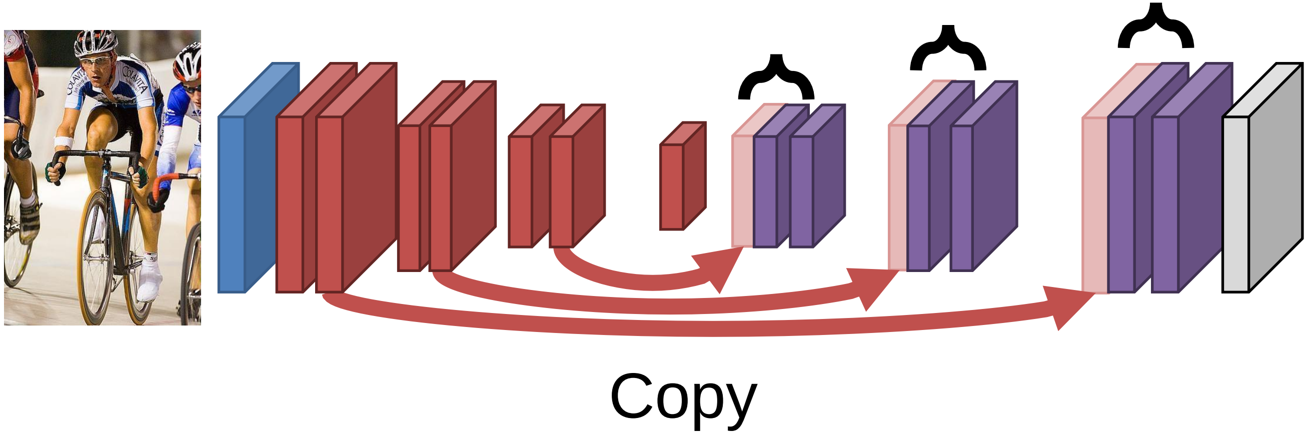
Slide by David Fouhey

Missing Details

How do you send details forward in the network?

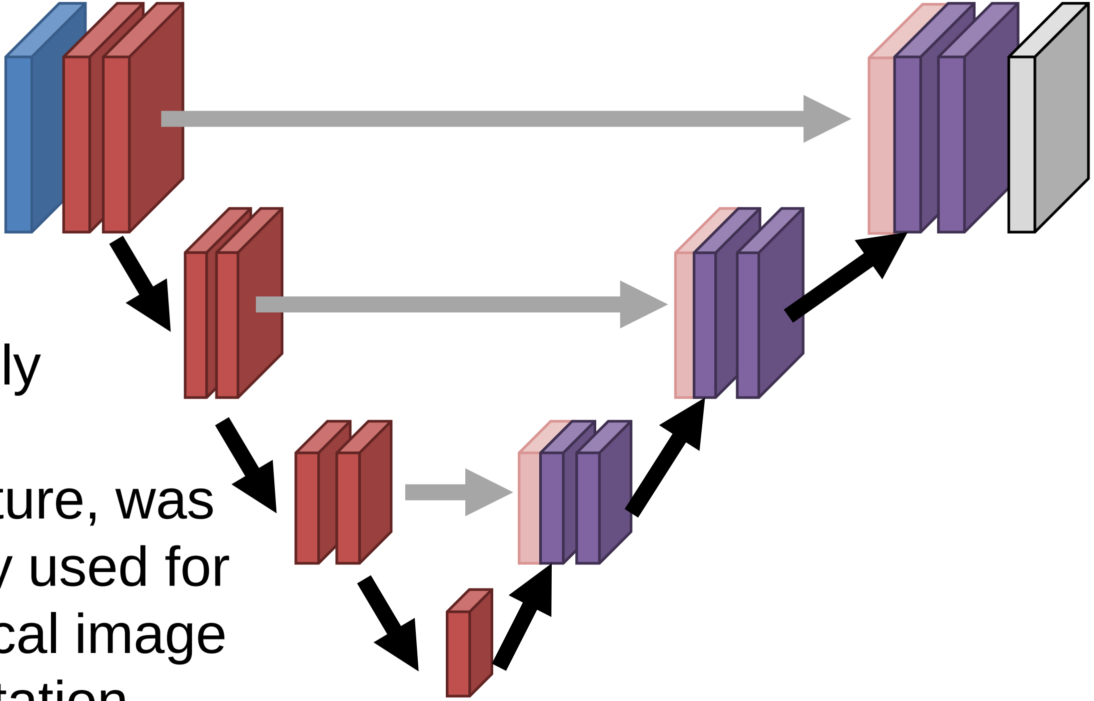
You copy the activations forward.

Subsequent layers at the same resolution figure out how to fuse things.



Result from Long et al. *Fully Convolutional Networks For Semantic Segmentation*. CVPR 2014

U-Net



Extremely popular architecture, was originally used for biomedical image segmentation.

Ronneberger et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015

U-Net improves performance

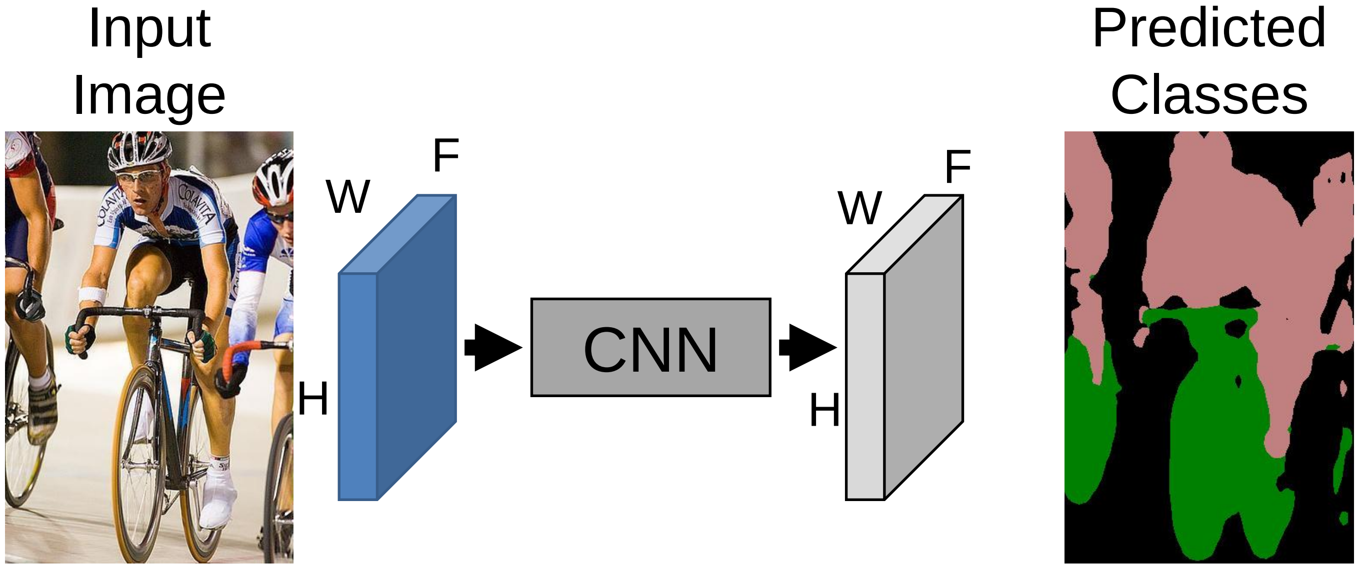
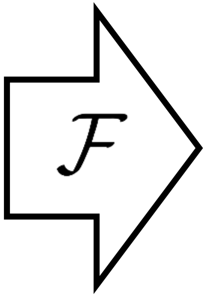


Image Colorization

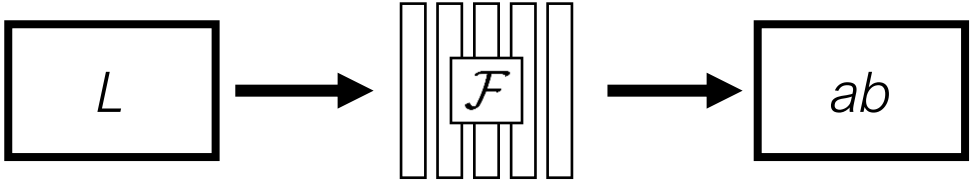


Grayscale image: *L* channel

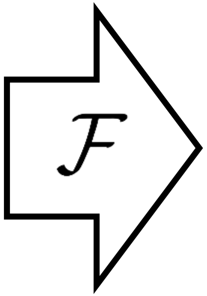
$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

Color information: *ab* channels

$$\widehat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$$



Zhang, Isola, Efros. ***Colorful Image Colorization.*** In *ECCV*, 2016.

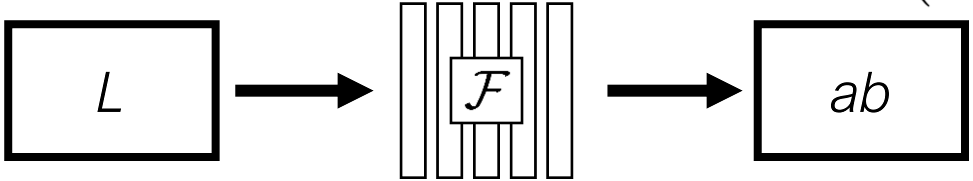


Grayscale image: *L* channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

Concatenate (*L*,*ab*) channels

$$(\mathbf{X}, \widehat{\mathbf{Y}})$$



Zhang, Isola, Efros. ***Colorful Image Colorization.*** In *ECCV*, 2016.

Regressing to pixel values doesn't work ☹

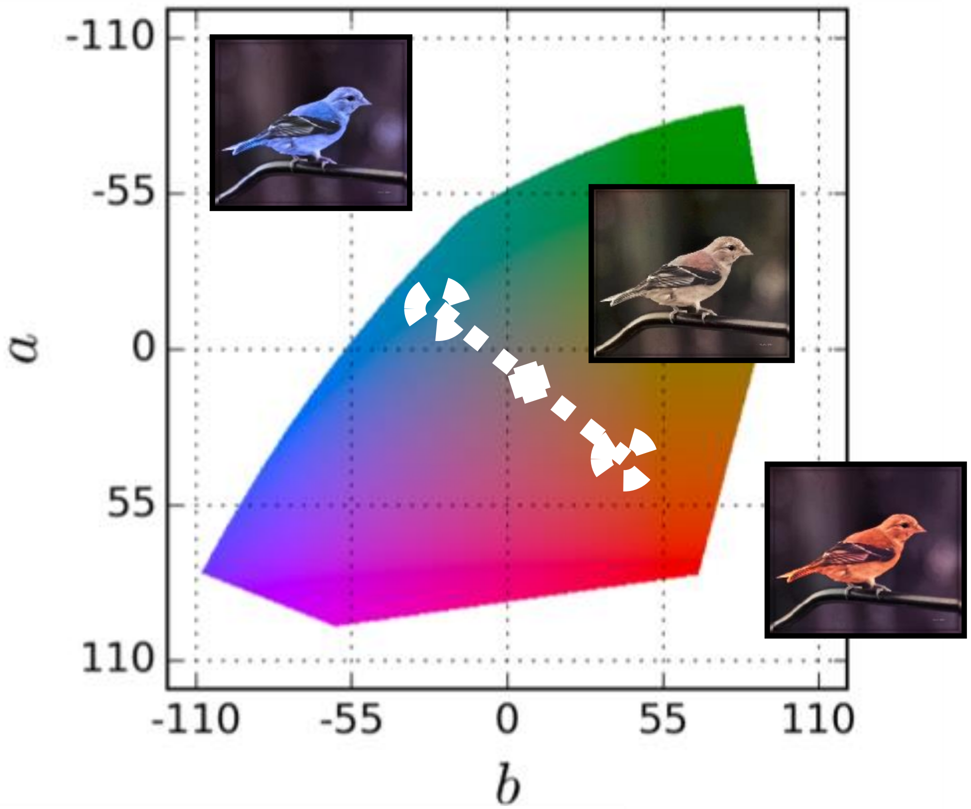
Input

Output

Ground truth

$$L_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

Slide by Richard Zhang



$$L_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

Slide by Richard Zhang

Better Loss Function

$$\theta^* = \arg\min_{\theta} \ell(\mathcal{F}_\theta(\mathbf{X}), \mathbf{Y})$$

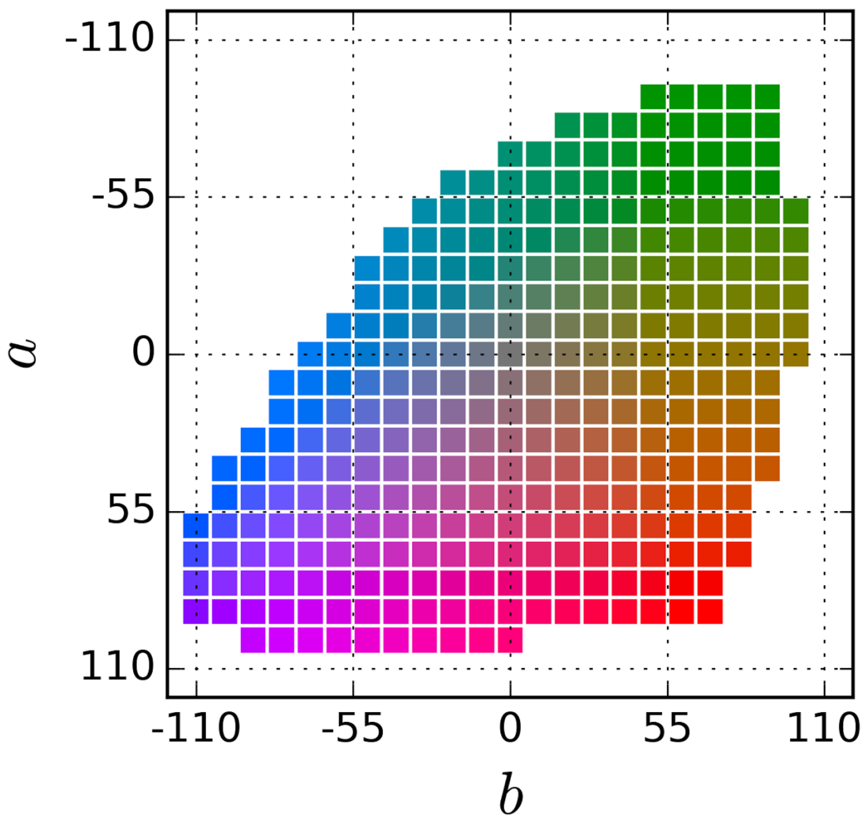
- Regression with L2 loss inadequate

$$L_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

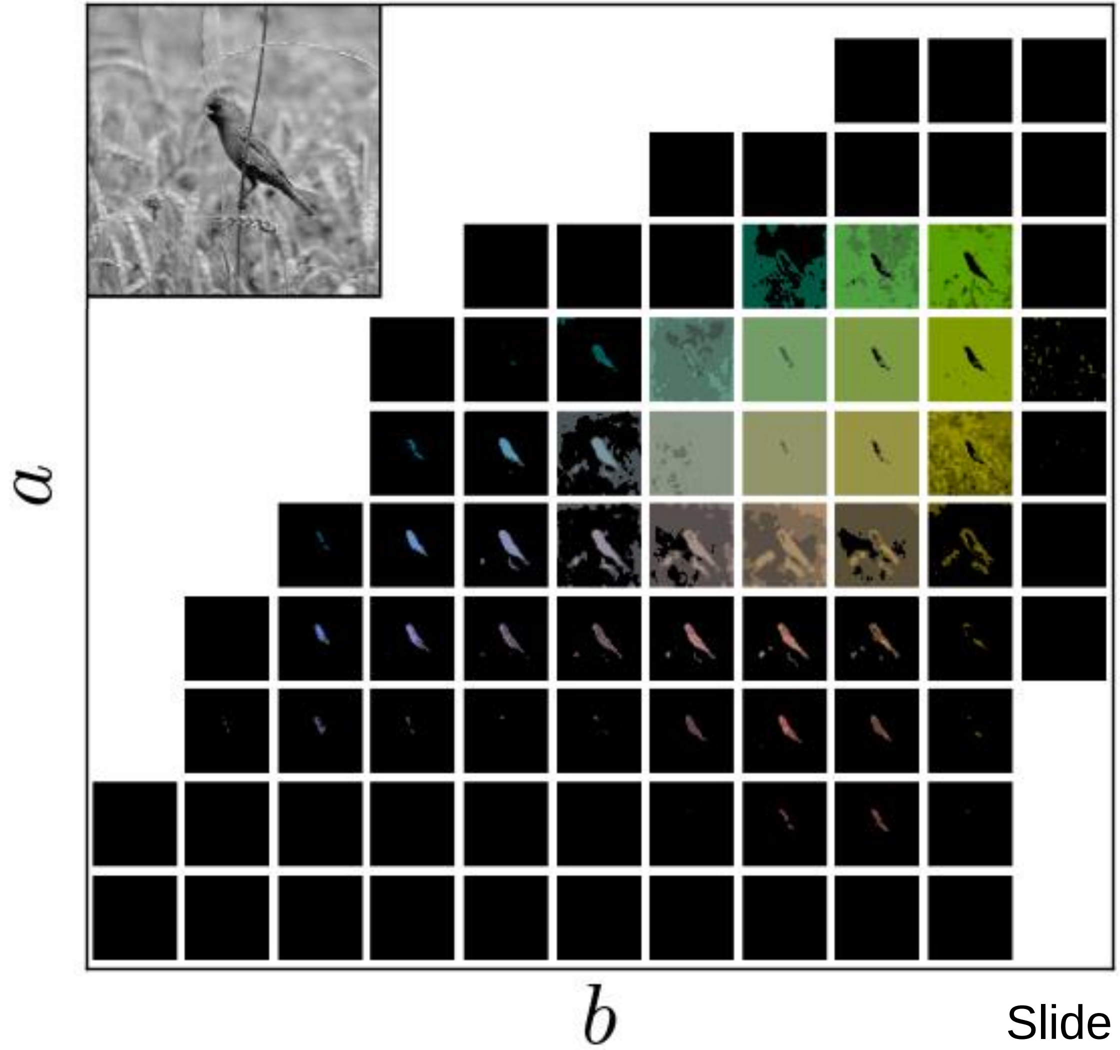
- Use per-pixel multinomial classification

$$L(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW}\sum_{h,w}\sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$

Colors in *ab* space
(discrete)



Slide by Richard Zhang



Slide by Richard Zhang

Designing pixel loss functions

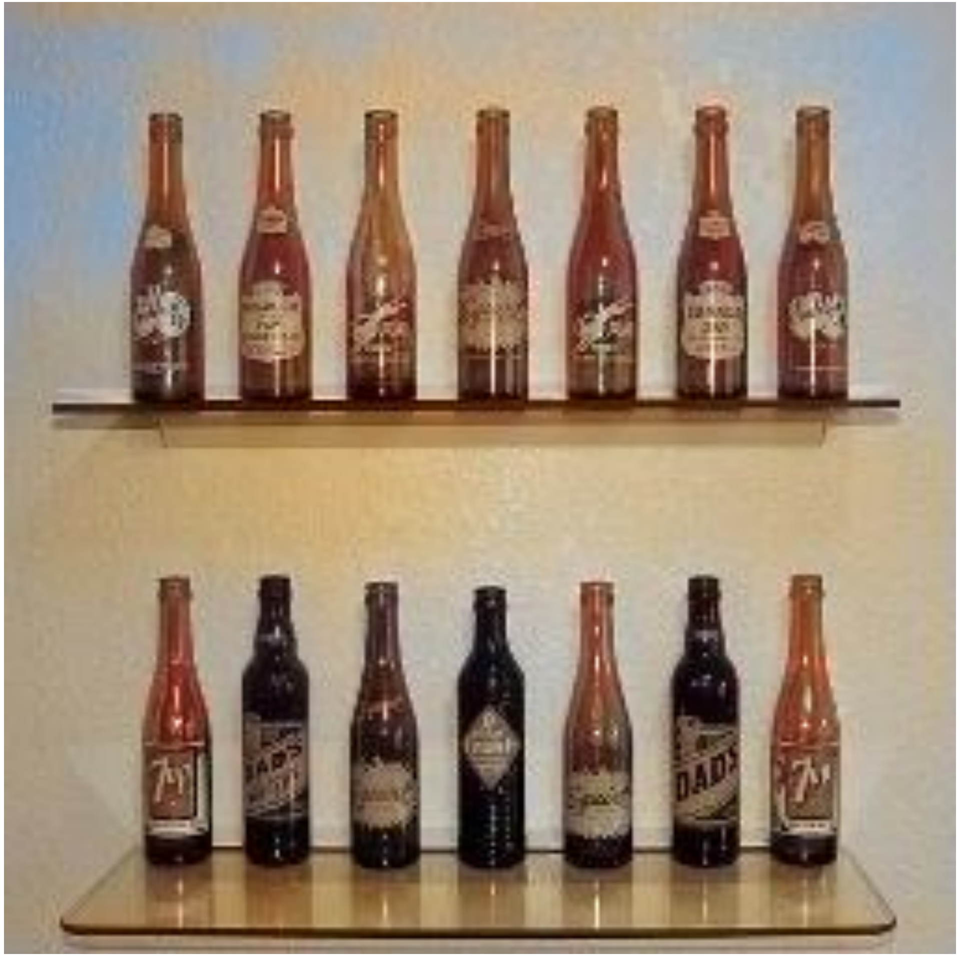
Input

Zhang et al. 2016

Ground truth

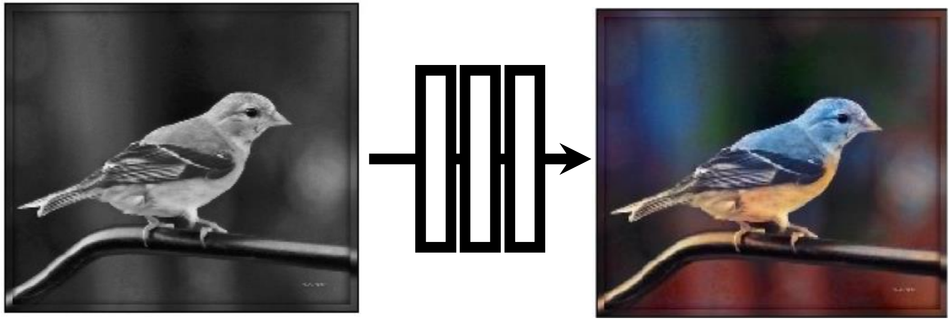
Color distribution cross-entropy loss with colorfulness enhancing term.

[Zhang, Isola, Efros, ECCV 2016]


DADS

Designing pixel loss functions

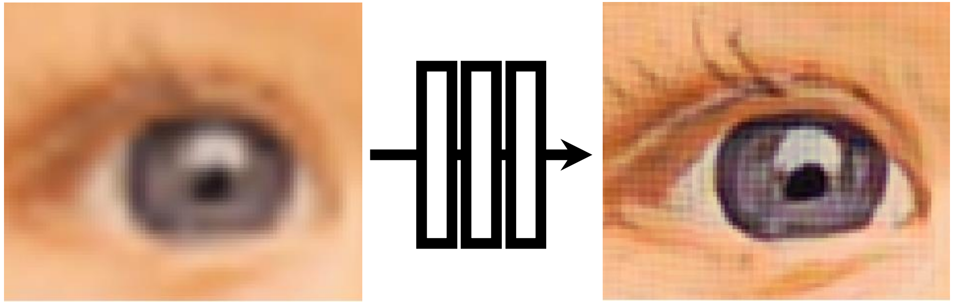
Image colorization



[Zhang et al. 2016]

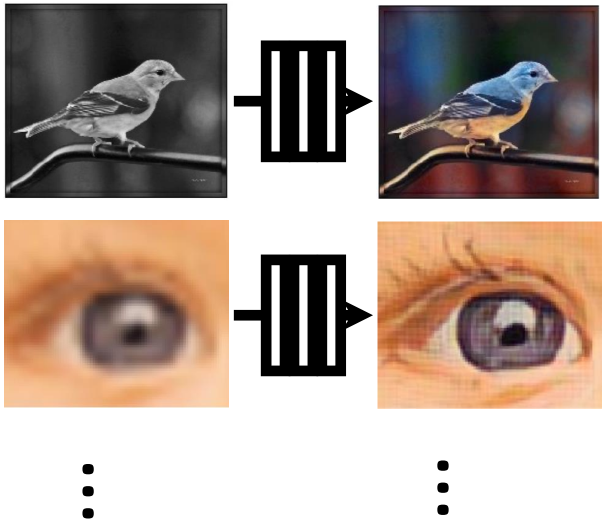
Cross entropy loss, with colorfulness term

Super-resolution



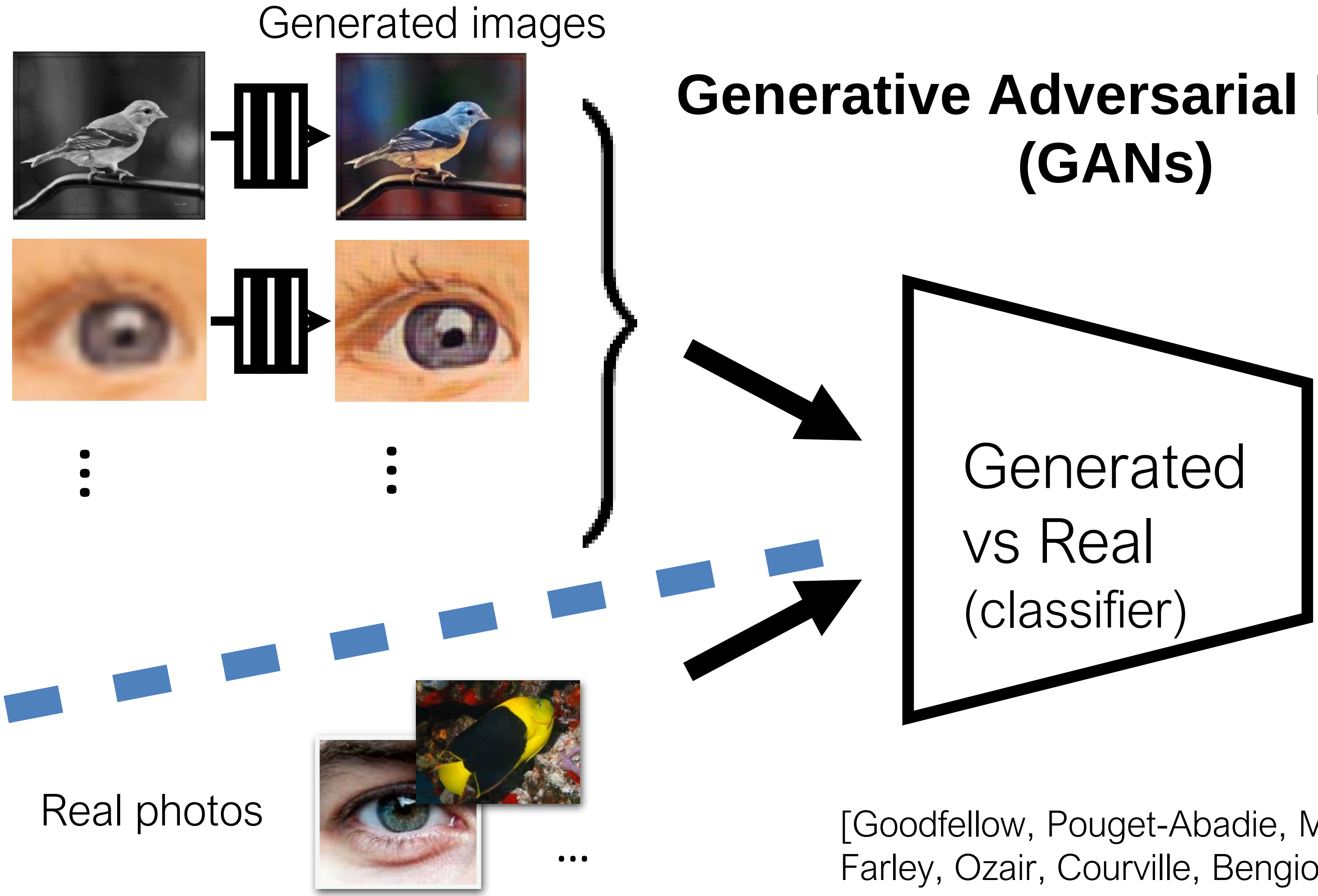
[Johnson et al. 2016]

“semantic feature loss” (VGG feature covariance matching objective)

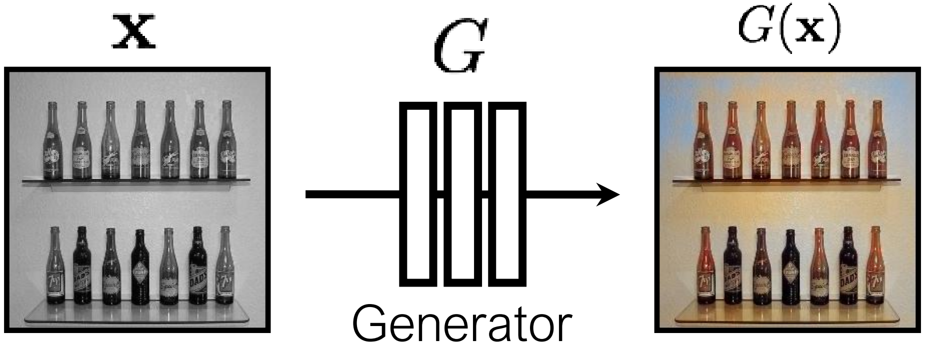


Universal loss?

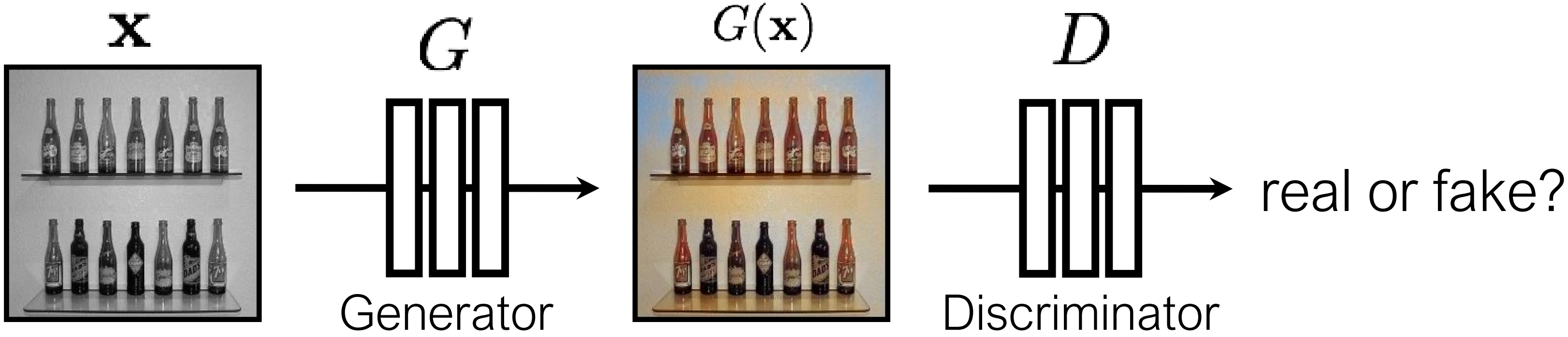
Generative Adversarial Network (GANs)



[Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, Bengio 2014]



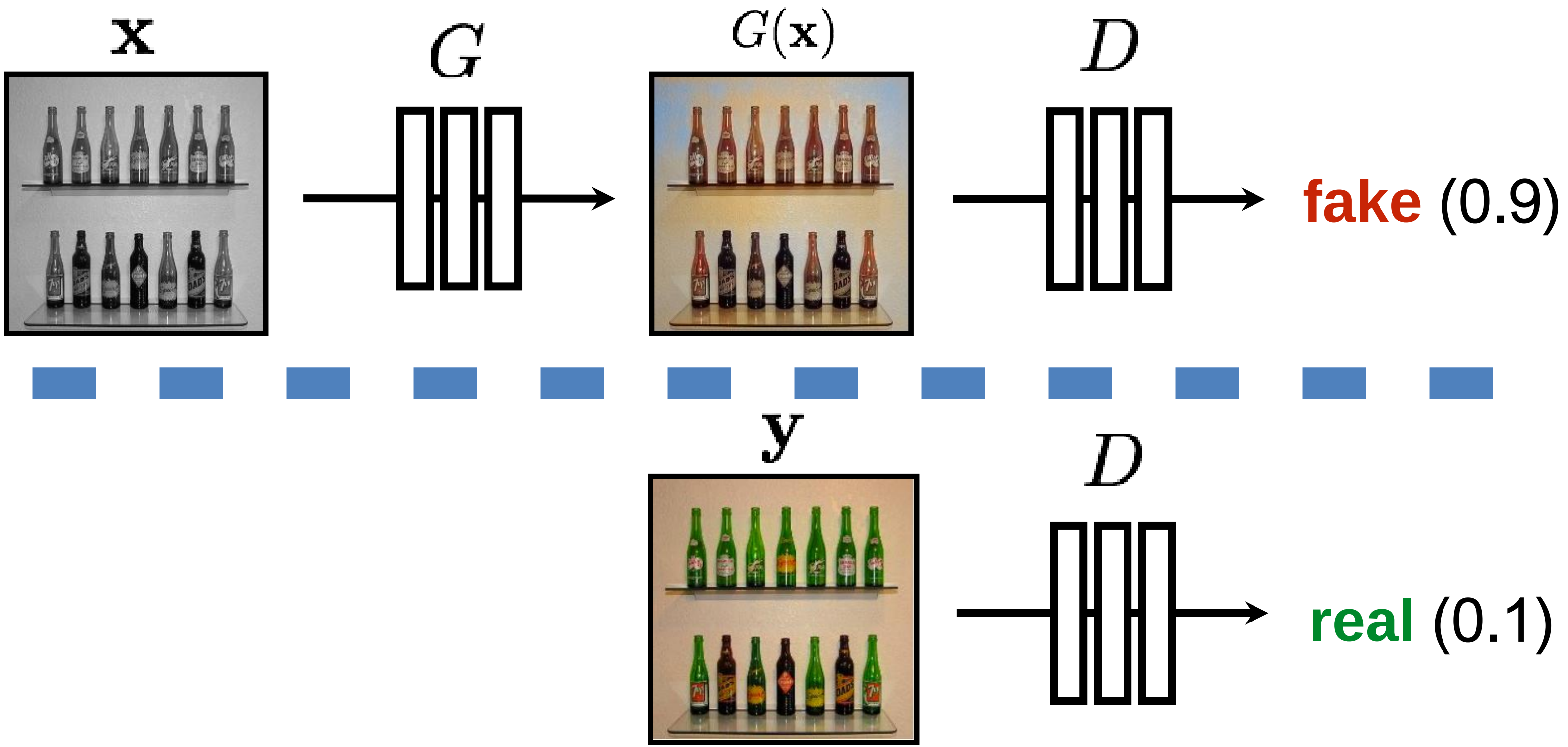
[Goodfellow et al., 2014]



G tries to synthesize fake images that fool **D**

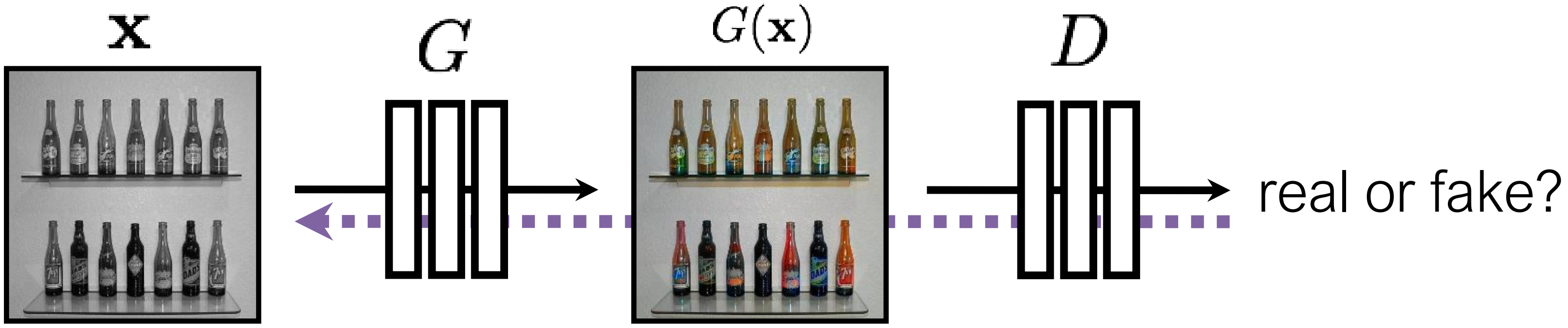
D tries to identify the fakes

[Goodfellow et al., 2014]



$$\arg\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \boxed{\log D(G(\mathbf{x}))} + \boxed{\log(1 - D(\mathbf{y}))}\ ]$$

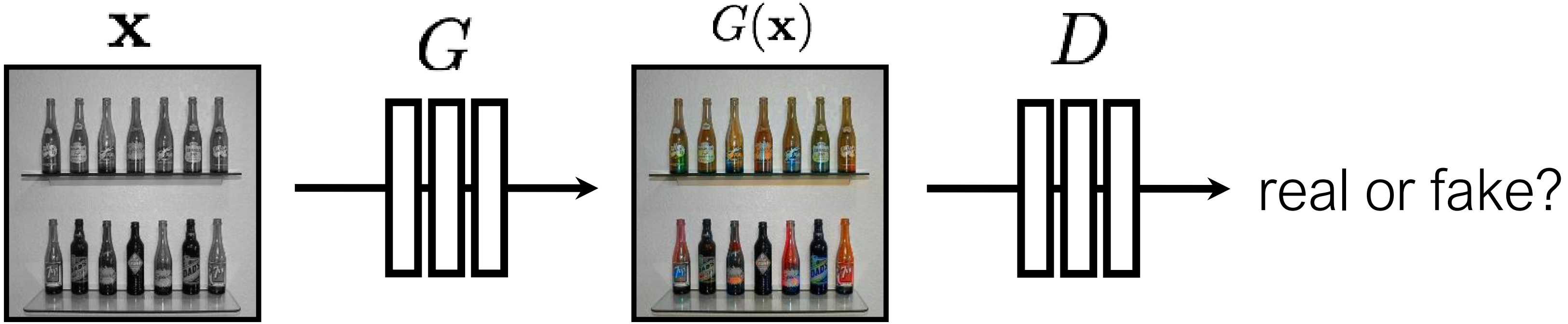
[Goodfellow et al., 2014]



G tries to synthesize fake images that ***fool*** **D**:

$$\arg \boxed{\min_{G}} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y})) \;]$$

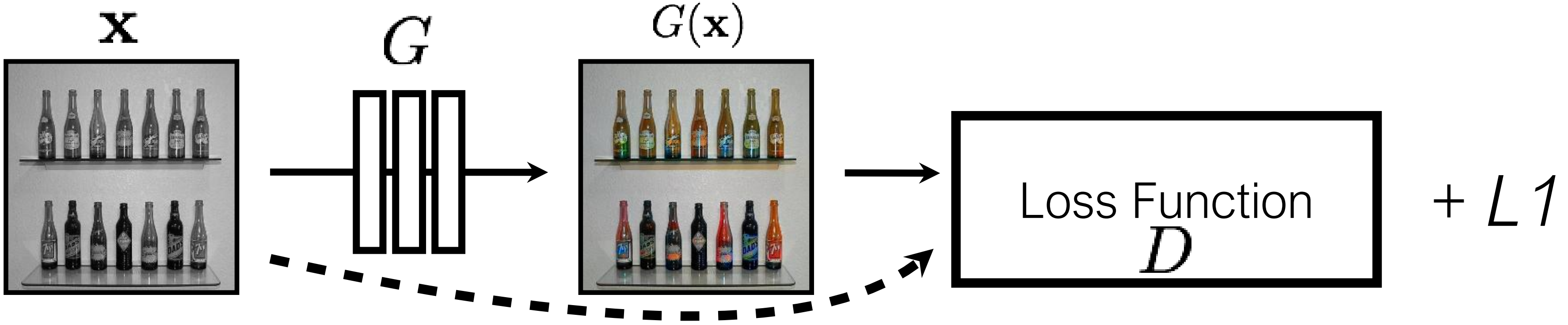
[Goodfellow et al., 2014]



G tries to synthesize fake images that ***fool*** the ***best*** **D**:

$$\arg \boxed{\min_{G}} \boxed{\max_{D}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ ]$$

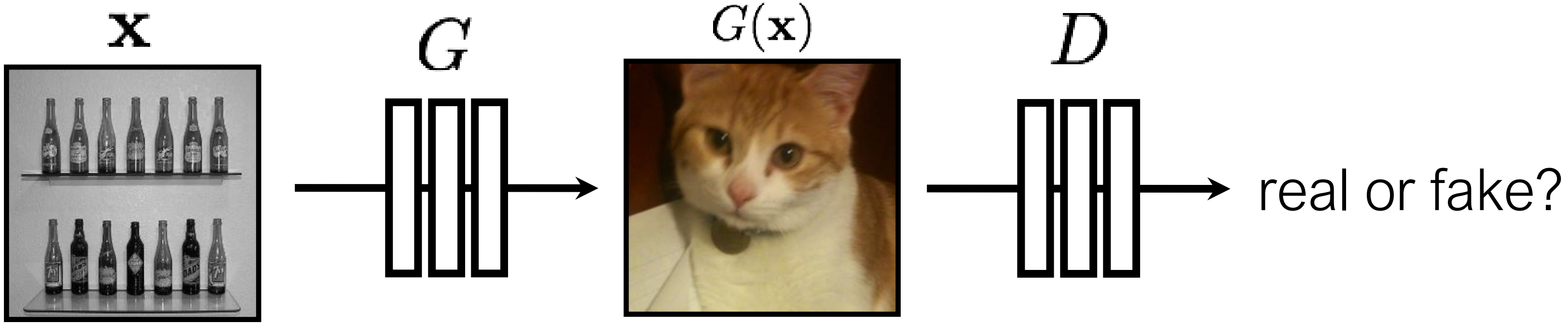
[Goodfellow et al., 2014]



G’s perspective: **D** is a loss function.

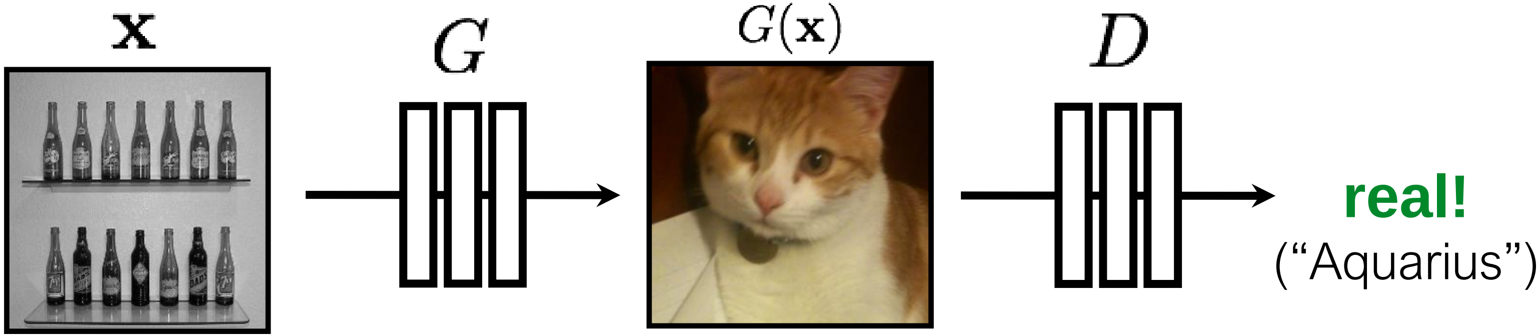
Rather than being hand-designed, it is *learned*.

[Goodfellow et al., 2014]
[Isola et al., 2017]



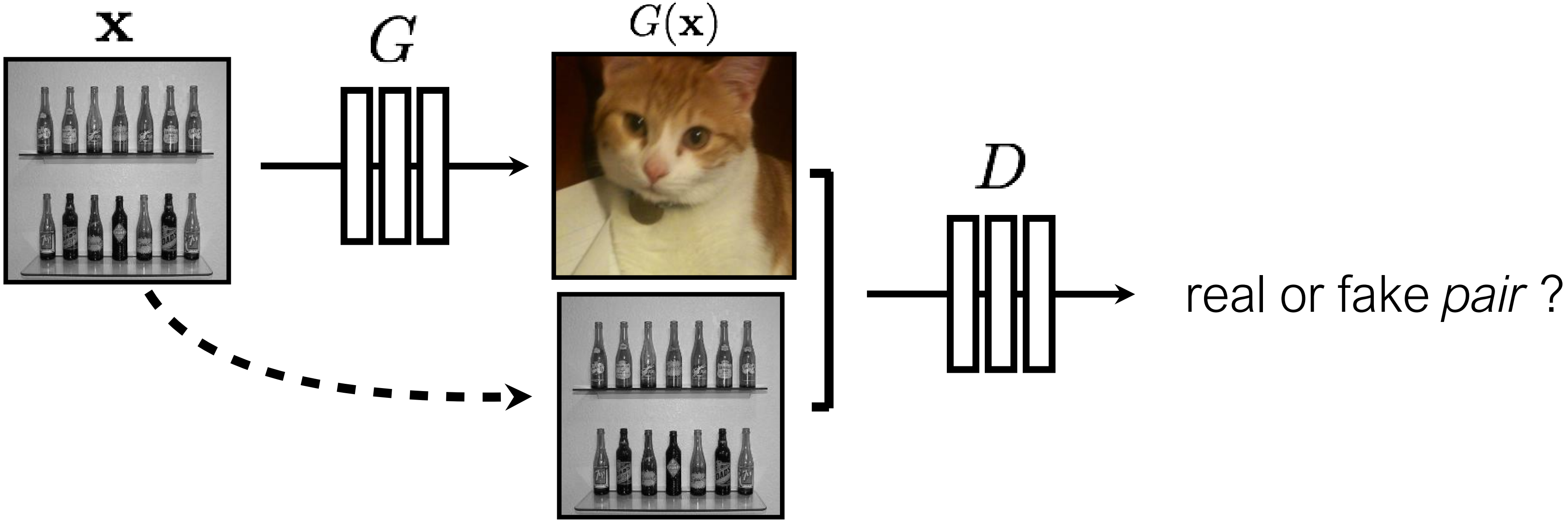
$$\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \ + \ \log(1 - D(\mathbf{y}))\ ]$$

[Goodfellow et al., 2014]



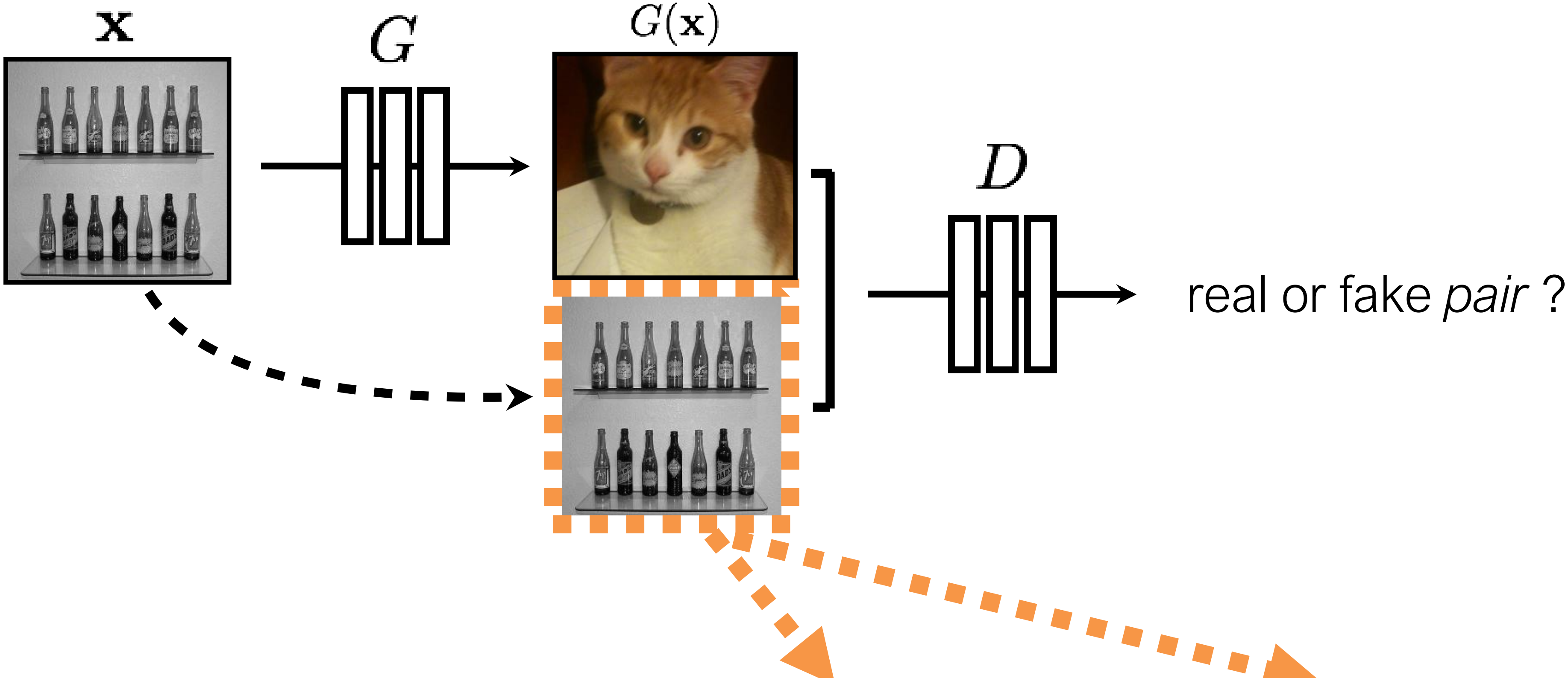
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ \ ]$$

[Goodfellow et al., 2014]



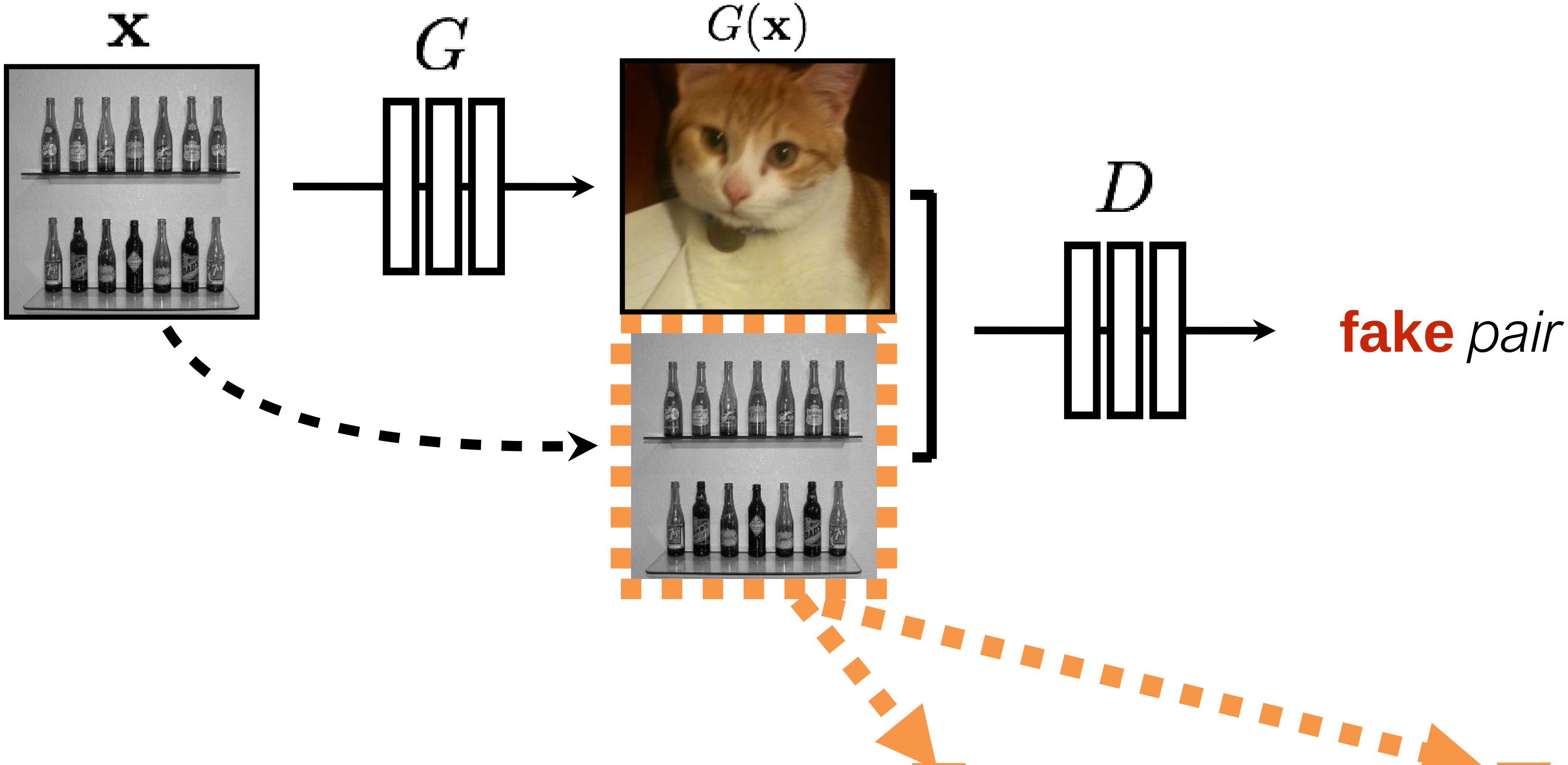
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ ]$$

[Goodfellow et al., 2014]
[Isola et al., 2017]



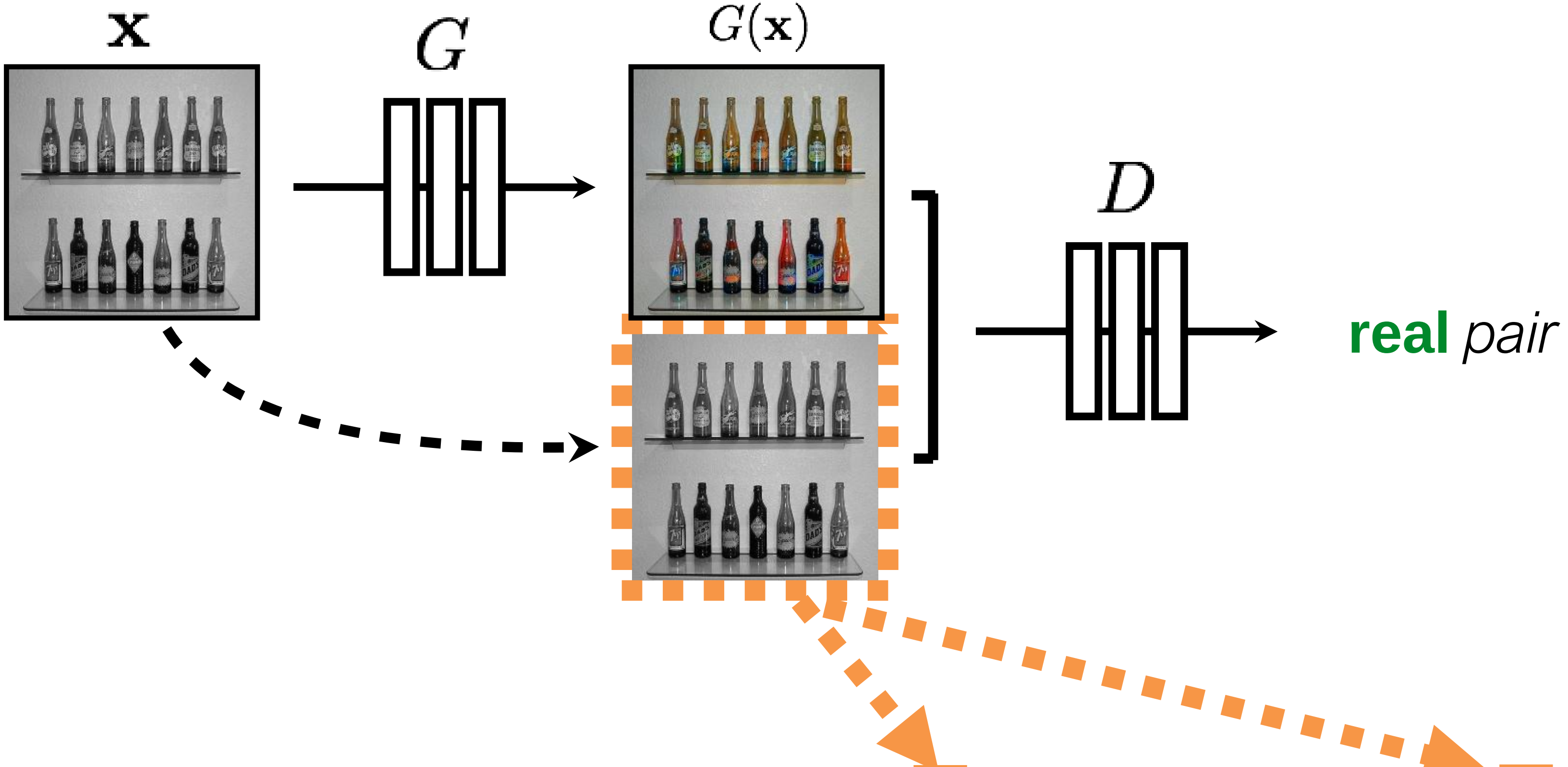
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ ]$$

[Goodfellow et al., 2014]
[Isola et al., 2017]



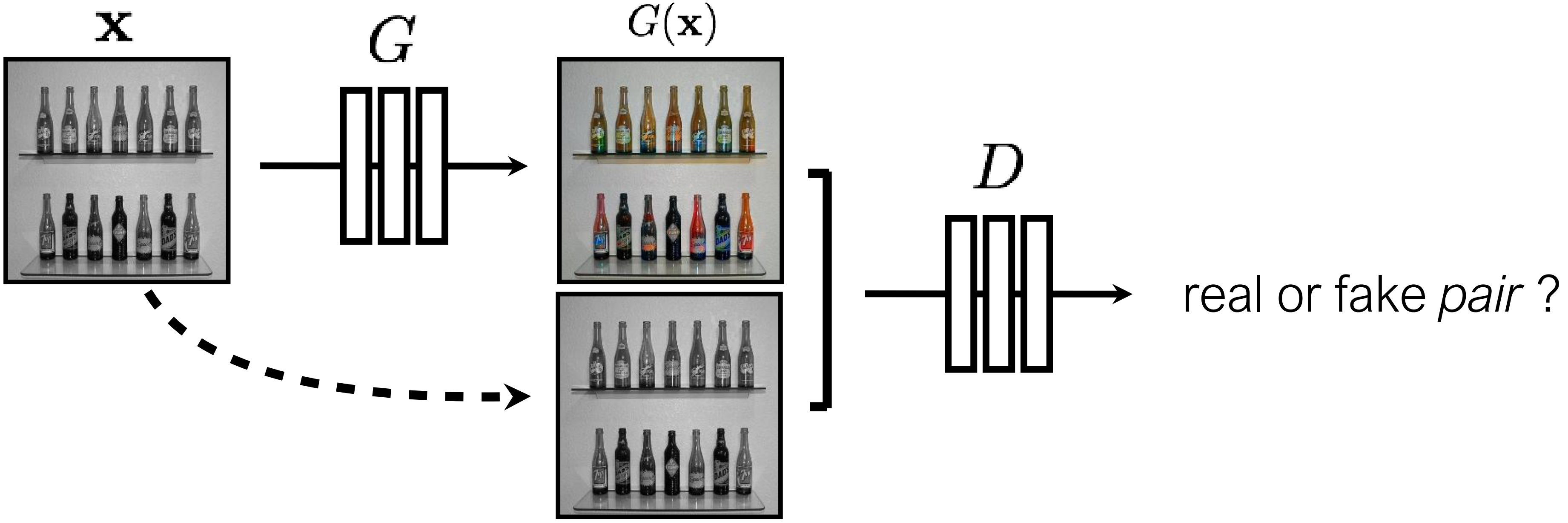
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y})) \;]$$

[Goodfellow et al., 2014]
[Isola et al., 2017]



$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ ]$$

[Goodfellow et al., 2014]
[Isola et al., 2017]



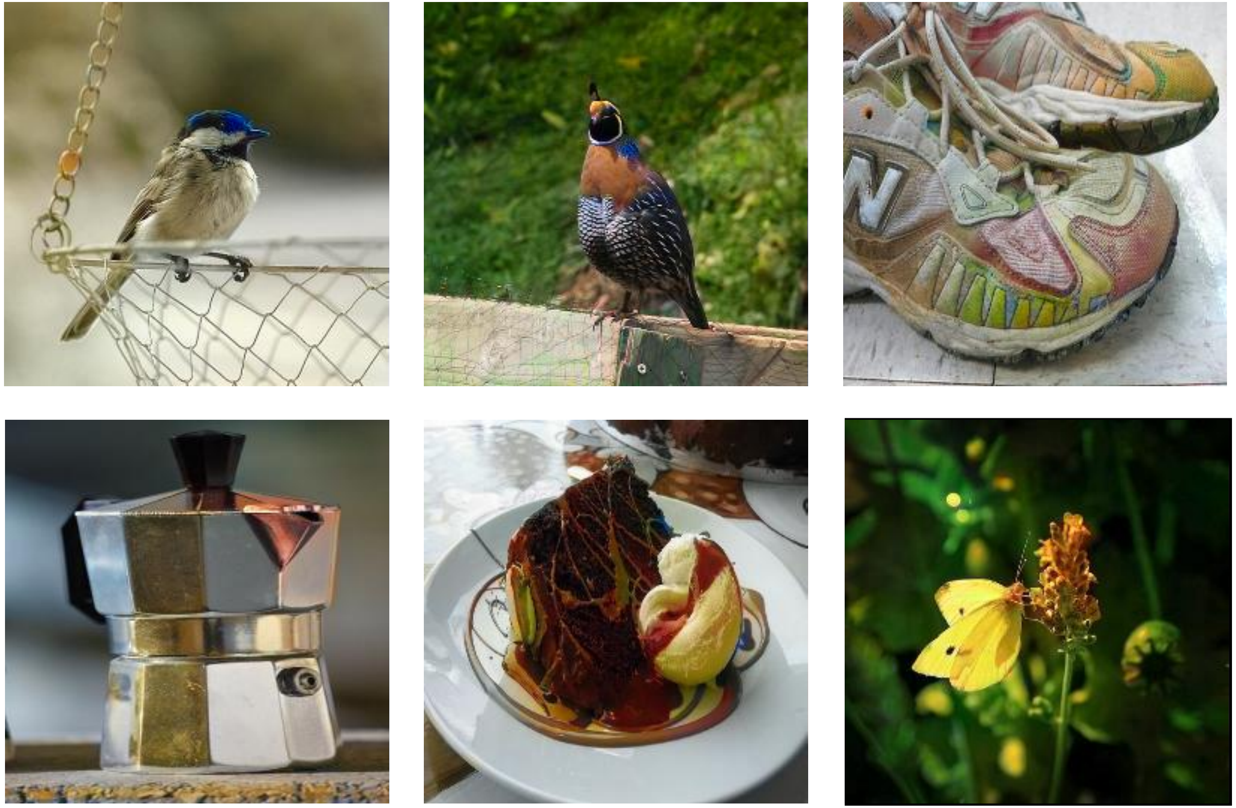
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\;\log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$

[Goodfellow et al., 2014]
[Isola et al., 2017]

BW → Color

Data from [Russakovsky et al. 2015]

BW → Color



Data from [Russakovsky et al. 2015]

Input

Output

Groundtruth

Data from
[maps.google.com]

Input

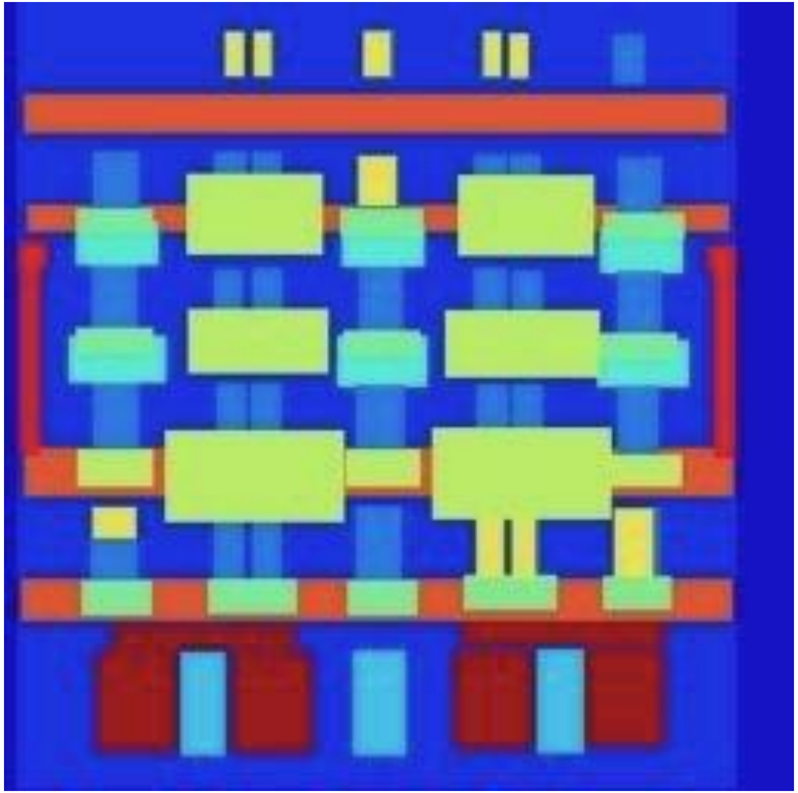
Output

Groundtruth

Data from [maps.google.

Labels → Facades

Input

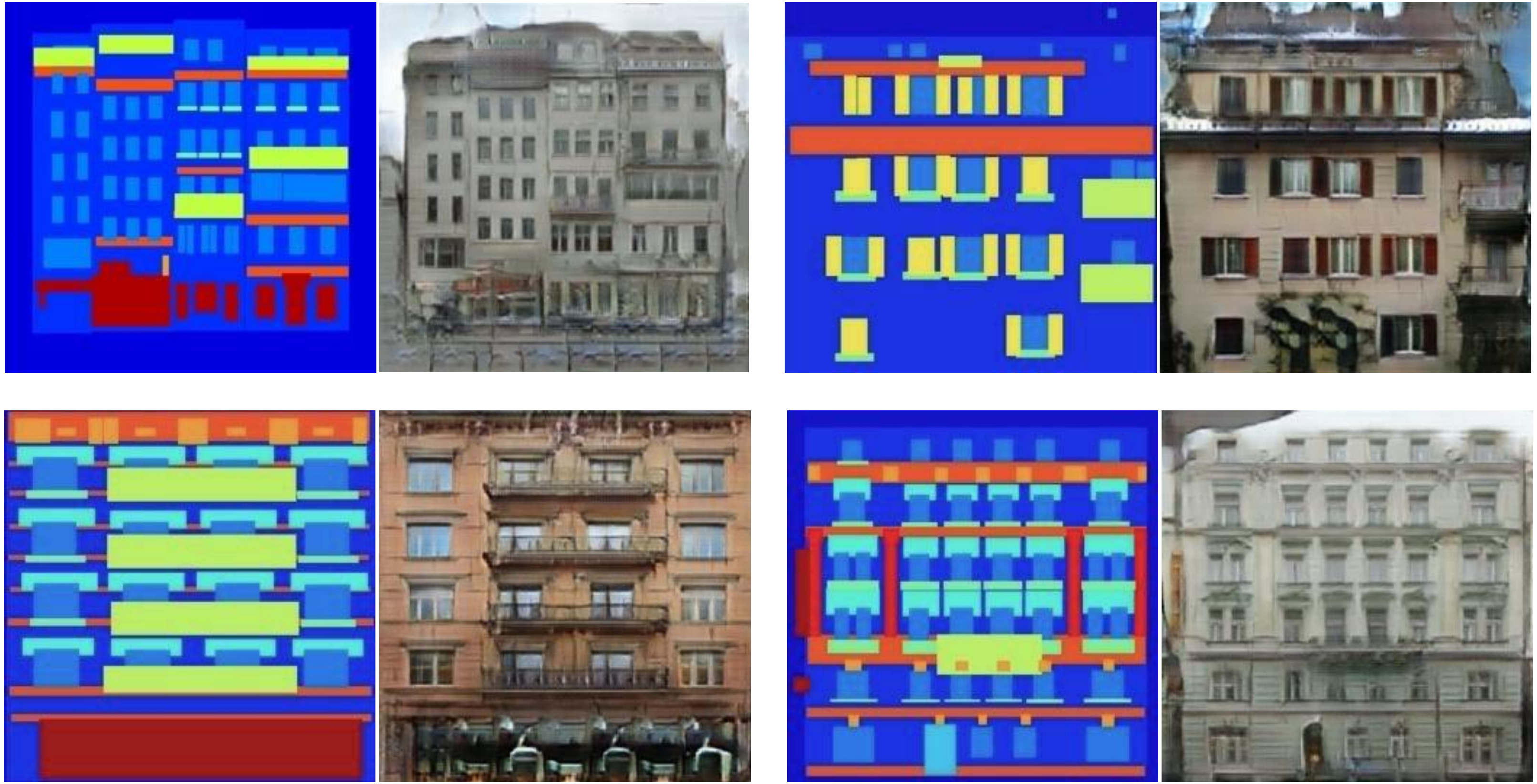


Output

Data from [Tylecek, 2013]

Labels → Facades

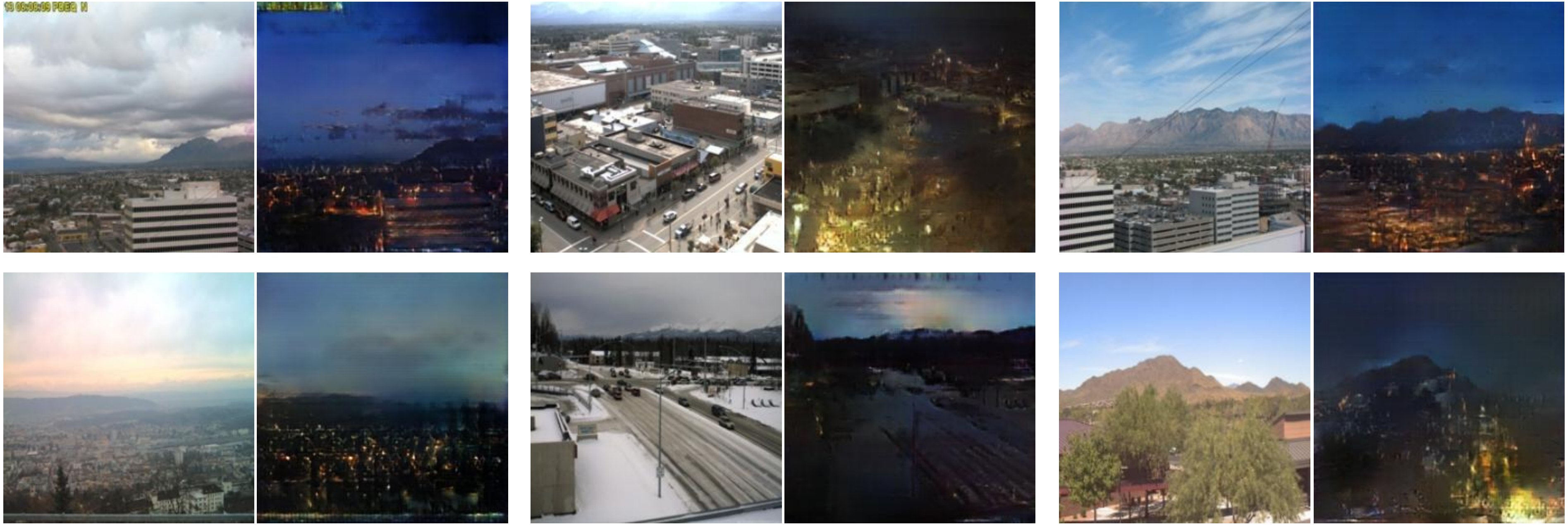
Input | Output | Input | Output



Data from [Tylecek, 2013]

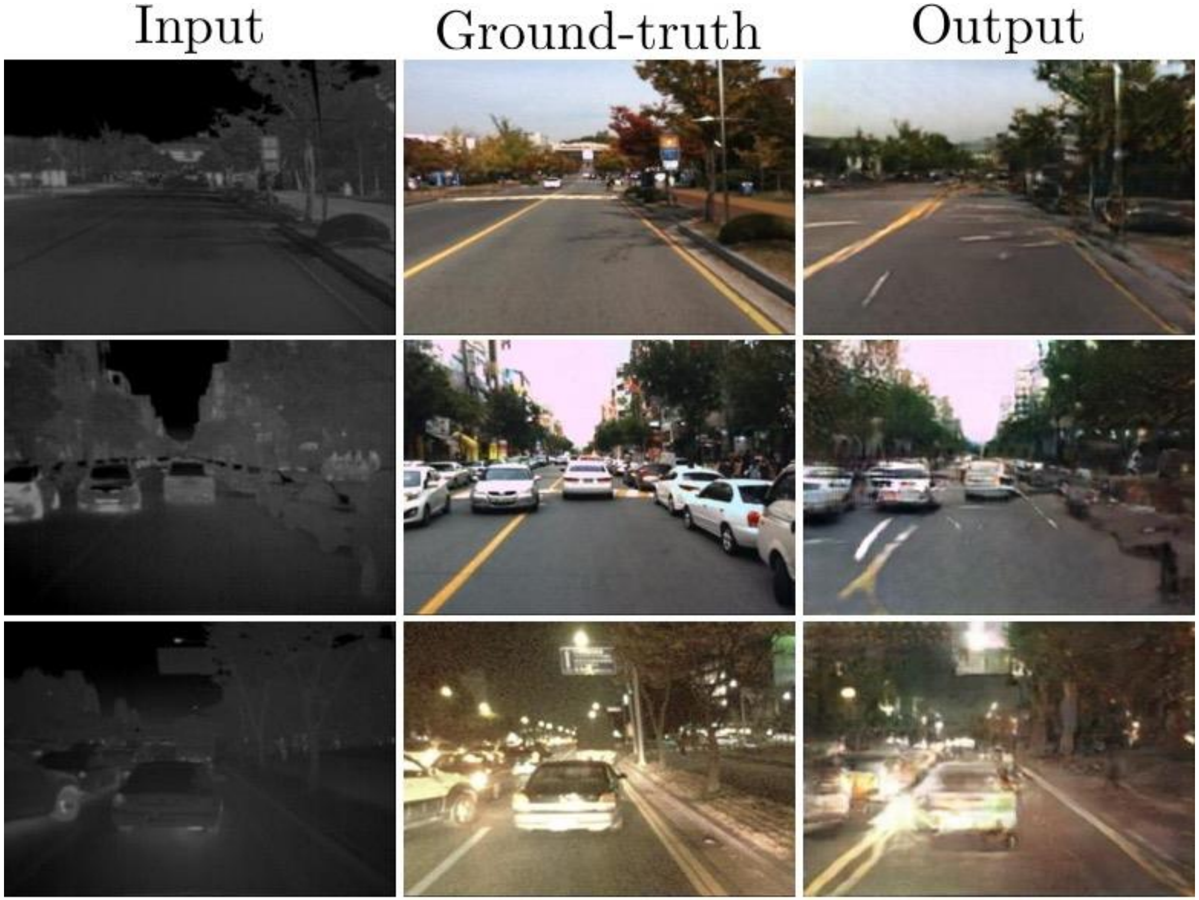
Day → Night

Input Output Input Output Input Output



Data from [Laffont et al., 2014]

Thermal → RGB

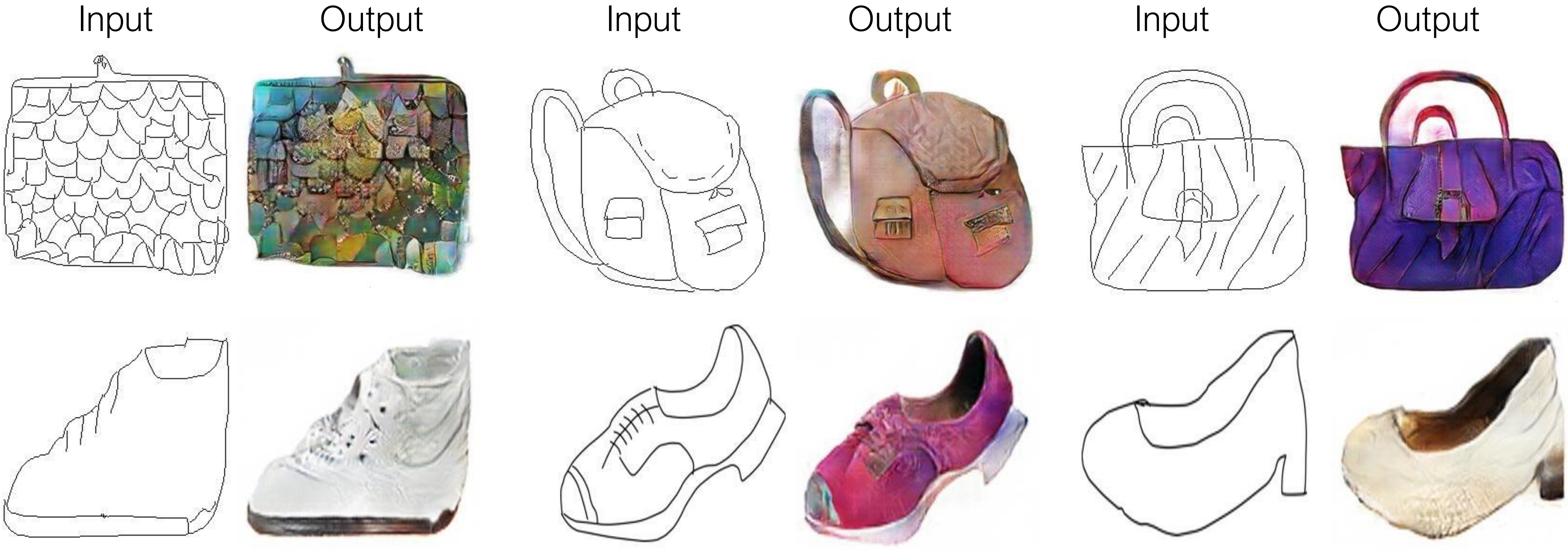


Edges → Images

Input Output Input Output Input Output

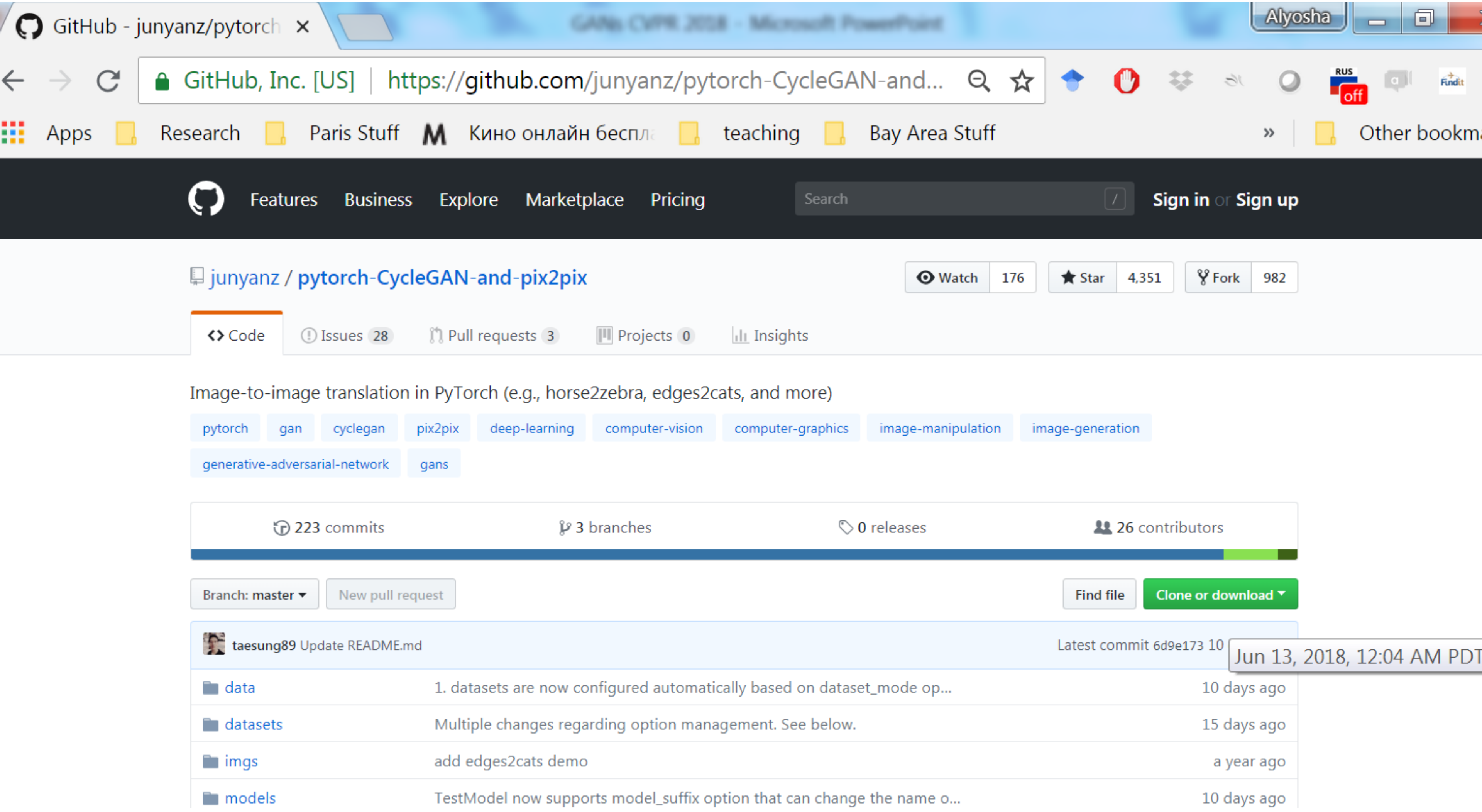
Edges from [Xie & Tu, 2015]

Sketches → Images

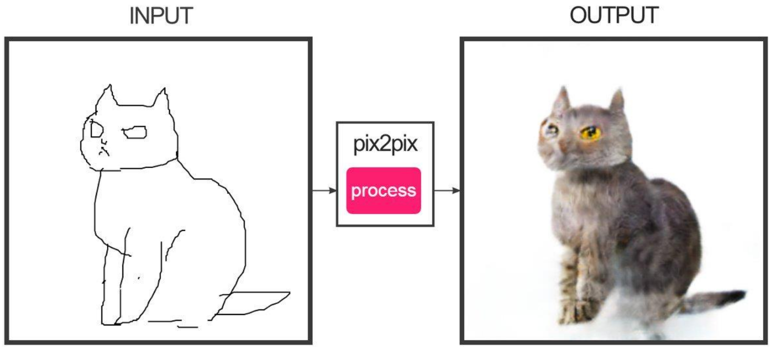


Trained on Edges → Images

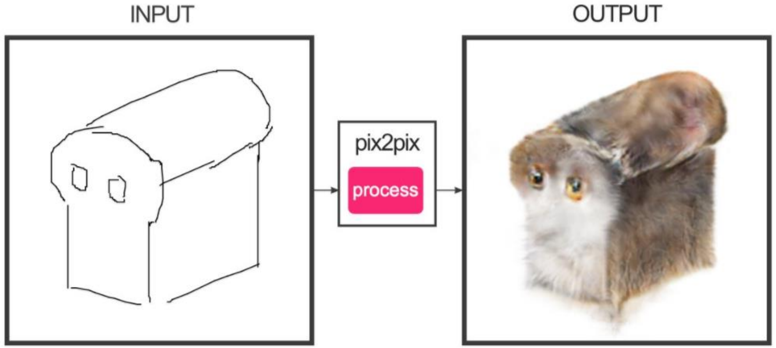
Data from [Eitz, Hays, Alexa, 2012]


GitHub - junyanz/pytorch
Alyosha
GitHub, Inc. [US] https://github.com/junyanz/pytorch-CycleGAN-and...
Apps
Research
Paris Stuff
Кино онлайн бесп
teaching
Bay Area Stuff
Other bookm
Features
Business
Explore
Marketplace
Pricing
Search
Sign in or Sign up
junyanz / pytorch-CycleGAN-and-pix2pix
Watch 176
Star 4,351
Fork 982
Code
Issues 28
Pull requests 3
Projects 0
Insights
Image-to-image translation in PyTorch (e.g., horse2zebra, edges2cats, and more)
pytorch
gan
cyclegan
pix2pix
deep-learning
computer-vision
computer-graphics
image-manipulation
image-generation
generative-adversarial-network
gans
223 commits
3 branches
0 releases
26 contributors
Branch: master
New pull request
Find file
Clone or download
taesung89 Update README.md
Latest commit 6d9e173 10
Jun 13, 2018, 12:04 AM PDT
data
1. datasets are now configured automatically based on dataset_mode op...
10 days ago
datasets
Multiple changes regarding option management. See below.
15 days ago
imgs
add edges2cats demo
a year ago
models
TestModel now supports model_suffix option that can change the name o...
10 days ago

#edges2cats [Christopher Hesse]



@gods_tail

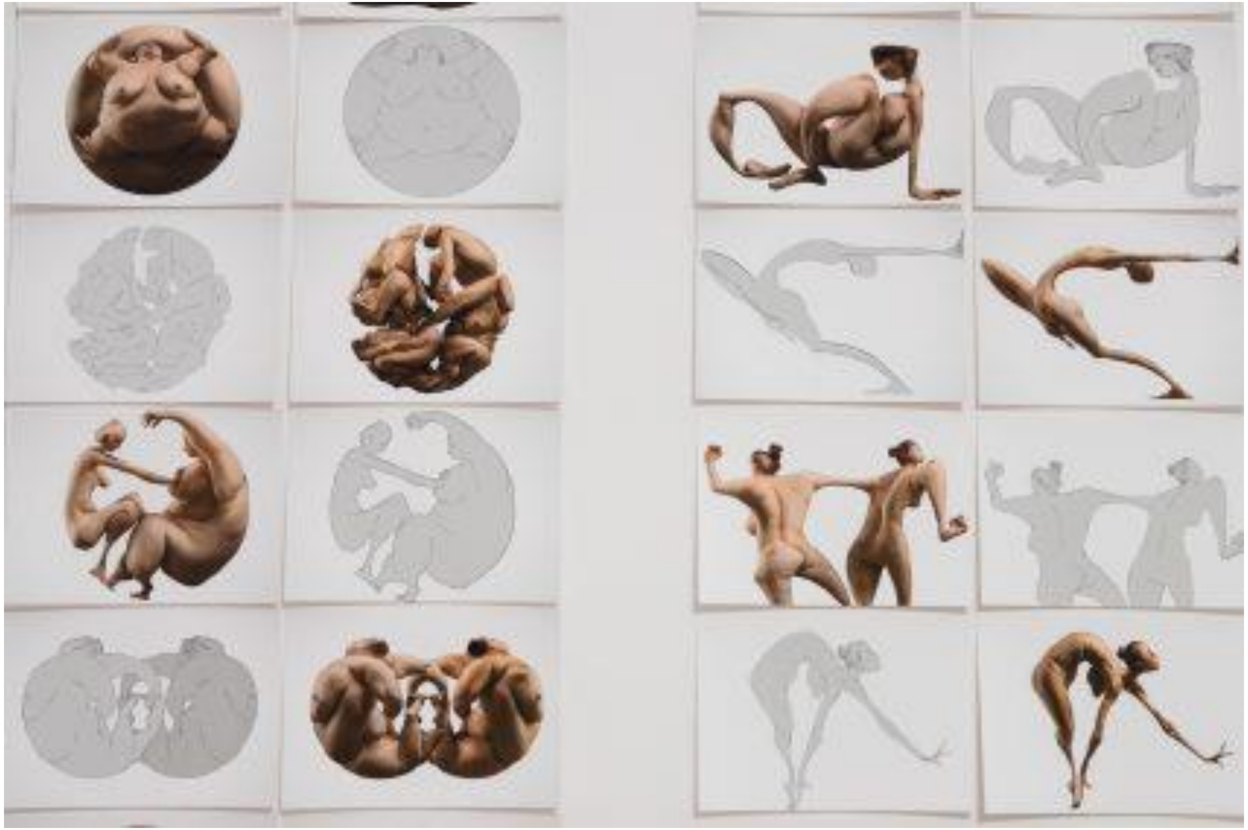


Ivy Tasi @ivymyt

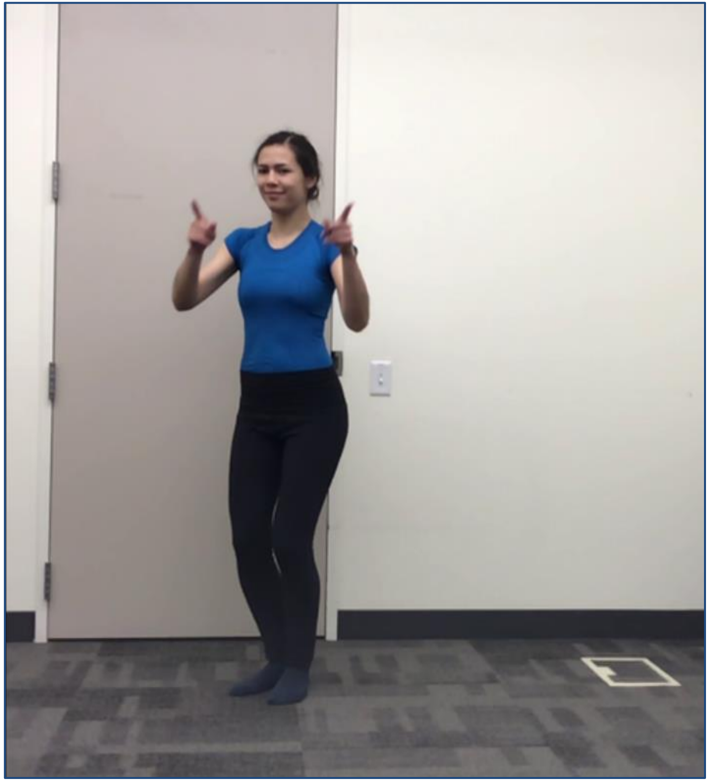
Vitaly Vidmirov @vvid

@ka92

Scott Eaton (http://www.scott-eaton.com/)

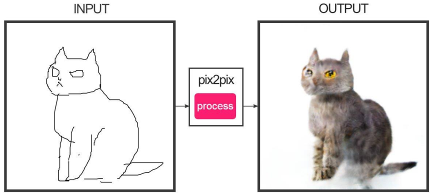


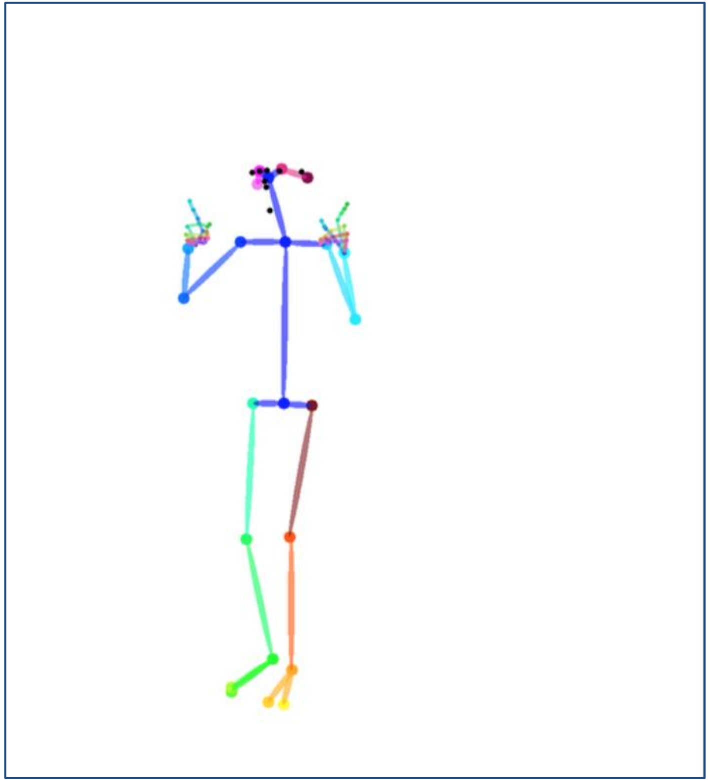
“Do as I Do”



OpenPose

pix2pix





Everybody Dance Now

Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros
UC Berkeley

Source Subject

Target Subject

Results

https://www.youtube.com/watch?v=PCBTZh41Ris&feature=youtu.be

"You need a lot of a data if you want to train/use CNNs"

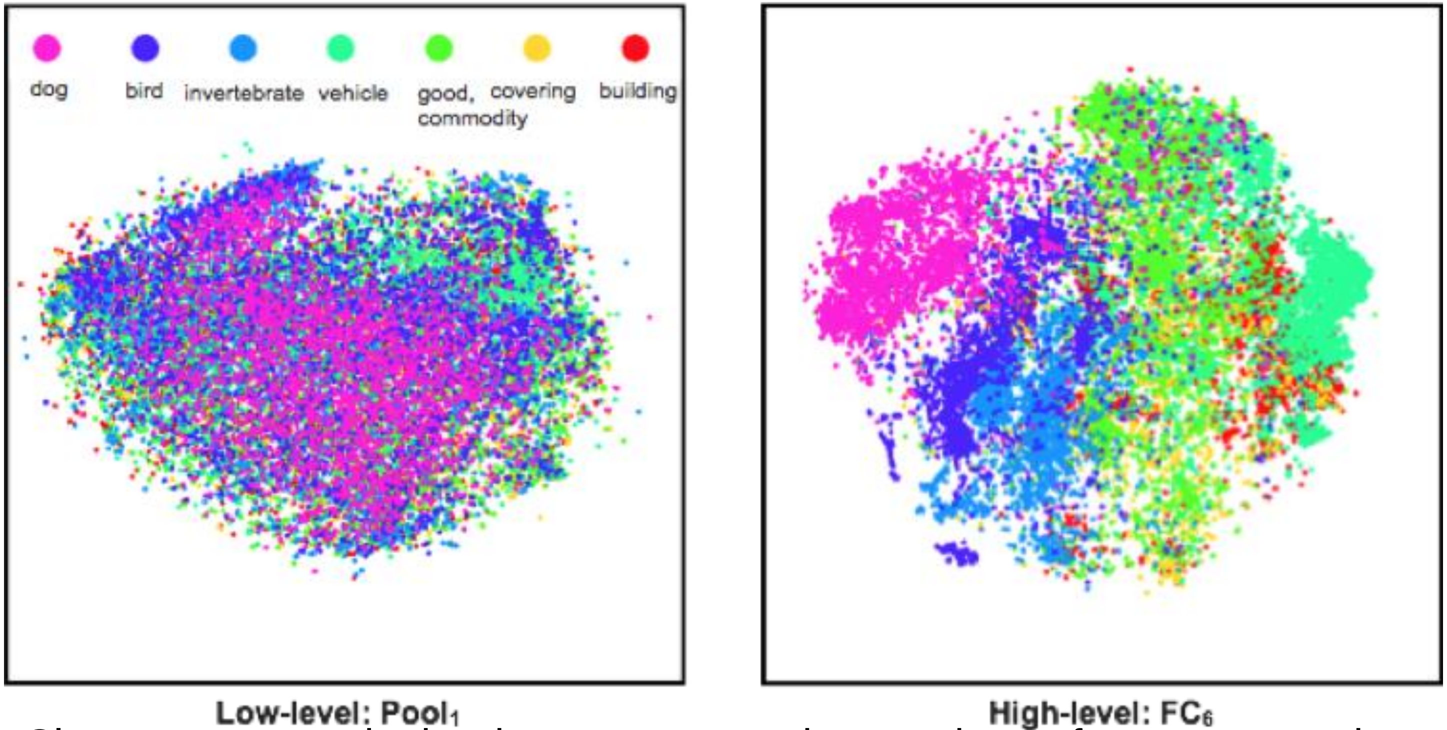
Transfer Learning

"You need a lot of a data if you want to train/use CNNs"

NOT ALWAYS

Deep Features & their Embeddings

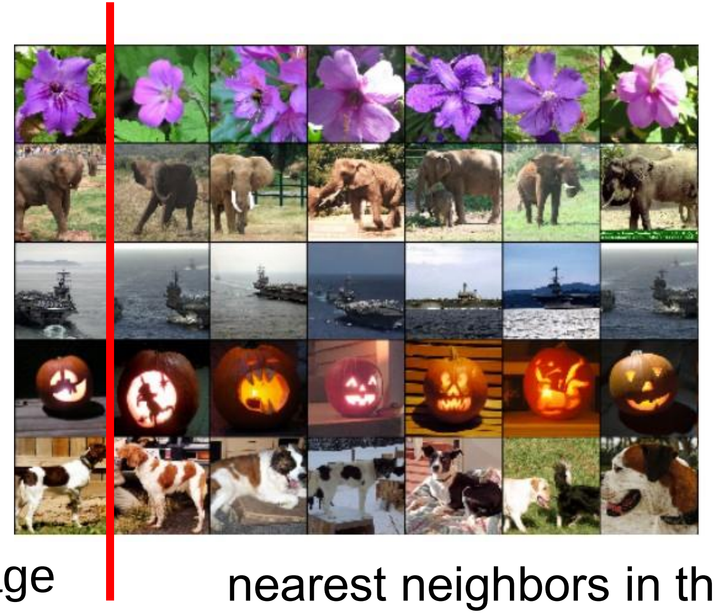
The Unreasonable Effectiveness of Deep Features



Classes separate in the deep representations and transfer to many tasks.
[DeCAF] [Zeiler-Fergus]

Can be used as a generic feature

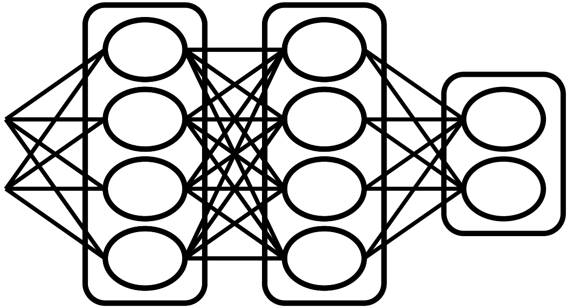
("CNN code" = 4096-D vector before classifier)



query image

nearest neighbors in the "code" space

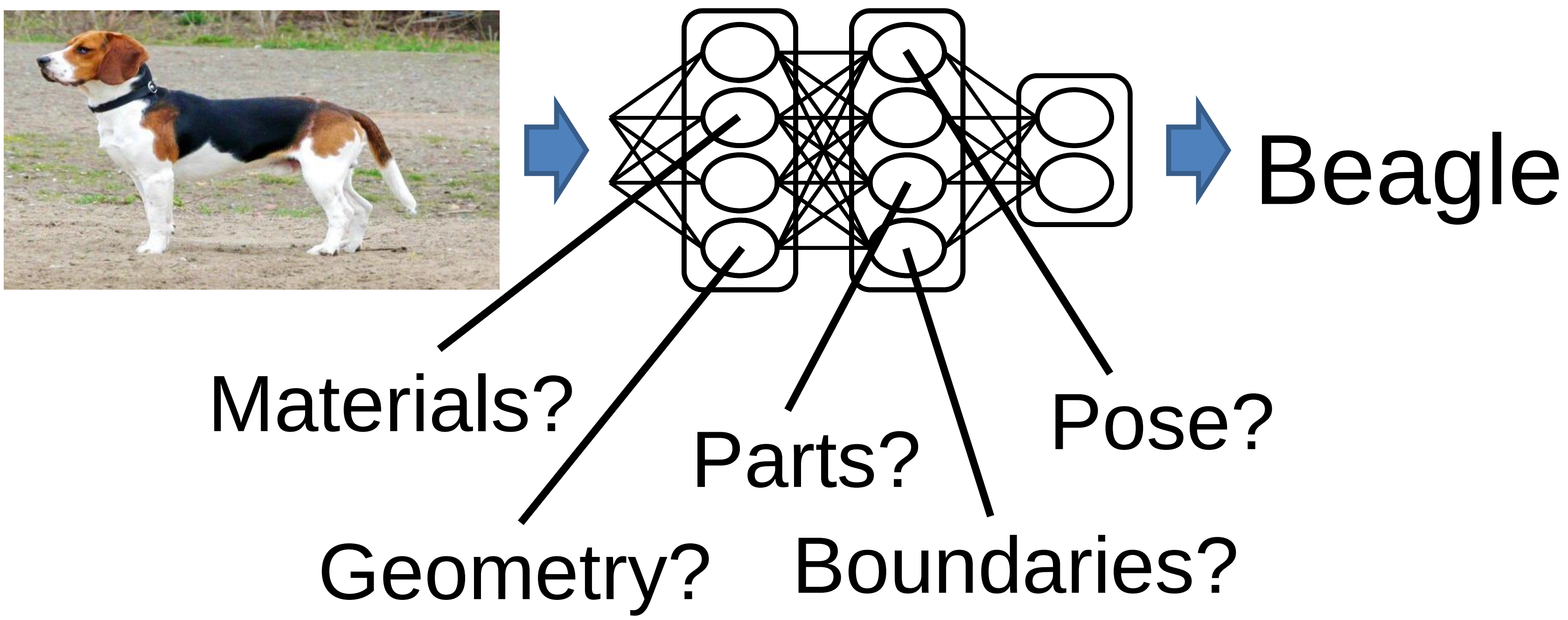
ImageNet + Deep Learning



Beagle

- Image Retrieval
- Detection (RCNN)
- Segmentation (FCN)
- Depth Estimation
- …

ImageNet + Deep Learning

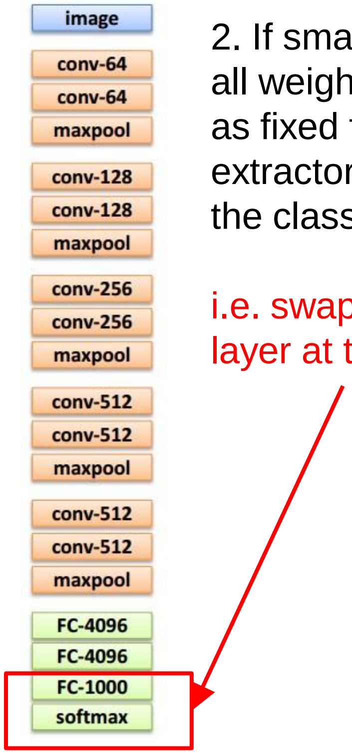


Transfer Learning with CNNs

1. Train on Imagenet

Transfer Learning with CNNs

1. Train on Imagenet



2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

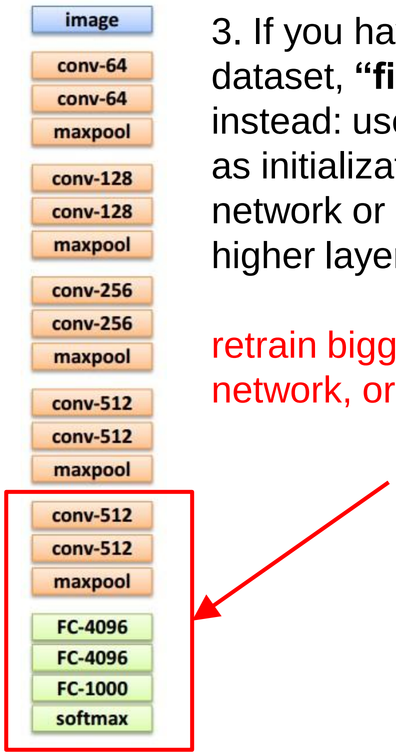
i.e. swap the Softmax layer at the end

Transfer Learning with CNNs

1. Train on Imagenet

2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

i.e. swap the Softmax layer at the end



3. If you have medium sized dataset, **"finetune"** instead: use the old weights as initialization, train the full network or only some of the higher layers

retrain bigger portion of the network, or even all of it.

Transfer Learning with CNNs

1. Train on Imagenet

2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

i.e. swap the Softmax layer at the end

3. If you have medium sized dataset, **"finetune"** instead: use the old weights as initialization, train the full network or only some of the higher layers

retrain bigger portion of the network, or even all of it.

tip: use only ~1/10th of the original learning rate in finetuning to player, and ~1/100th on intermediate layers