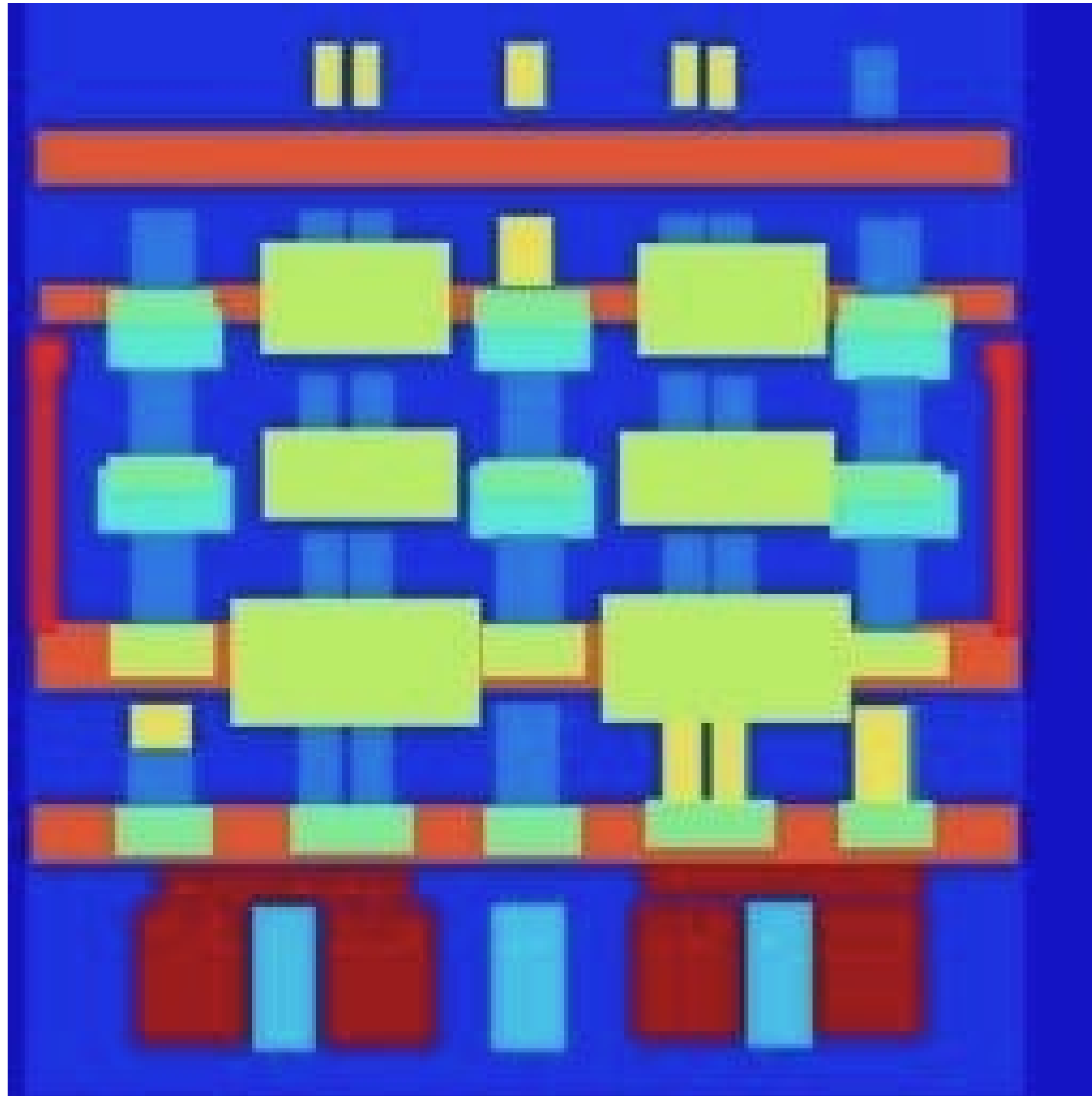


Image-to-Image Translation



CS180: Intro to Comp. Vision and Comp. Photo
Efros & Ng, UC Berkeley, Fall 2026

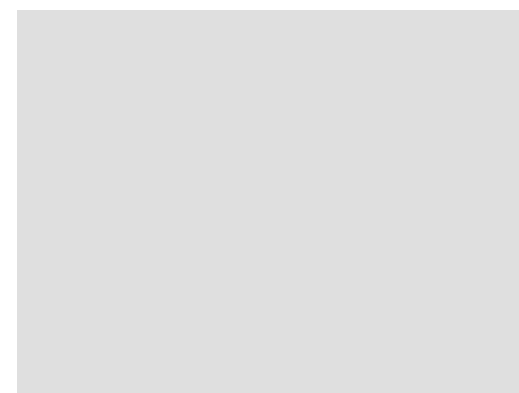
Visual Similarity is hard

$$42 - 24 = 18$$

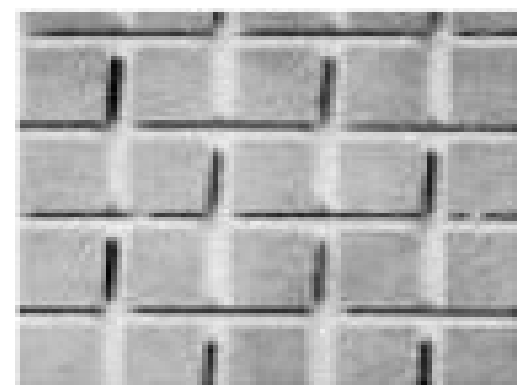
$$SLUMP - \bar{T}RUMP = \text{edit distance of 2 letters}$$



-



$$= \text{distance of 50 grey values}$$



-



$$= ?$$

When are two textures similar?



Finding Good Visual Similarity

- Learn from Biology
 - V1-inspired filter banks
- Learn from Data
 - Use labels (supervised)
 - Use patterns from data itself (self-supervised)

Visual similarity via labels



Machine Learning as data association



image X



black box
classifier



“Penguin”

label Y

At test time...



image X



black box
classifier



?

Deep Learning: a high-capacity classifier



image X



Neural
Network



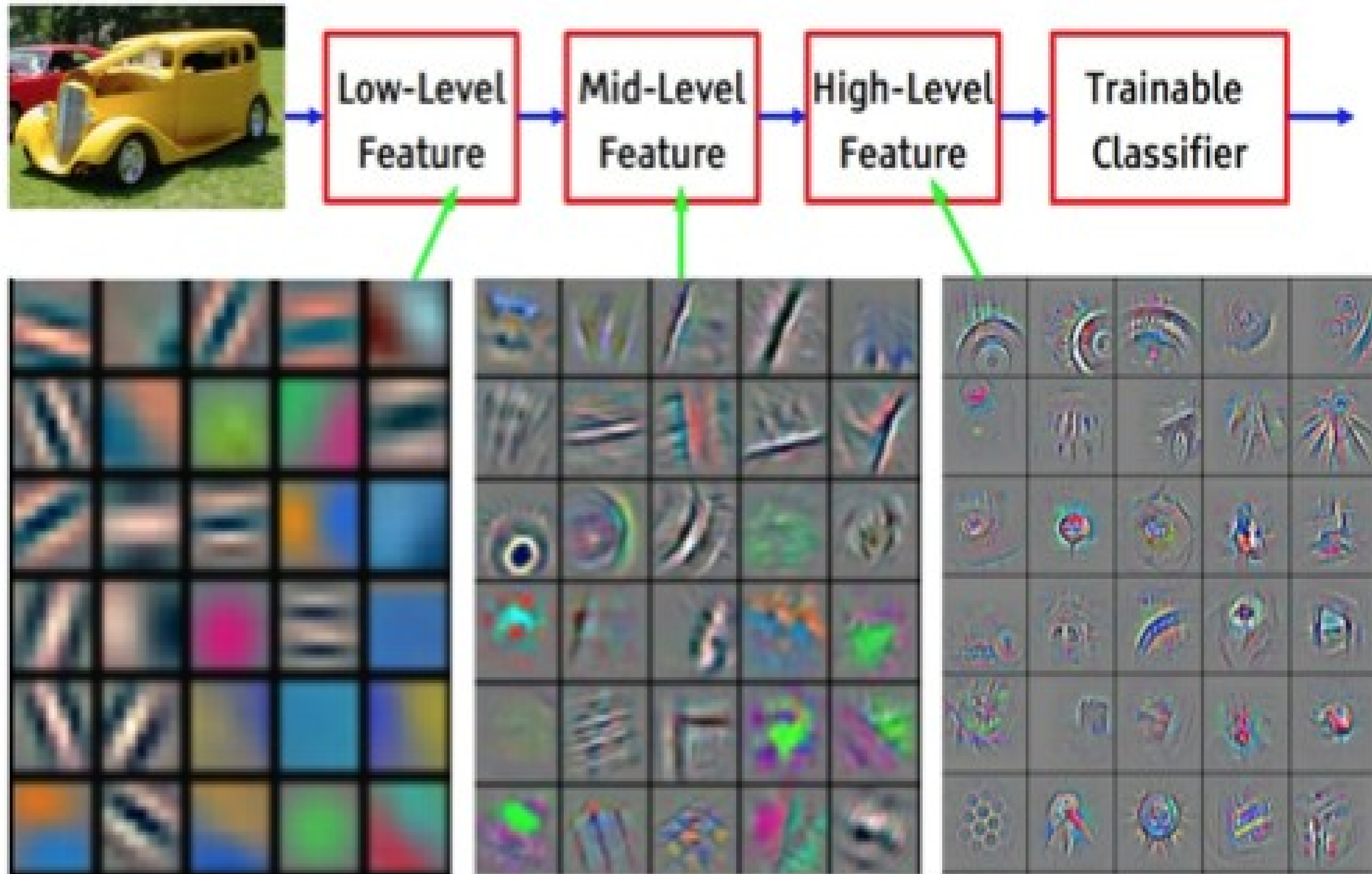
**Billions of
Paramete**



"Penguin"

label Y

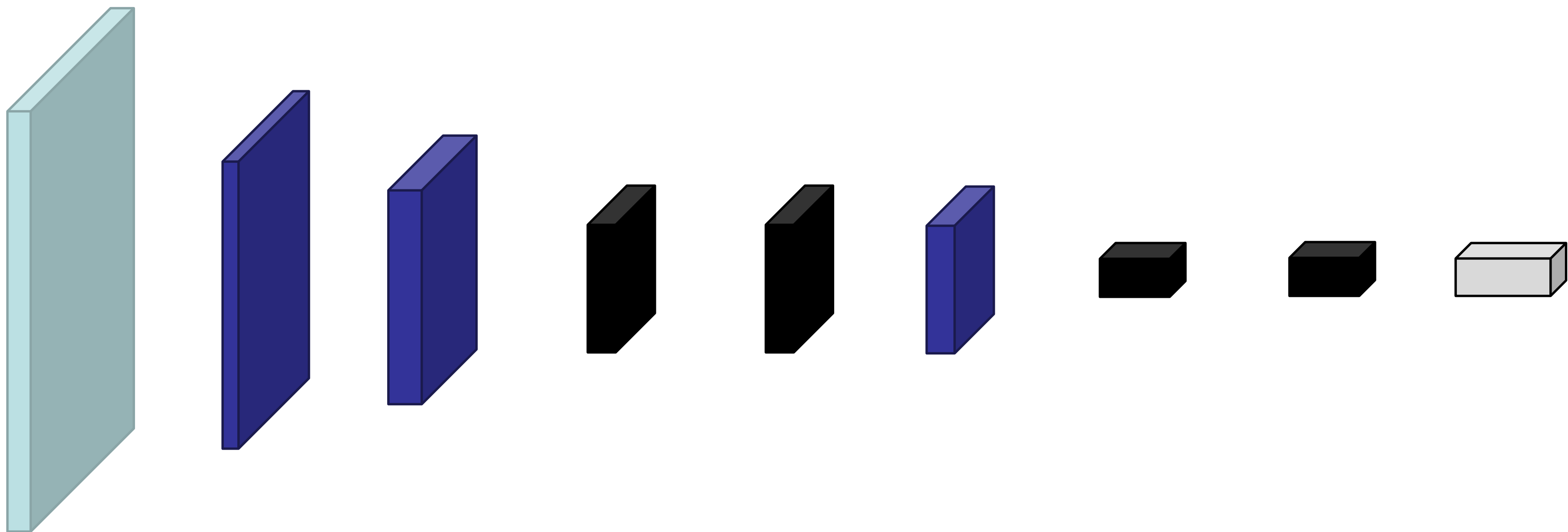
Convolutional Neural Networks



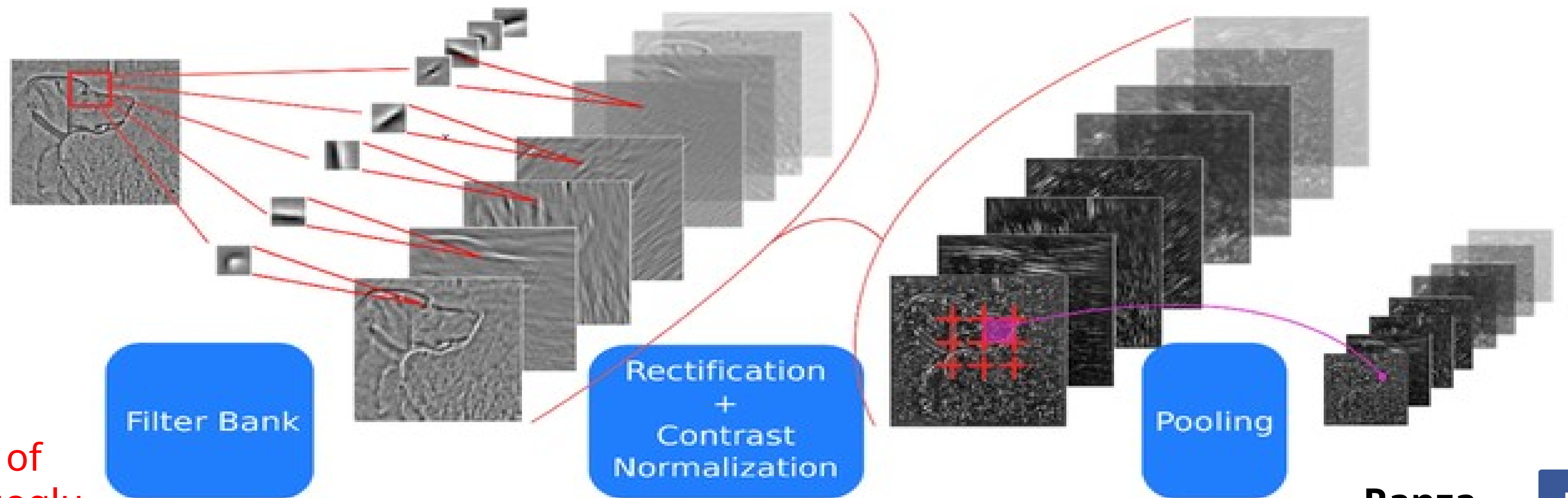
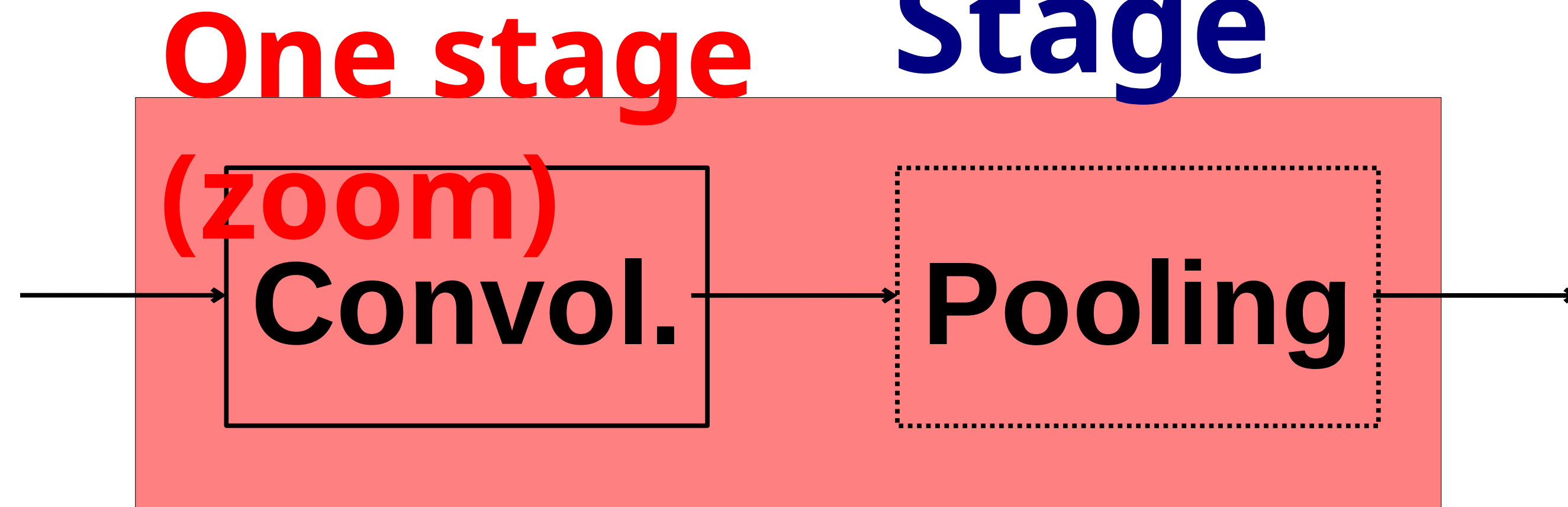
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Alexnet

Input	Conv 1	Conv 2	Conv 3	Conv 4	Conv 5	FC 6	FC 7	Output
227x227x3	55x55x96	27x27x128	13x13x48	13x13x128	13x13x128	1x1x4096	1x1x4096	1x1x1000



ConvNets: Typical Stage

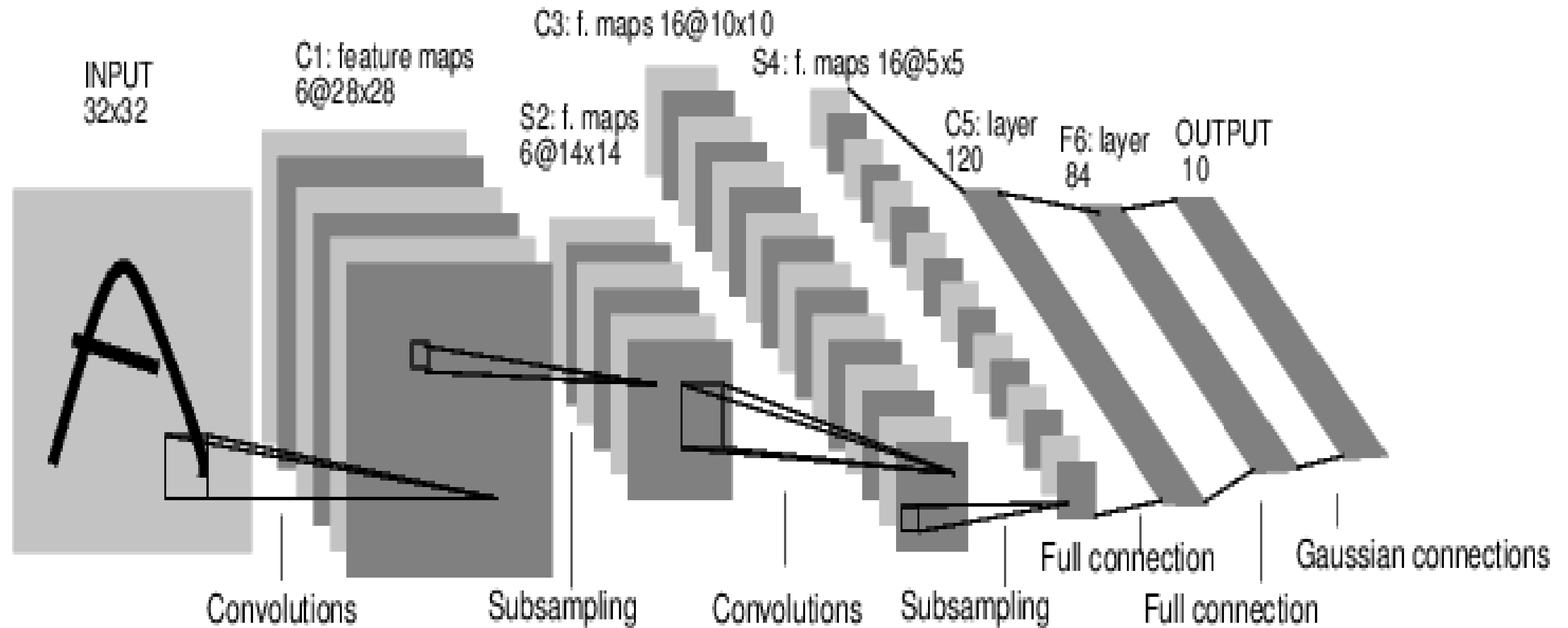


courtesy of
K. Kavukcuoglu

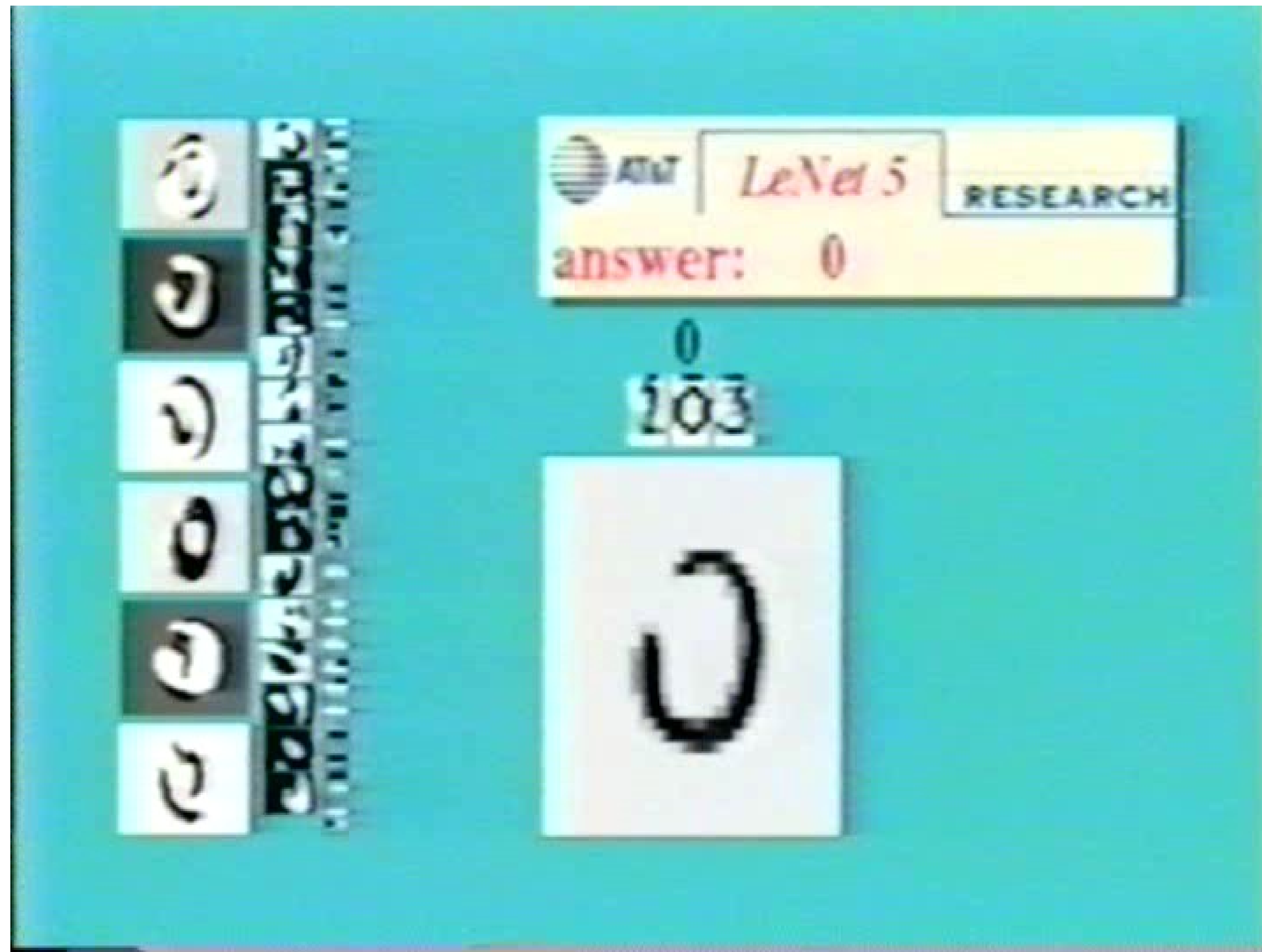
Ranza



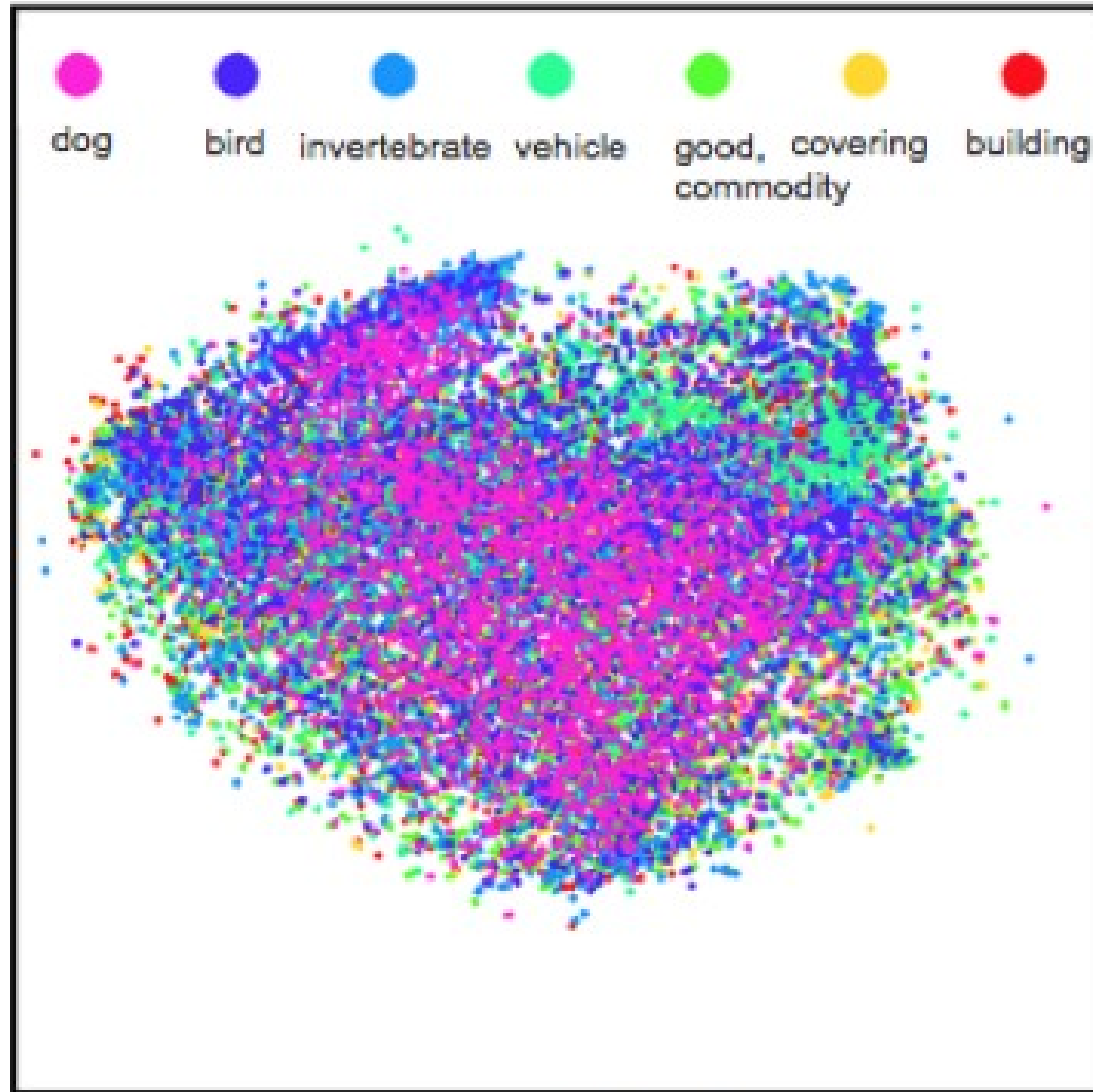
LeNet



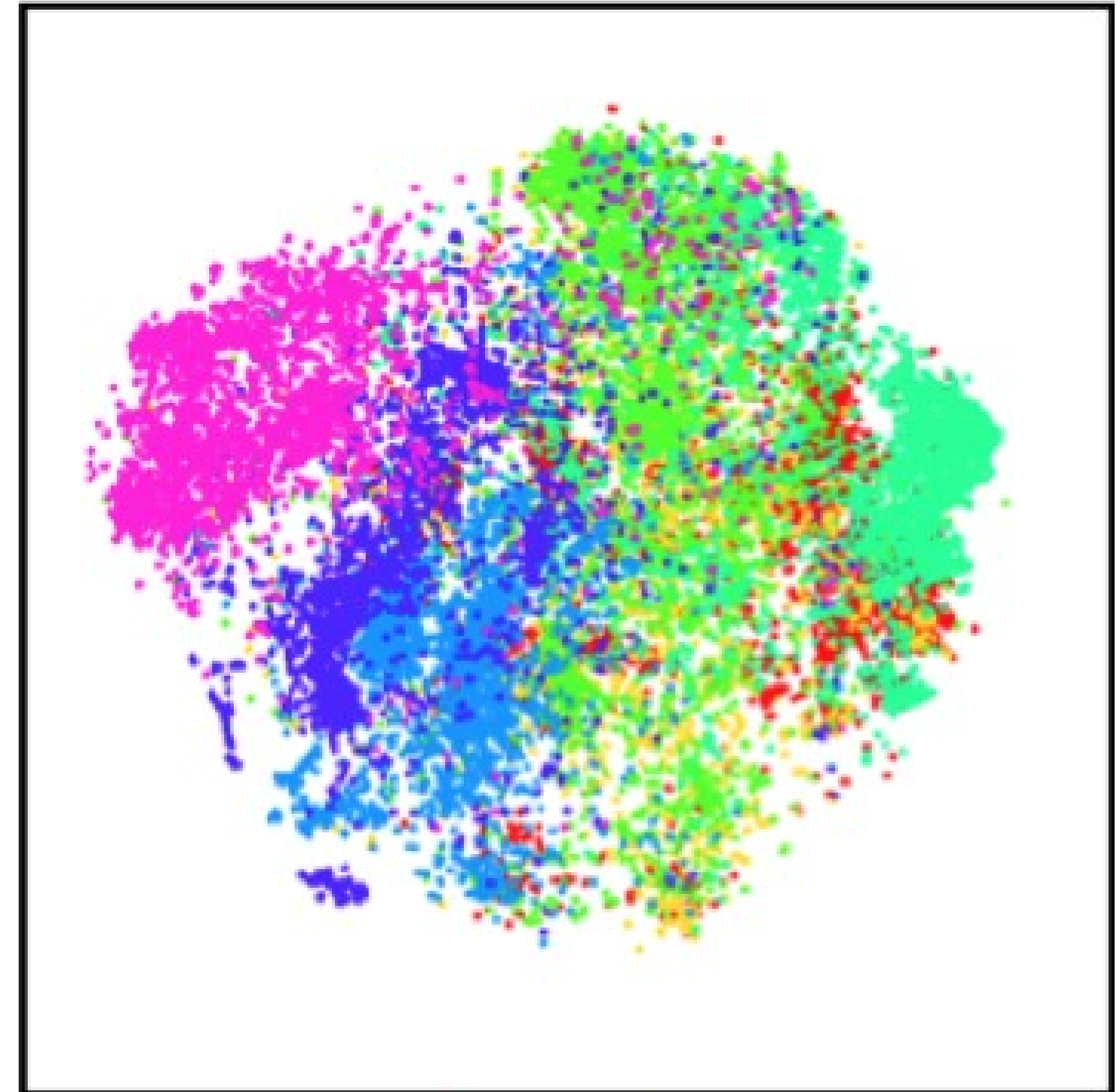
Handwritten digit classification



The Unreasonable Effectiveness of Deep Features



Low-level: Pool₁

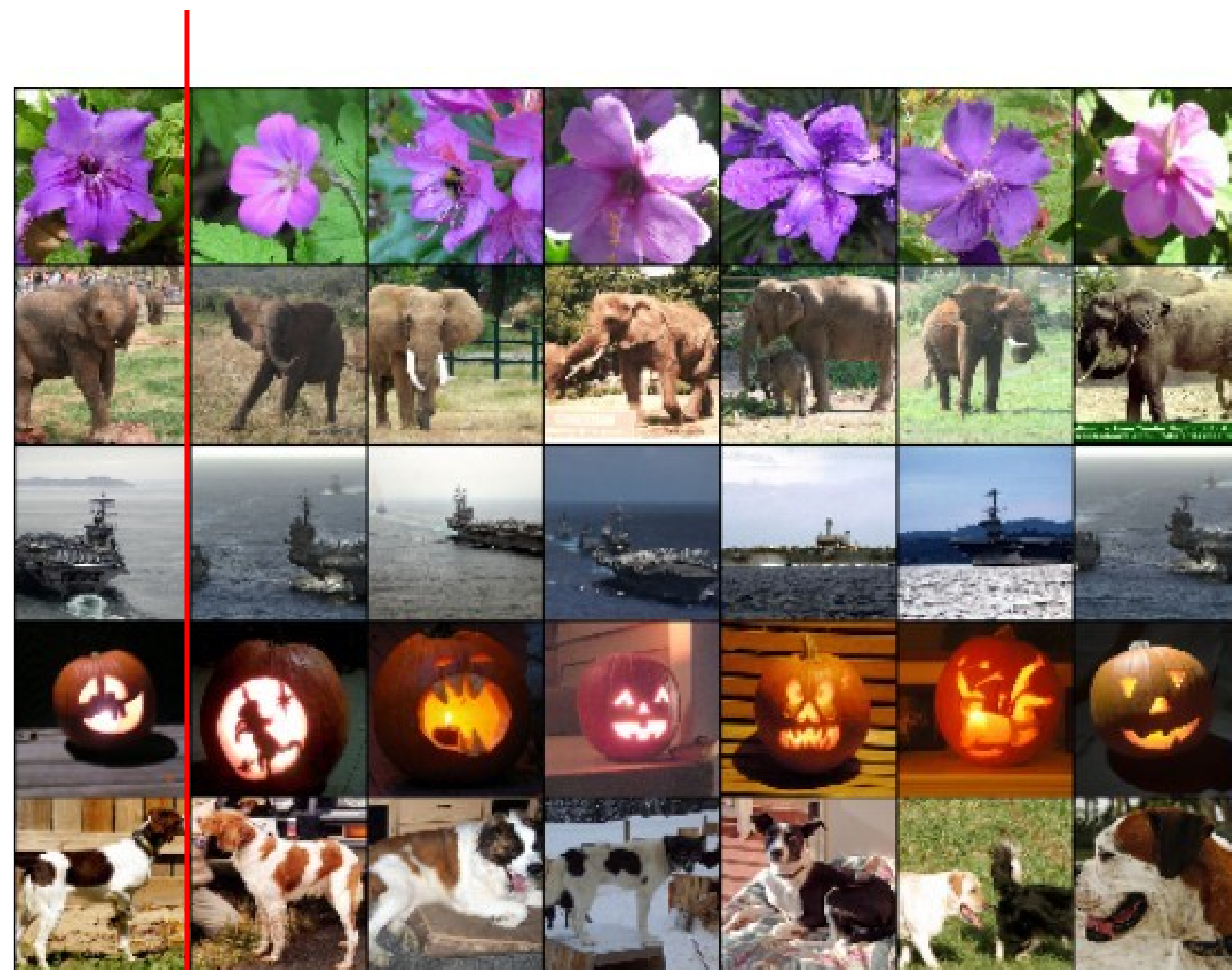


High-level: FC₆

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

Can be used as a generic feature

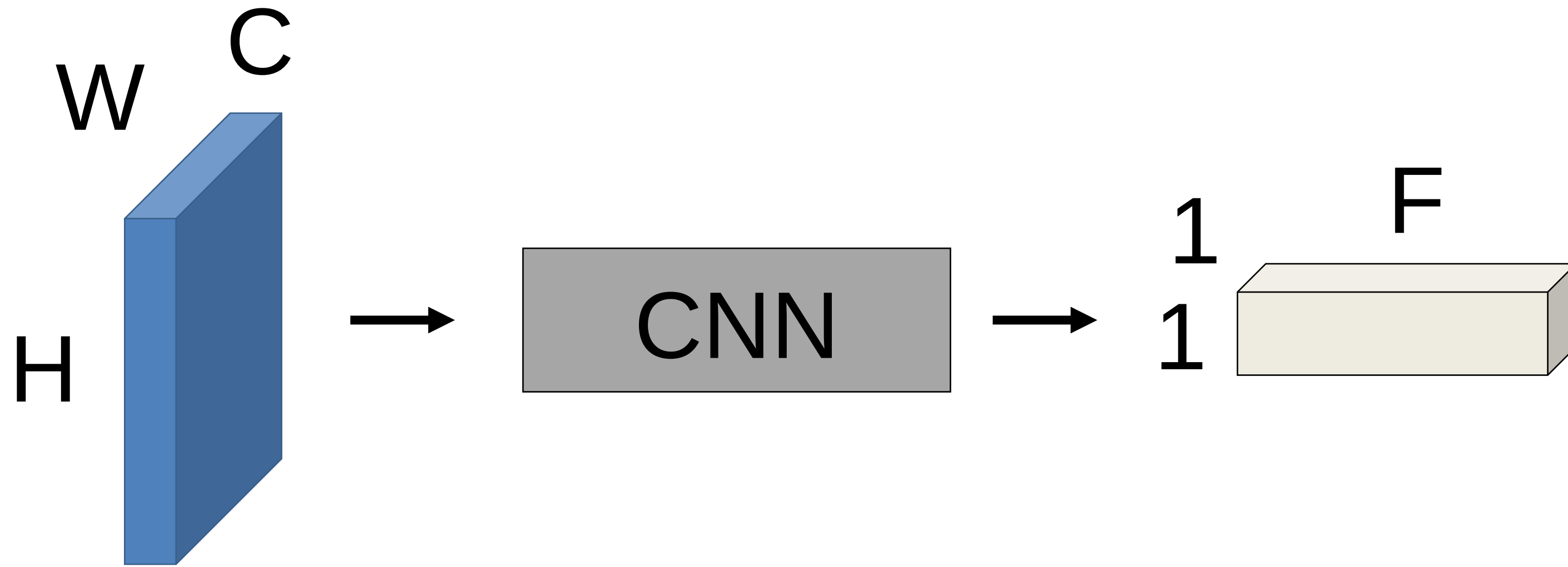
(“CNN code” = 4096-D vector before classifier)



query image

nearest neighbors in the “code” space

Recall CS189/CS182



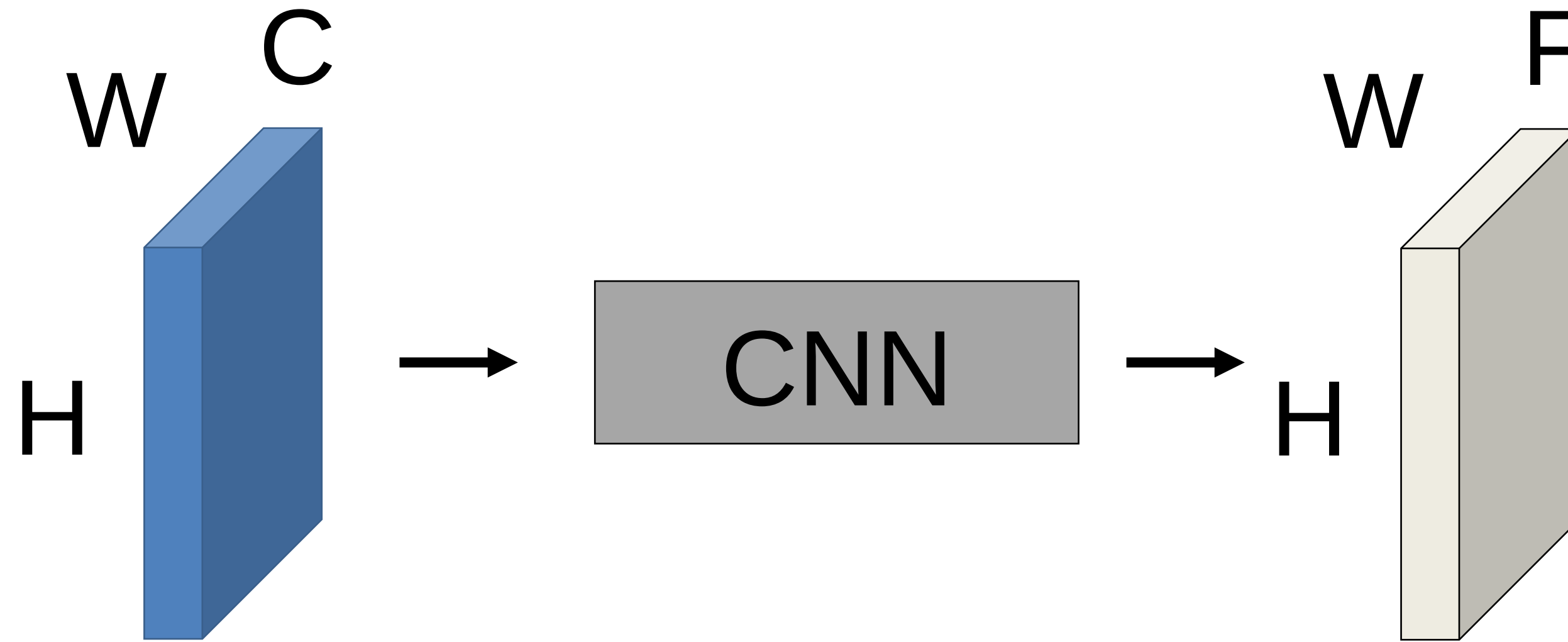
Convert $H \times W$ image into a F -dimensional vector

Is this image a cat?

At what distance was this photo taken?

Is this image fake?

Pixel Labeling



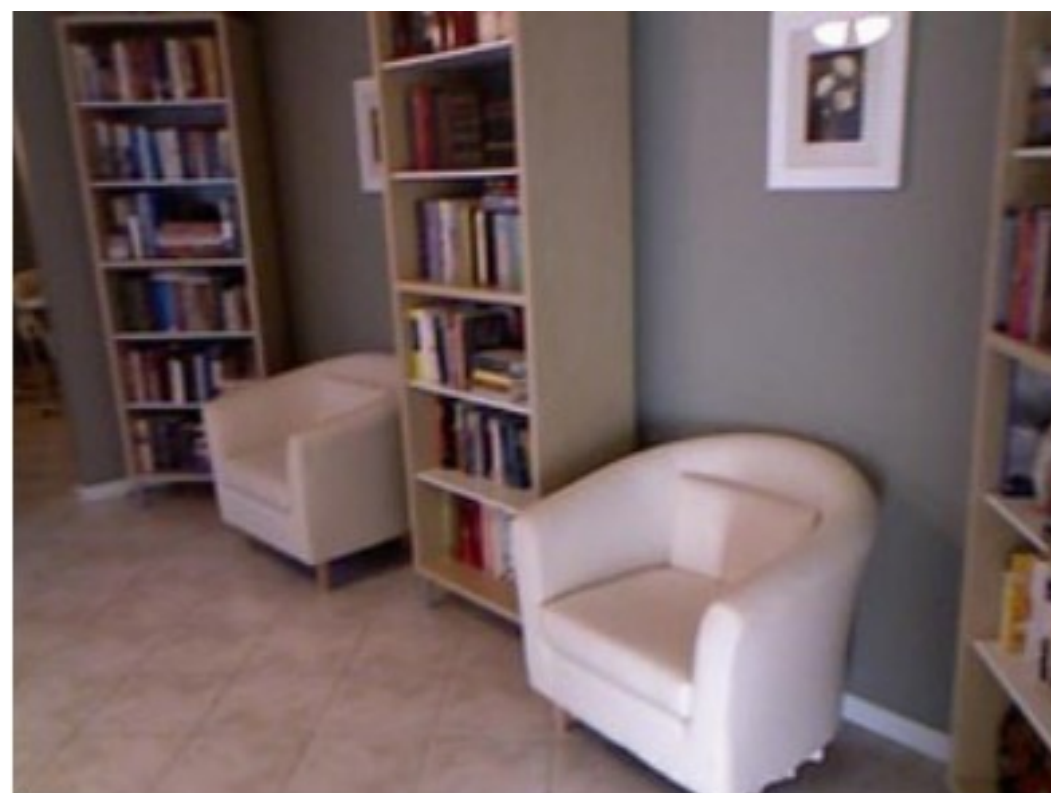
Convert $H \times W$ image into a F -dimensional vector

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

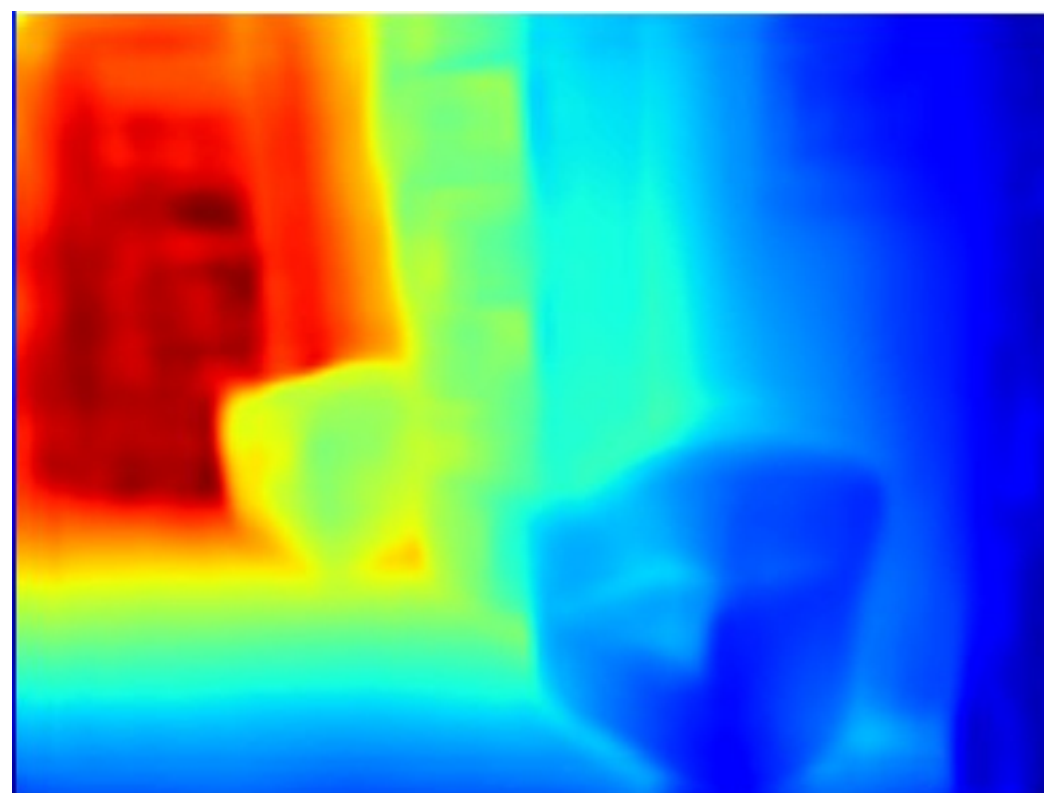
e.g. Depth Prediction

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).

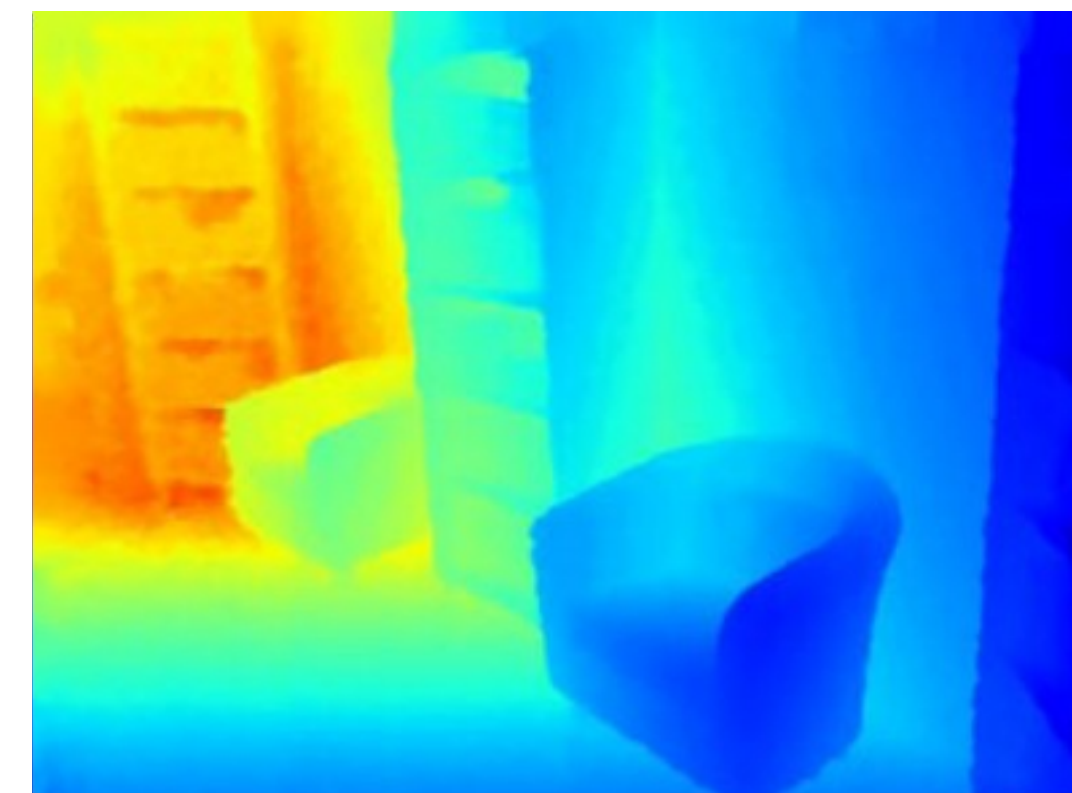
Input HxWx3
RGB Image



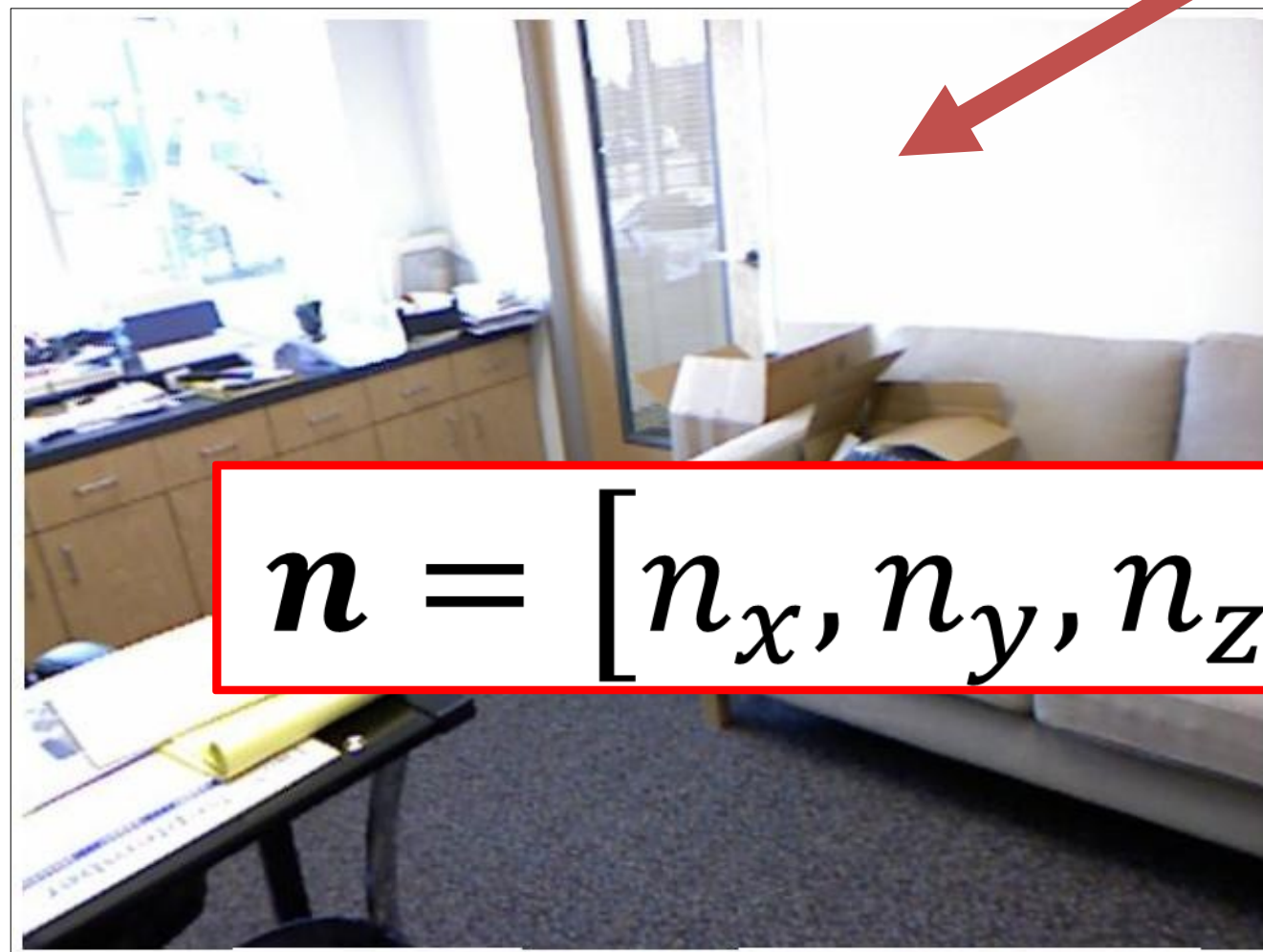
Output HxWx1
Depth Image



True HxWx1
Depth Image

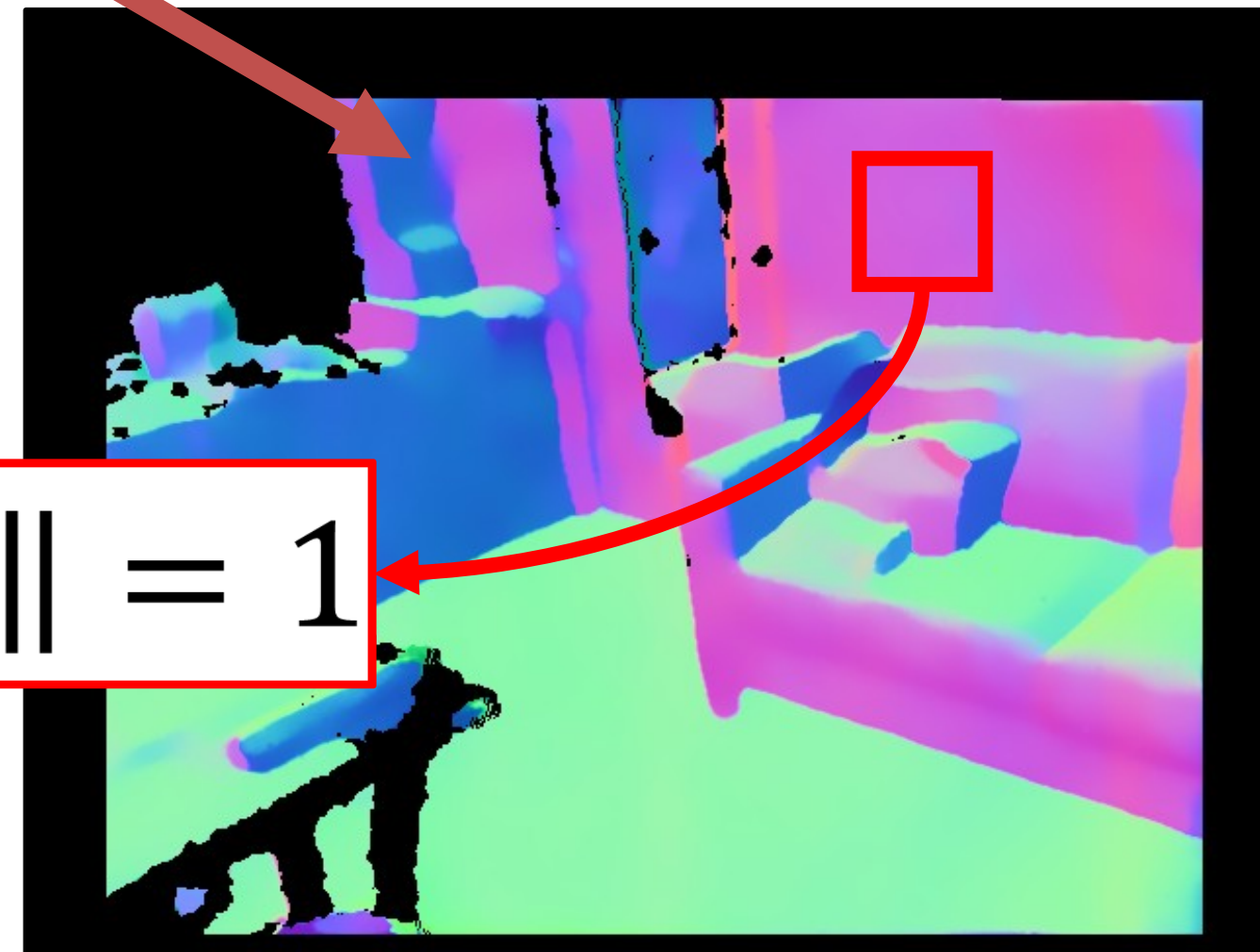


Surface Normals



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

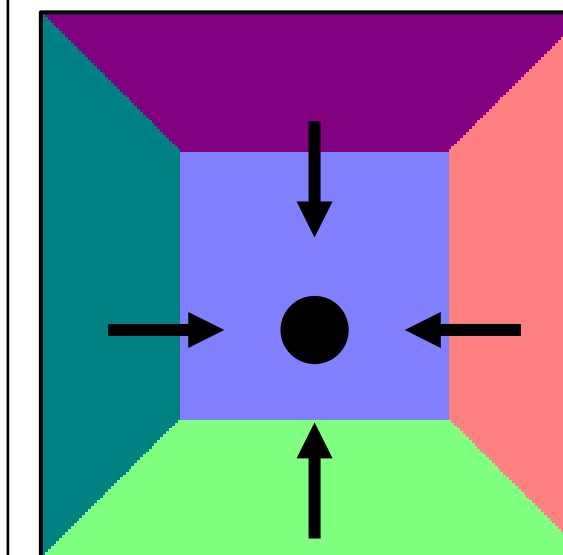
Color Image



Normals



Room



Legend

Surface Normals

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

Input: HxWx3
RGB Image



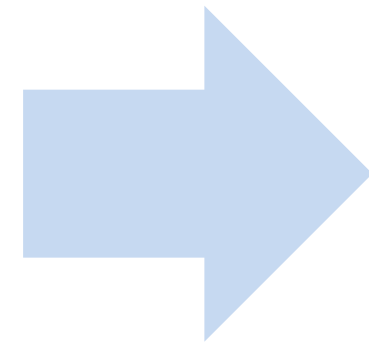
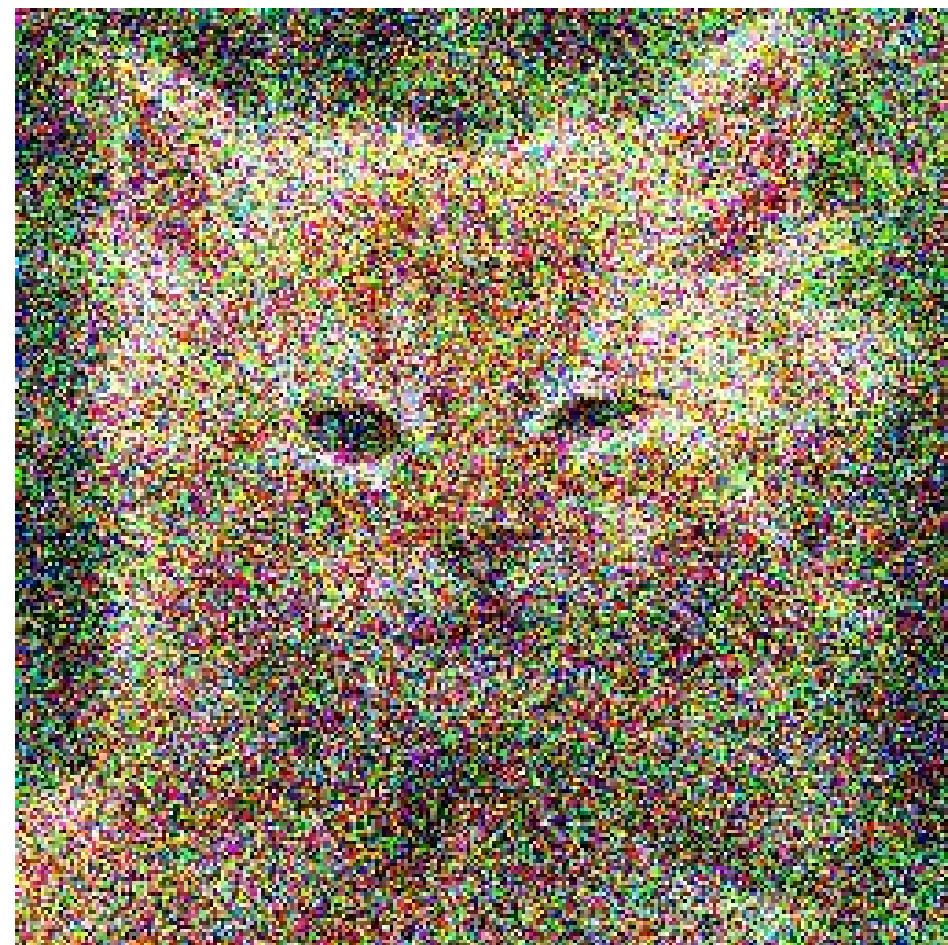
Output: HxWx3
Normals



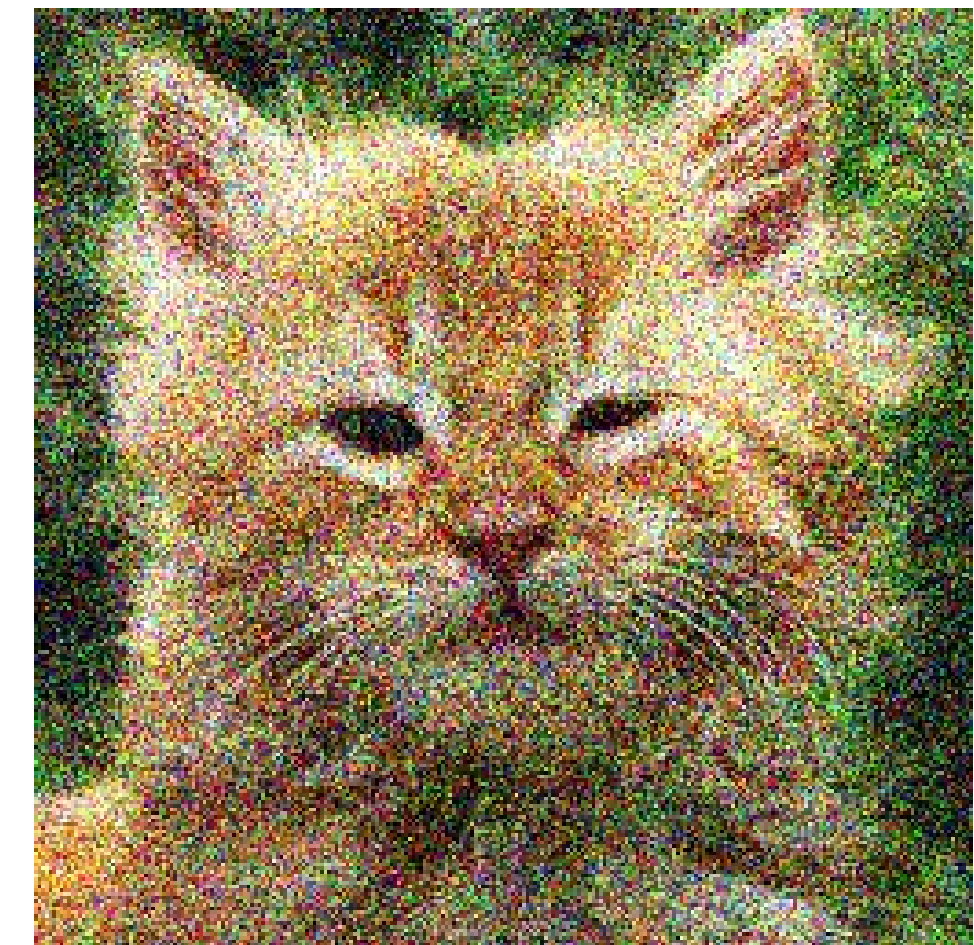
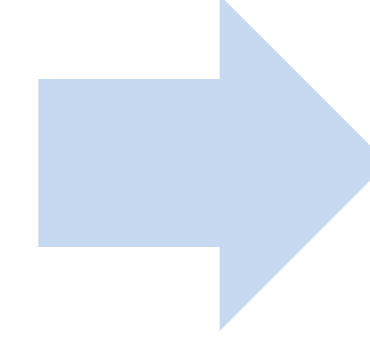
Result credit: X. Wang, D. Fouhey, A. Gupta, *Designing Deep Networks for Surface Normal Estimation*. CVPR 2014

Slide by David Fouhey

Image Denoising



**Denoising
neural
network**



“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



Generic: Image-to-Image Translation

Labels to Street Scene

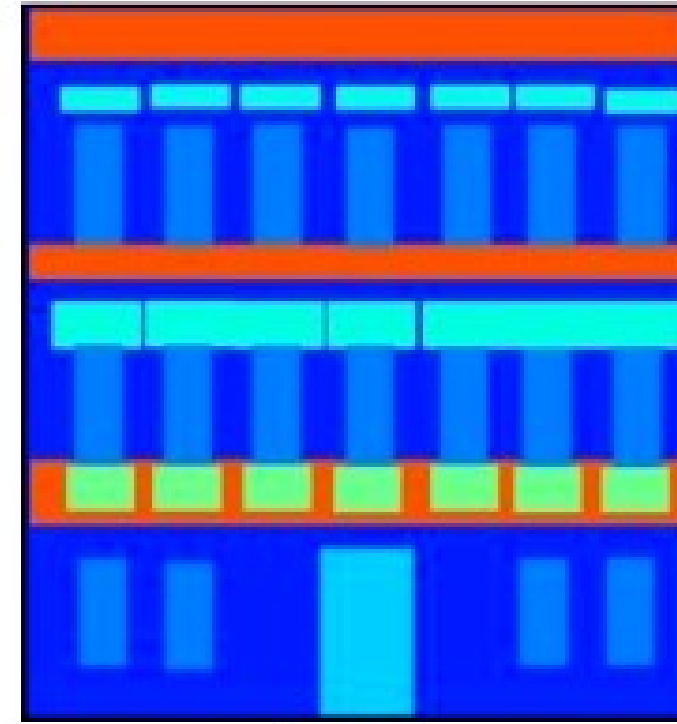


input



output

Labels to Facade



input



output

BW to Color



input



output

Aerial to Map



input



output

Day to Night



input



output

Edges to Photo



input

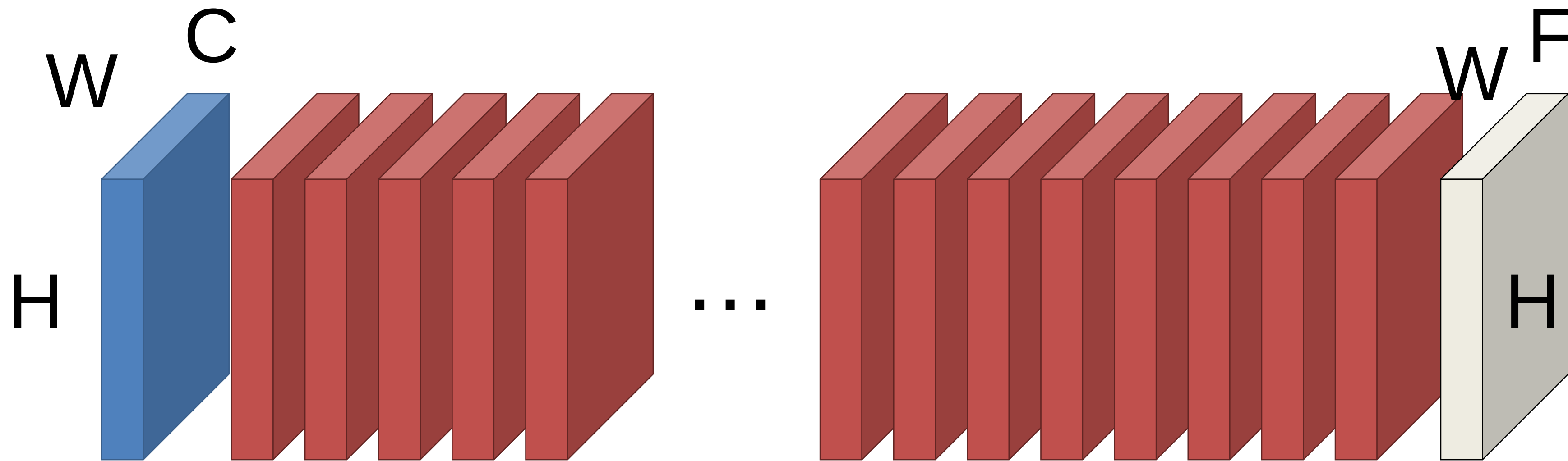


output

First – Two “Wrong” Ways

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

Why Not Stack Convolutions?



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

How many convolutions until ≥ 200 pixels?

100

Idea #2

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

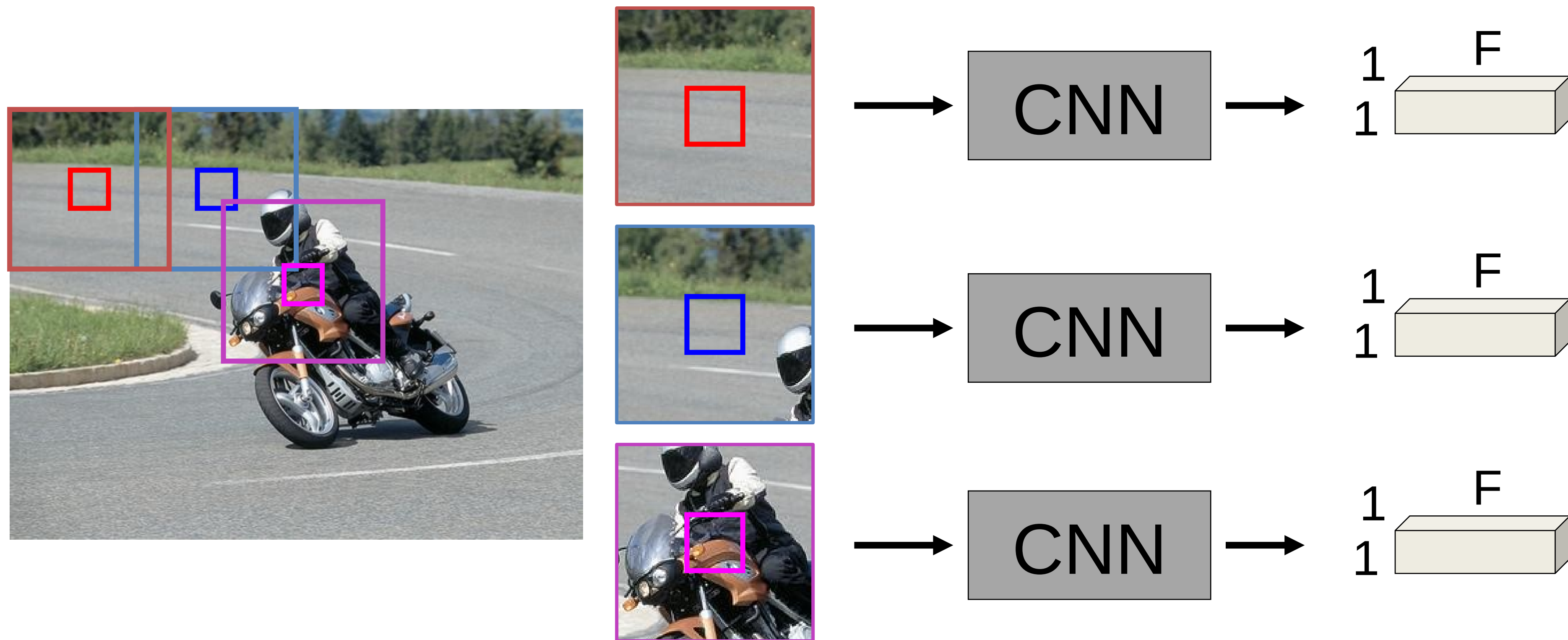
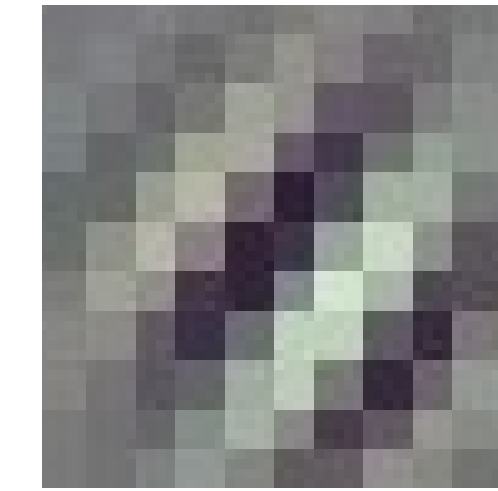


Image credit: PASCAL VOC, Everingham et al.

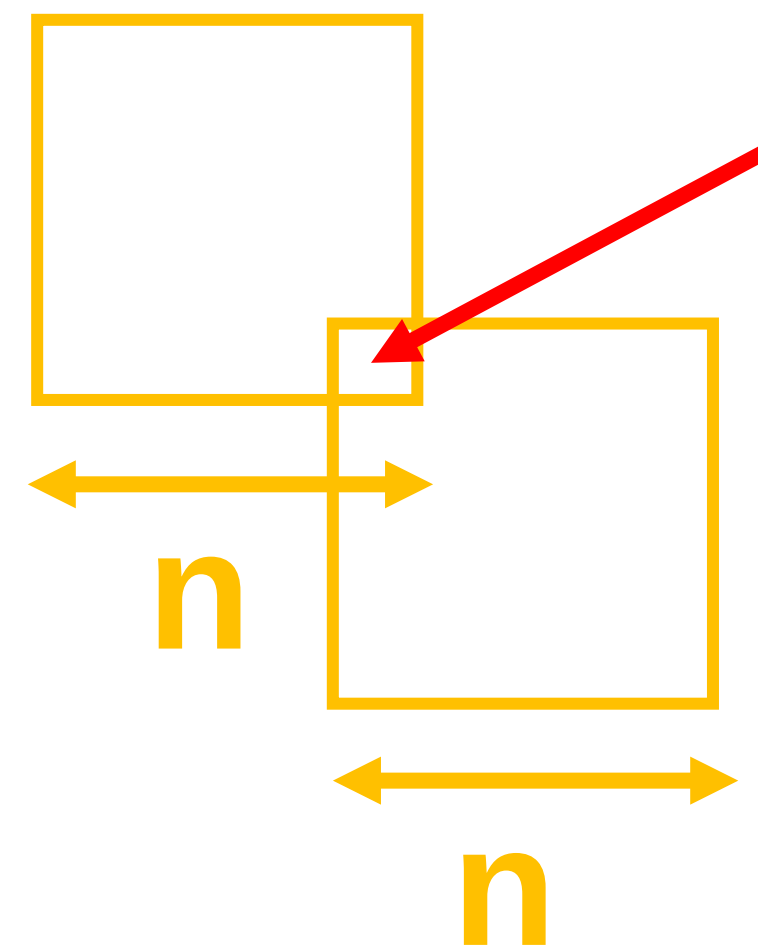
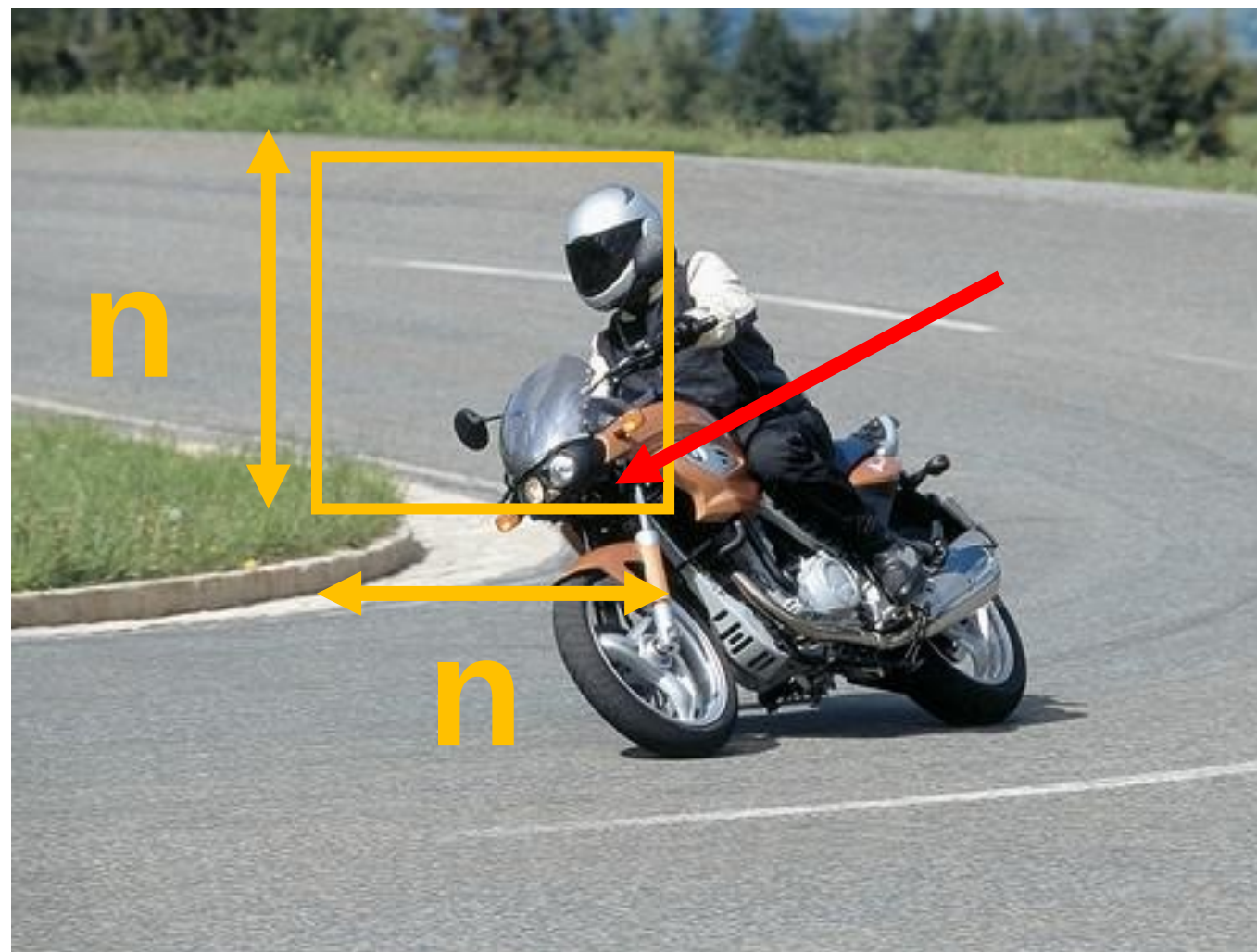
Slide by David Fouhey

Idea #2

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



Gabor



Answer:
 $(2n-1) \times (2n-1)$

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

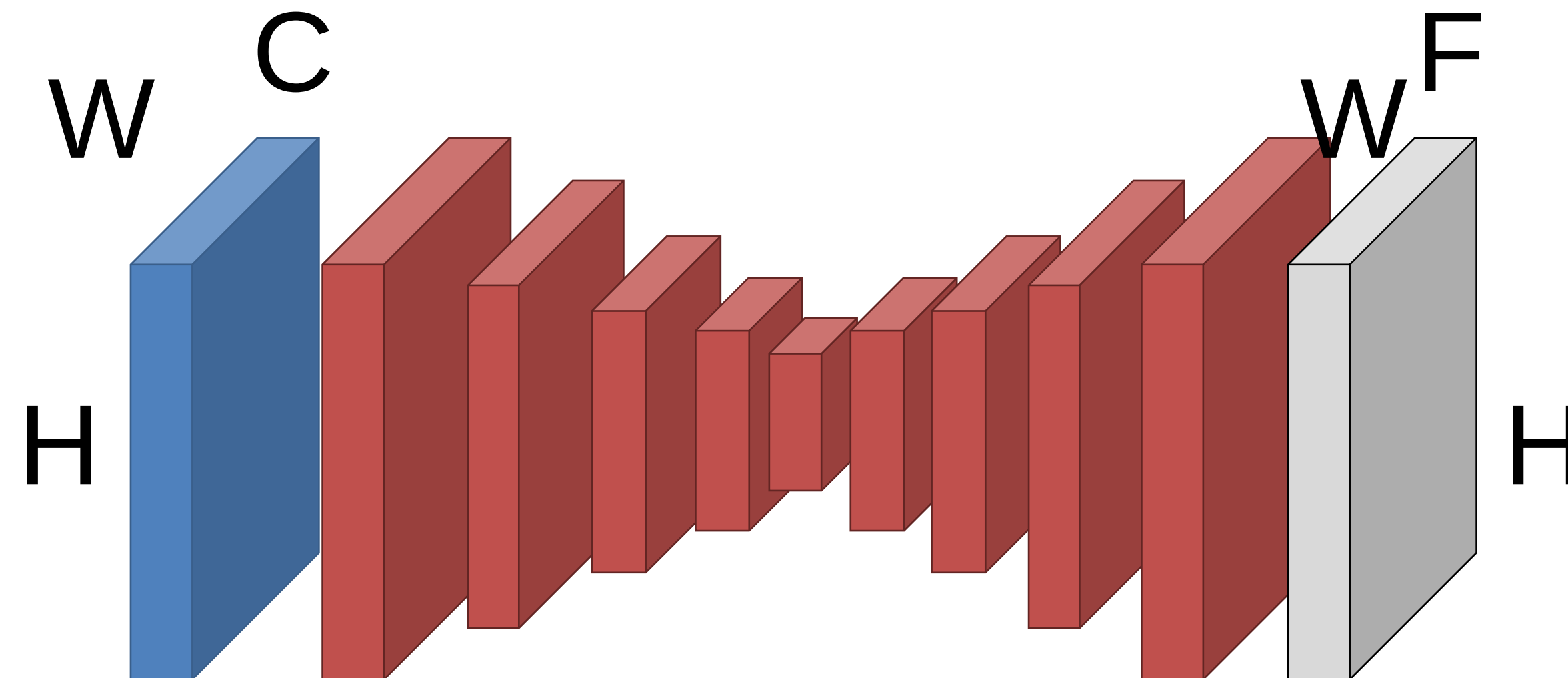
These two objectives are in total conflict

Encoder-Decoder

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

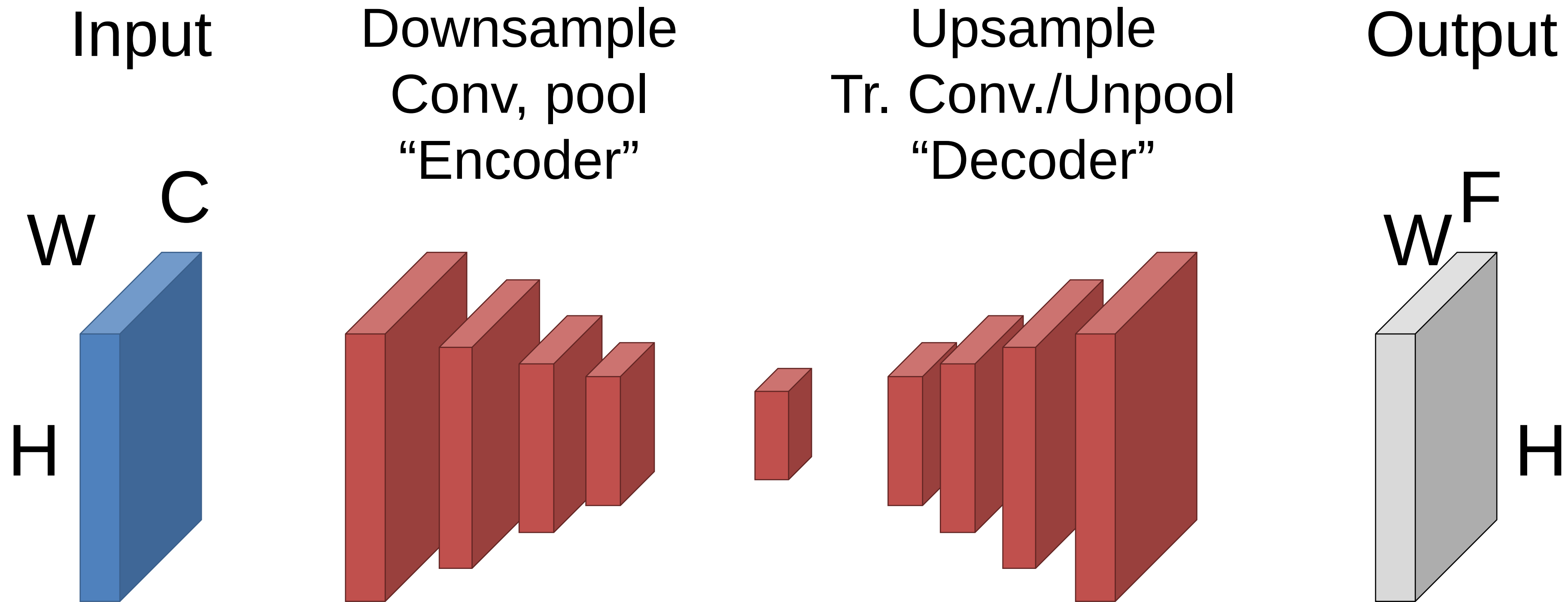
How do we downsample?

Convolutions, pooling



Putting it Together

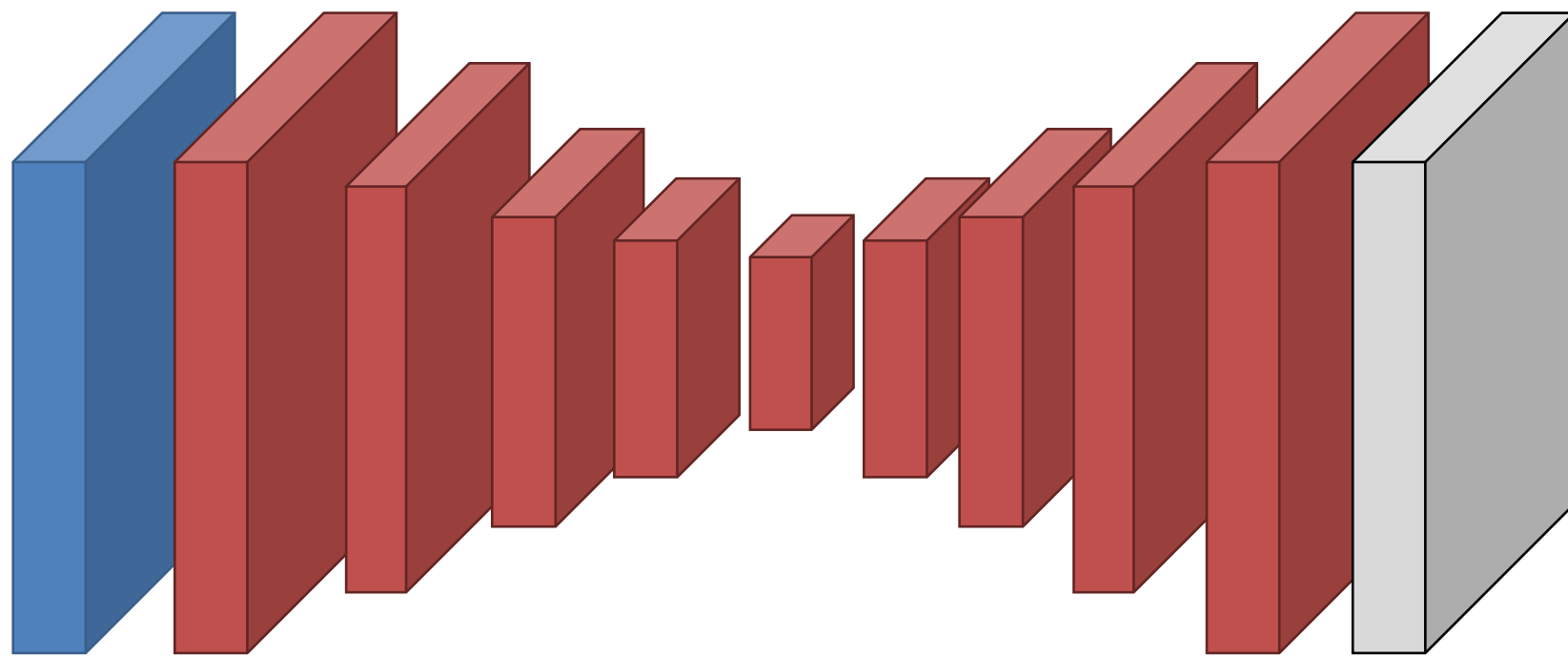
Convolutions + pooling downsample/compress/encode
Transpose convs./unpoolings upsample/uncompress/decode



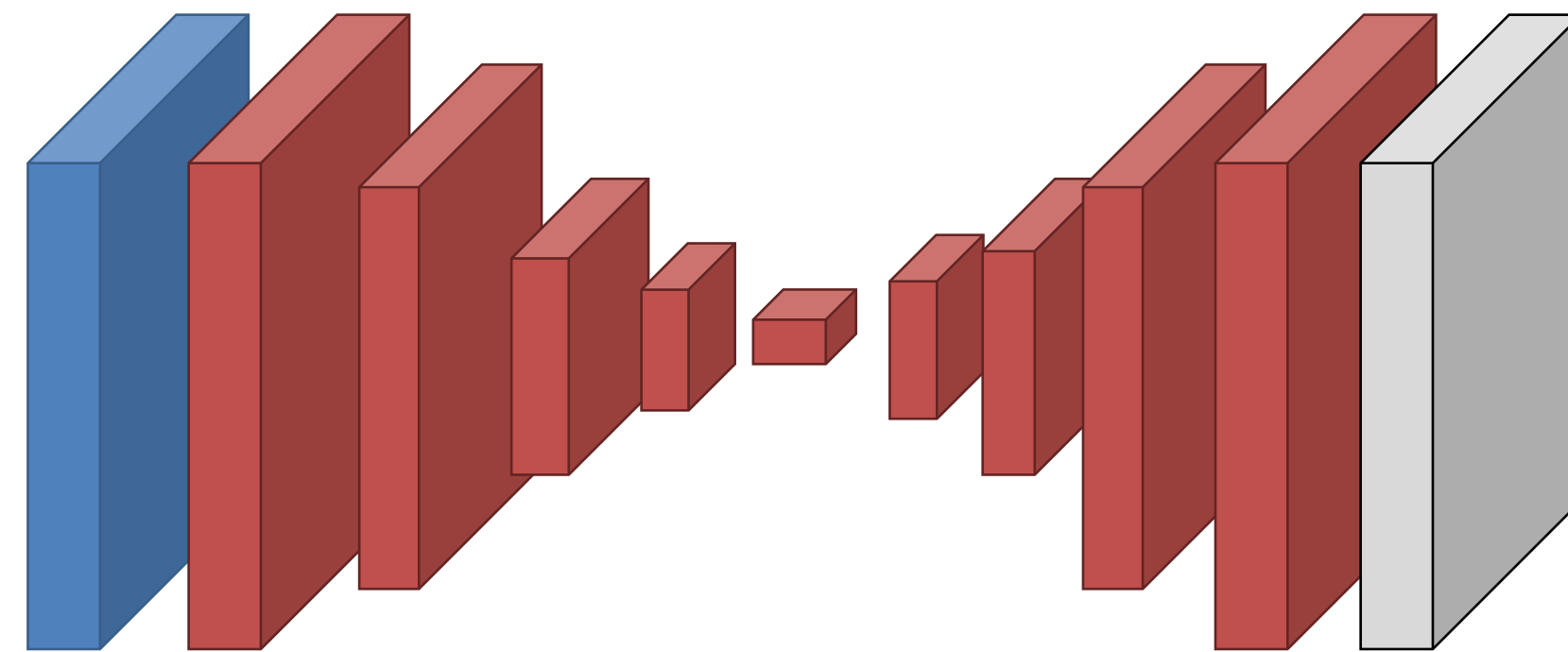
Putting It Together – Block Sizes

- 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



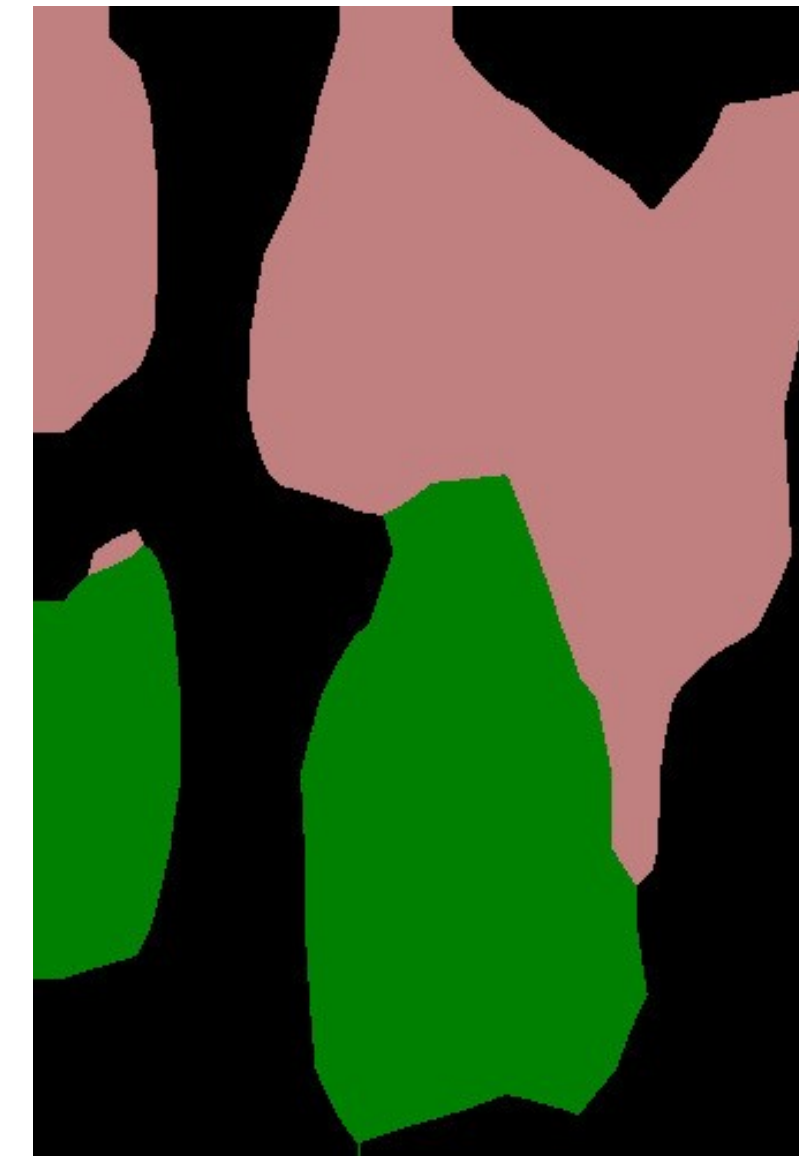
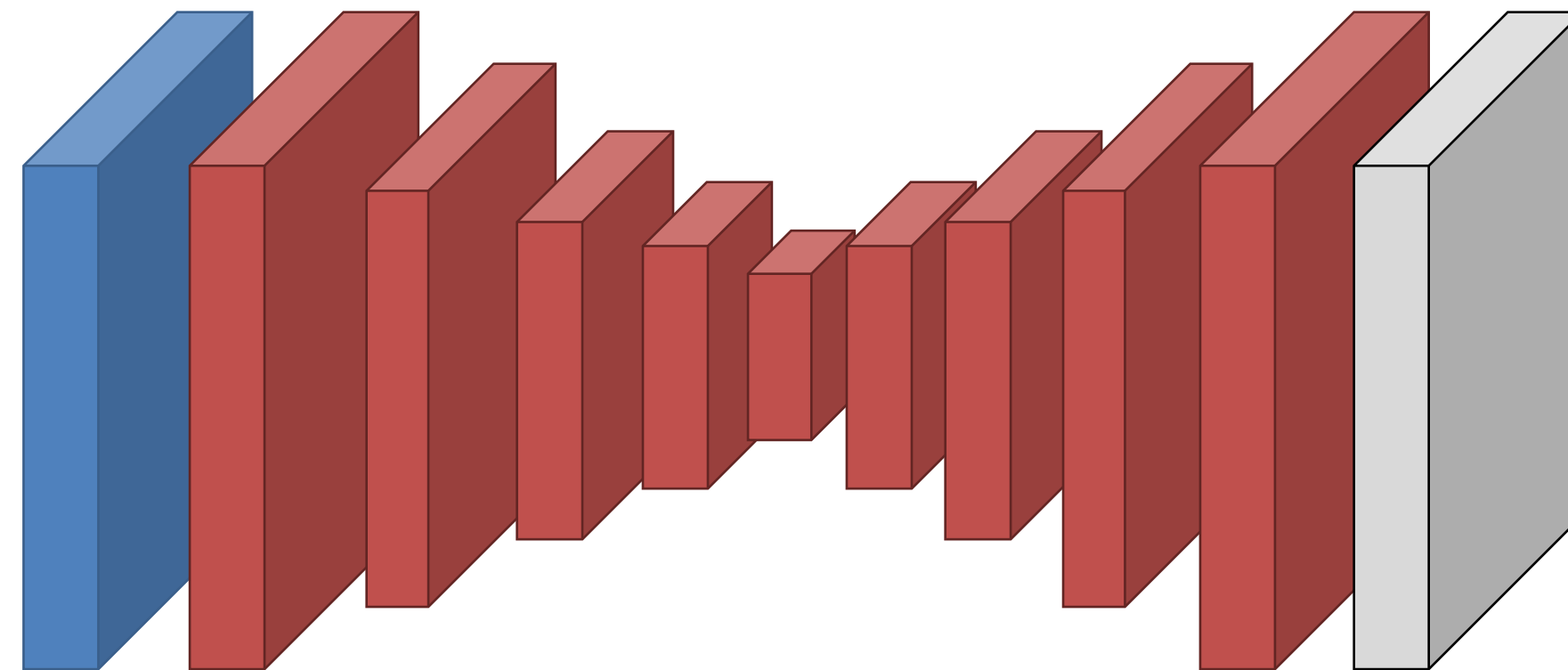
Missing Details

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

Why?

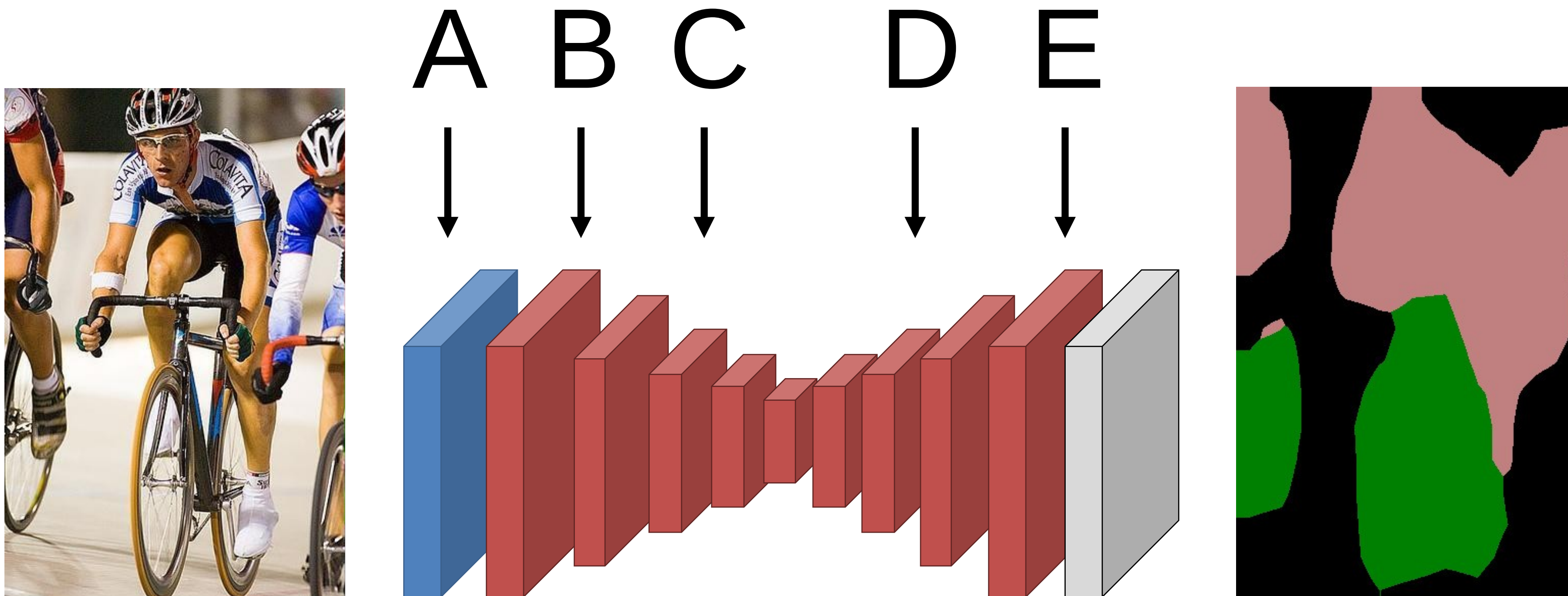


Information about details
lost when downsampling!



Missing Details

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

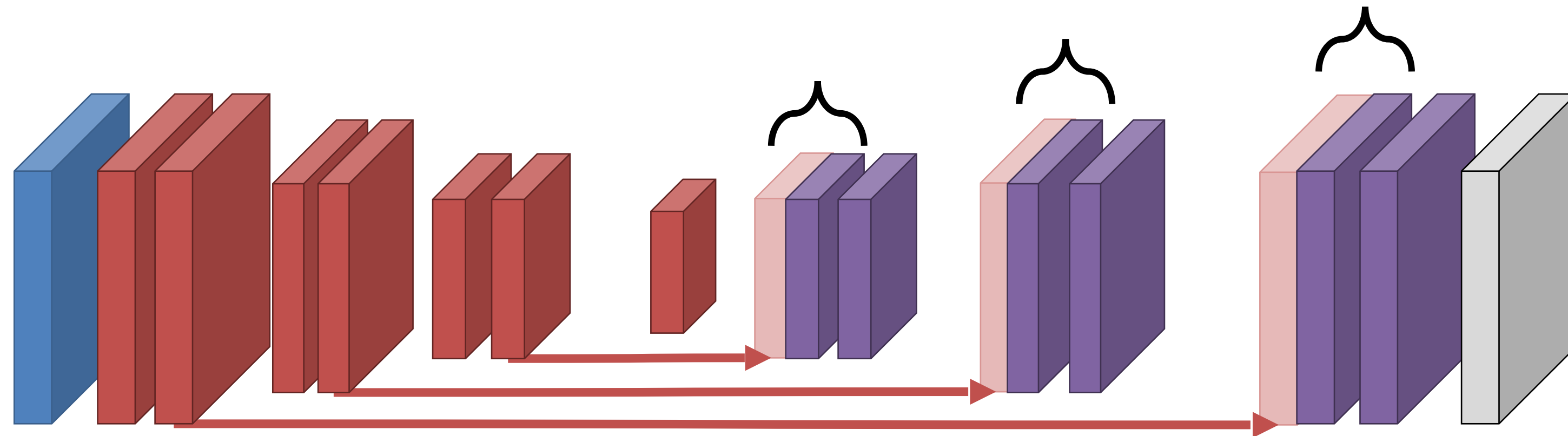


Missing Details

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.

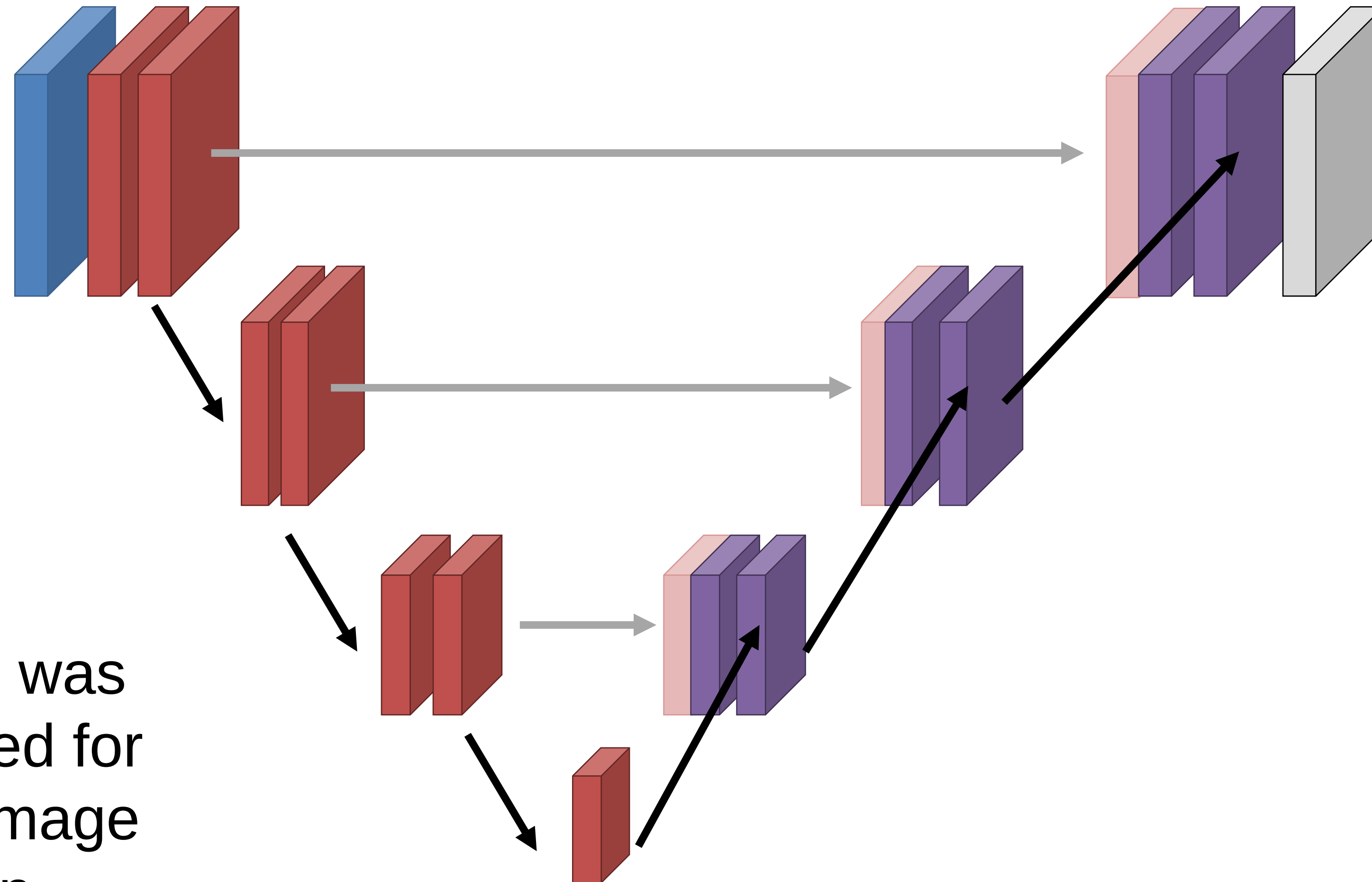


Copy

U-Net

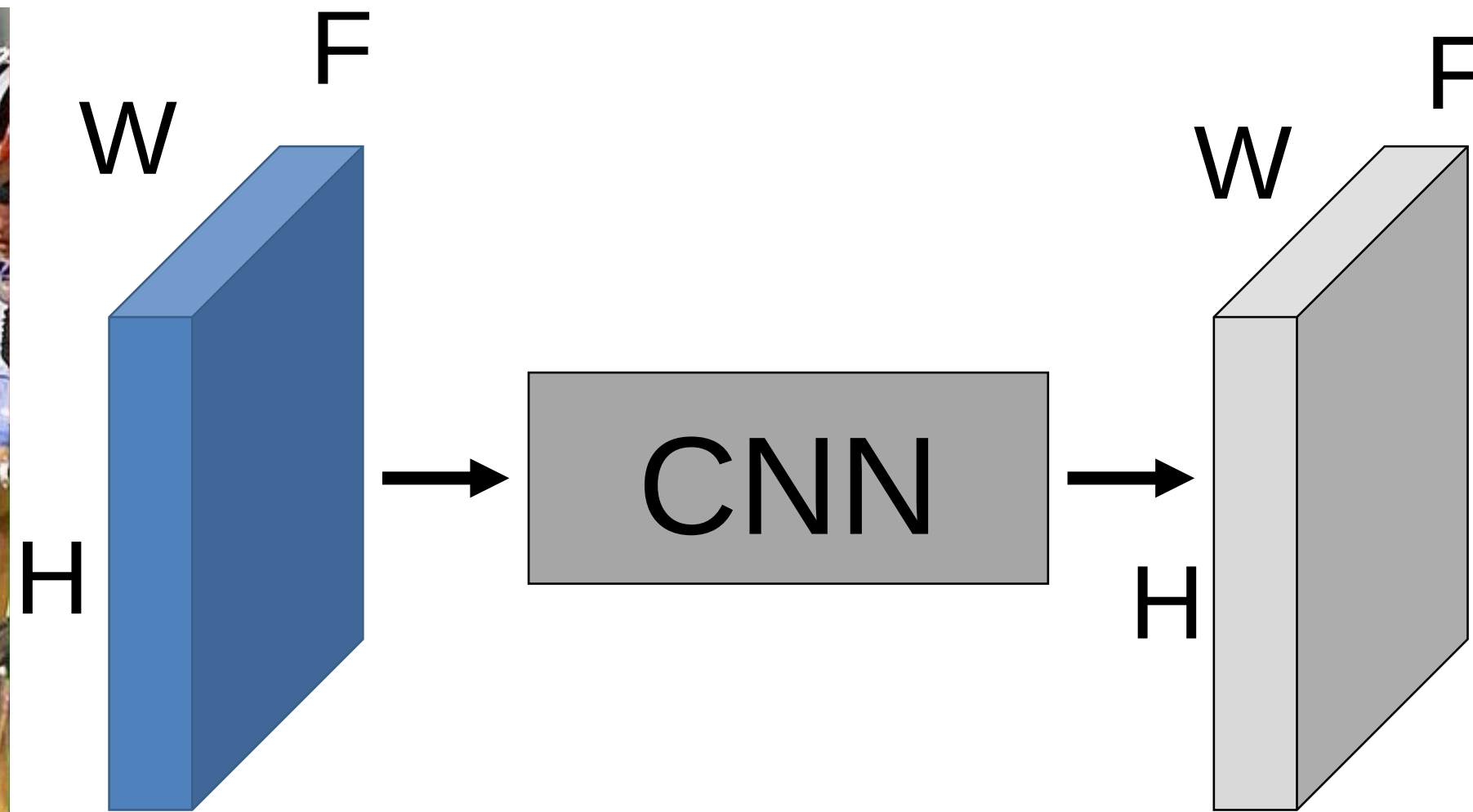


Extremely popular architecture, was originally used for biomedical image segmentation.



U-Net improves performance

Input
Image



Predicted
Classes



ResNet: Residual Learning

New Building Block: $x + F(x)$

Lets you train networks with 100s of layers.

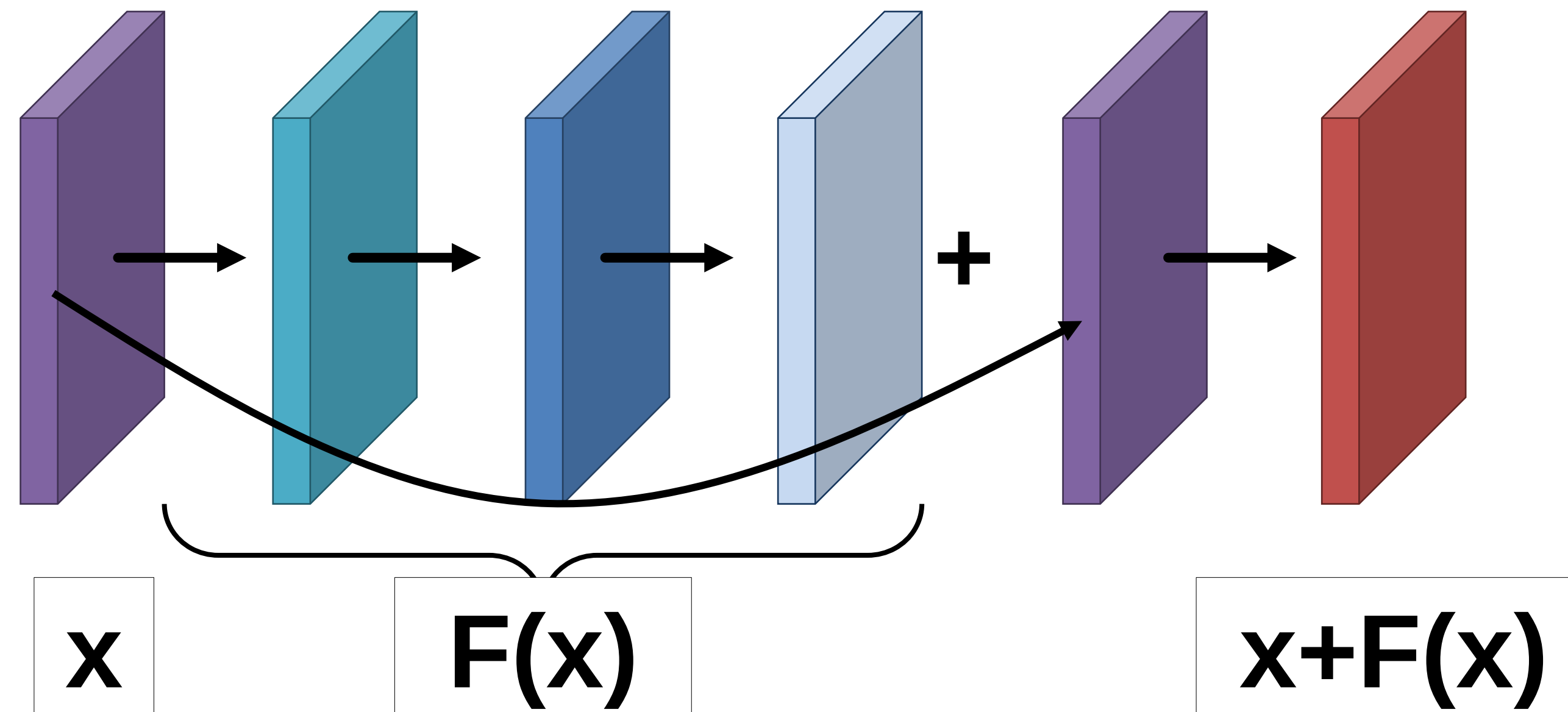
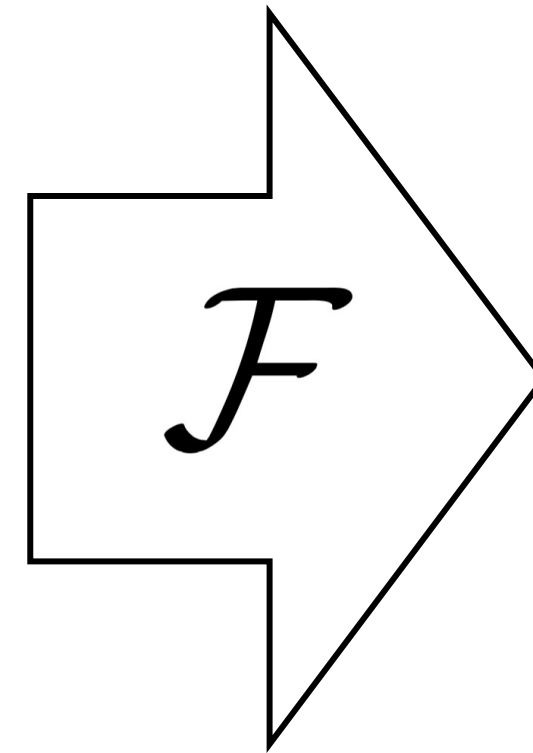
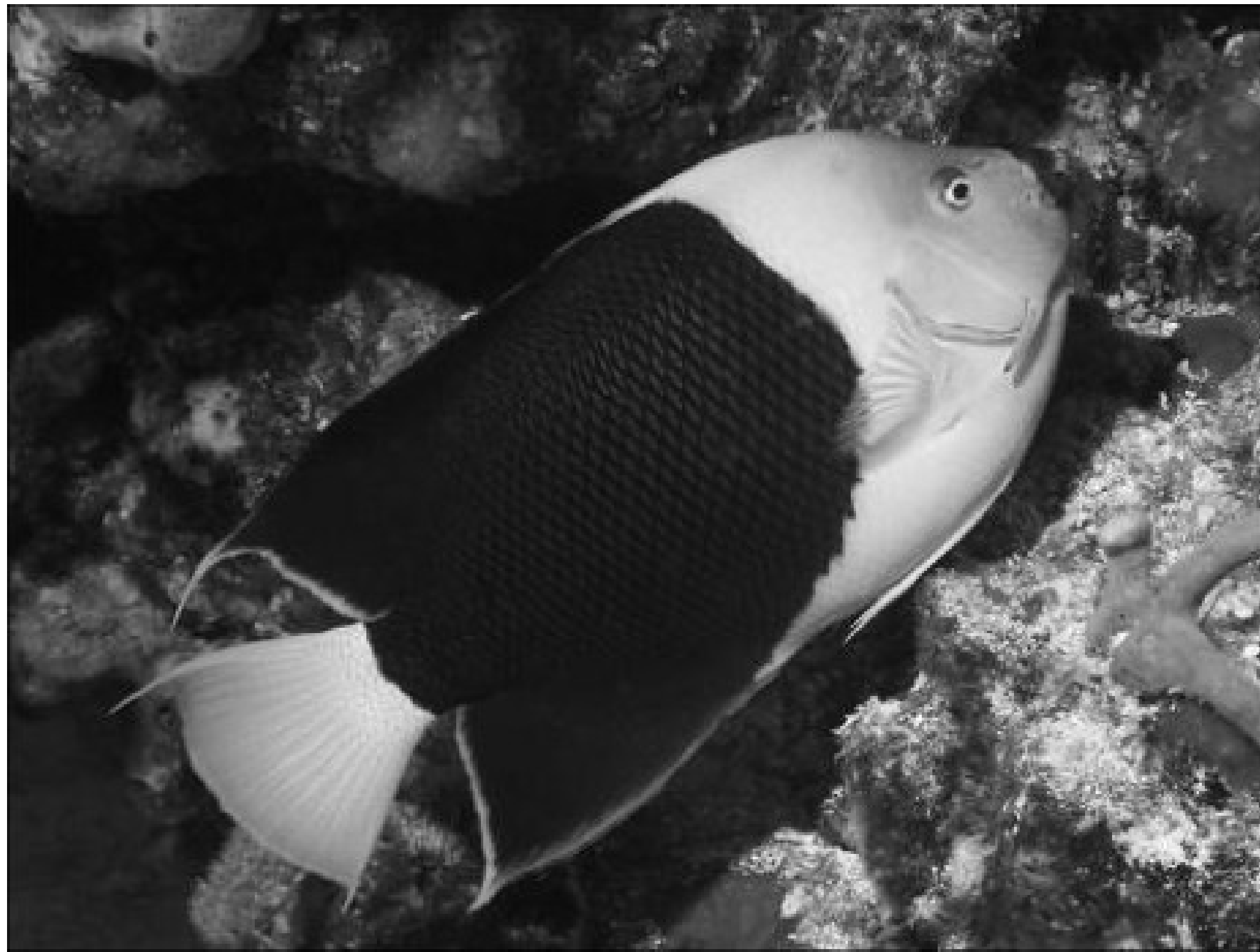
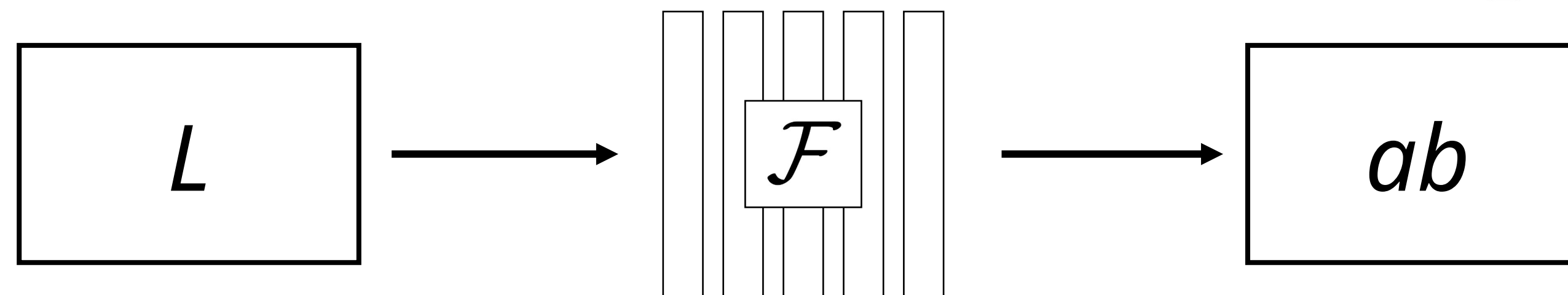


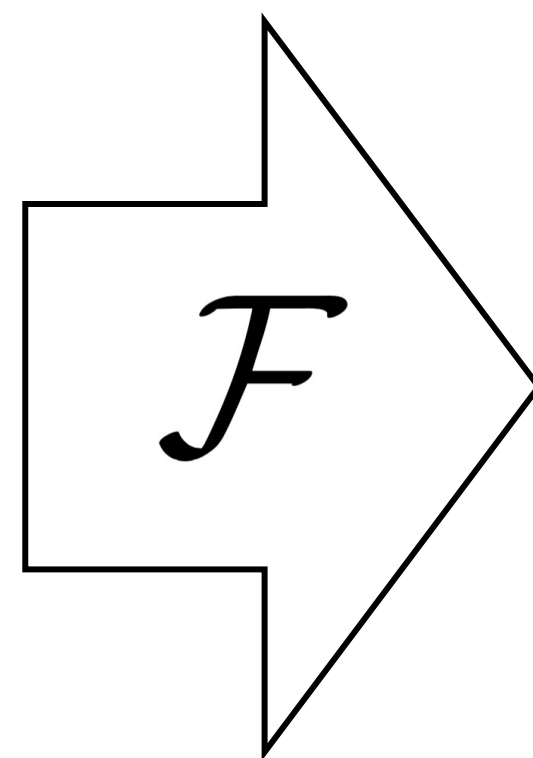
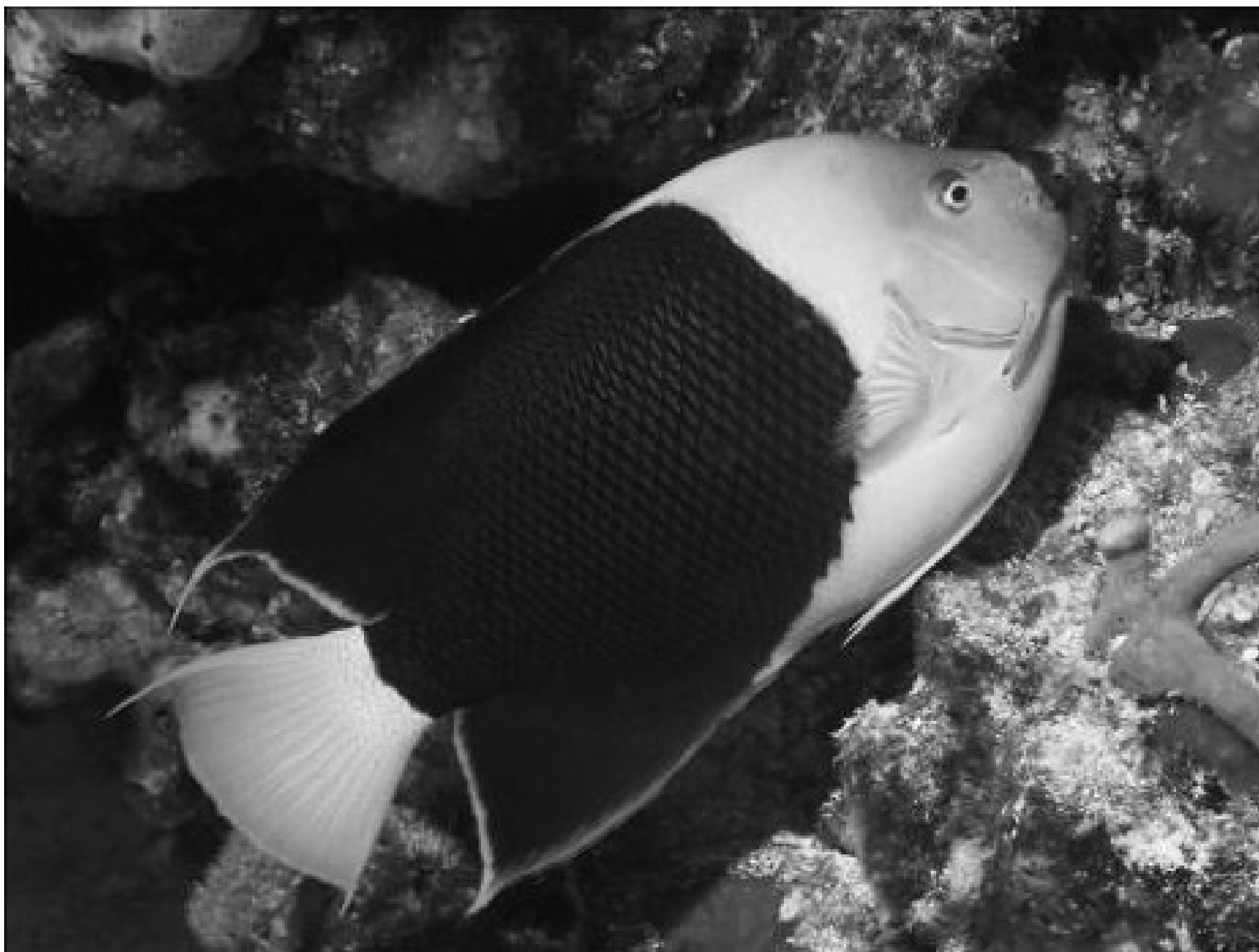
Image Colorization



Grayscale image: L
 chan $\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

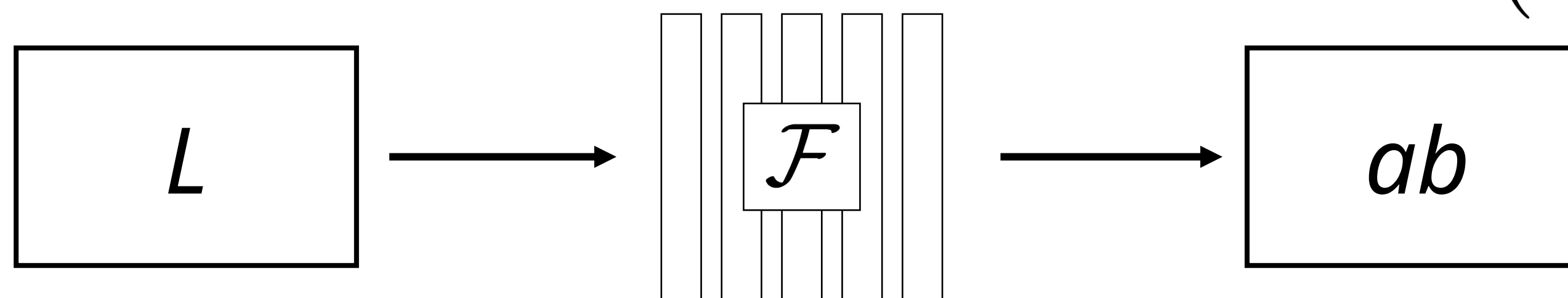
Color information: ab
 chan $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$





Grayscale image: L
channel $\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

Concatenate (L, ab)
channels $(\mathbf{X}, \hat{\mathbf{Y}})$



Regressing to pixel values doesn't work 😞

Input



Output
+



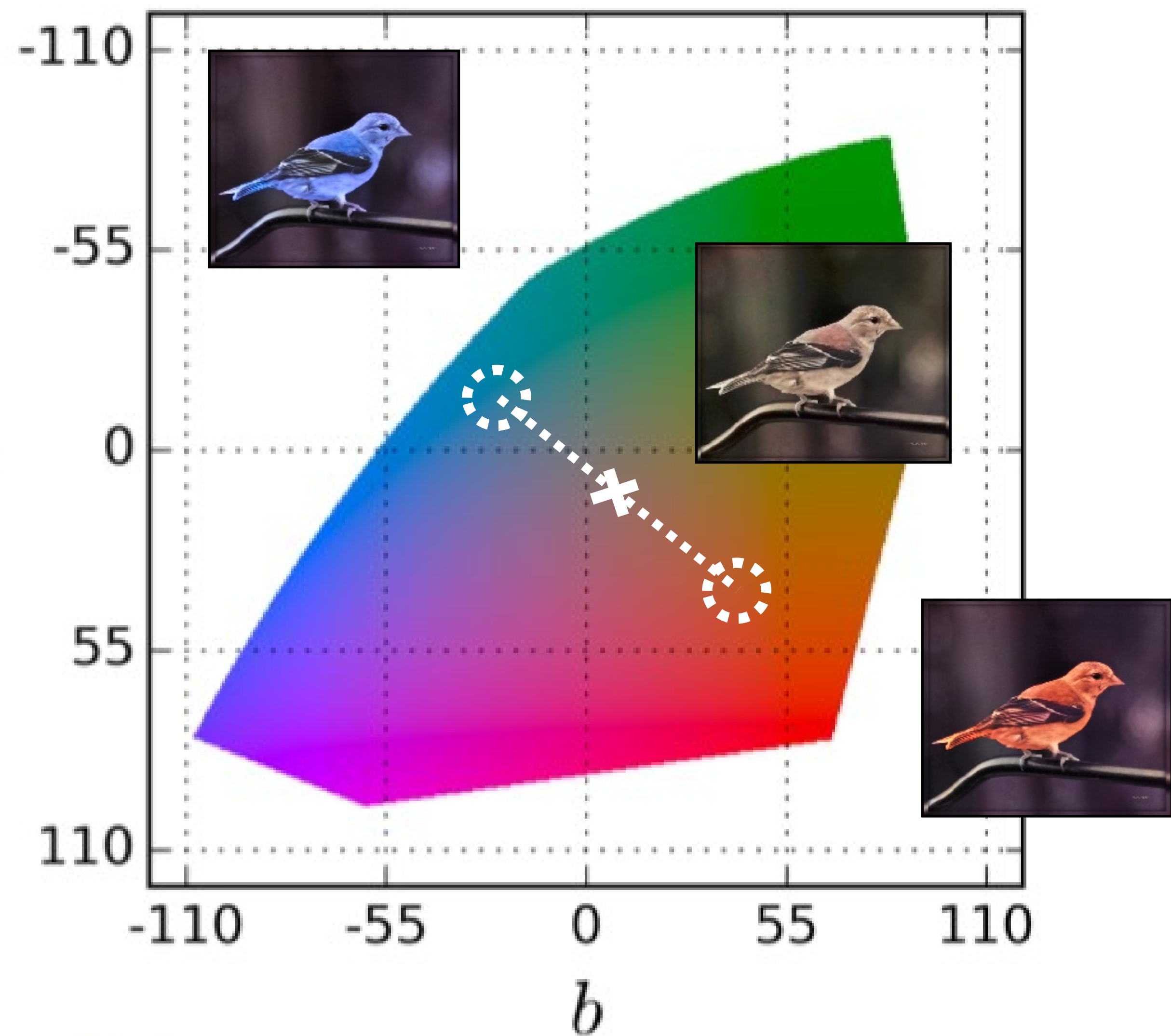
Ground
truth



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



a



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

Better Loss Function

Colors in *ab* space
(discrete)

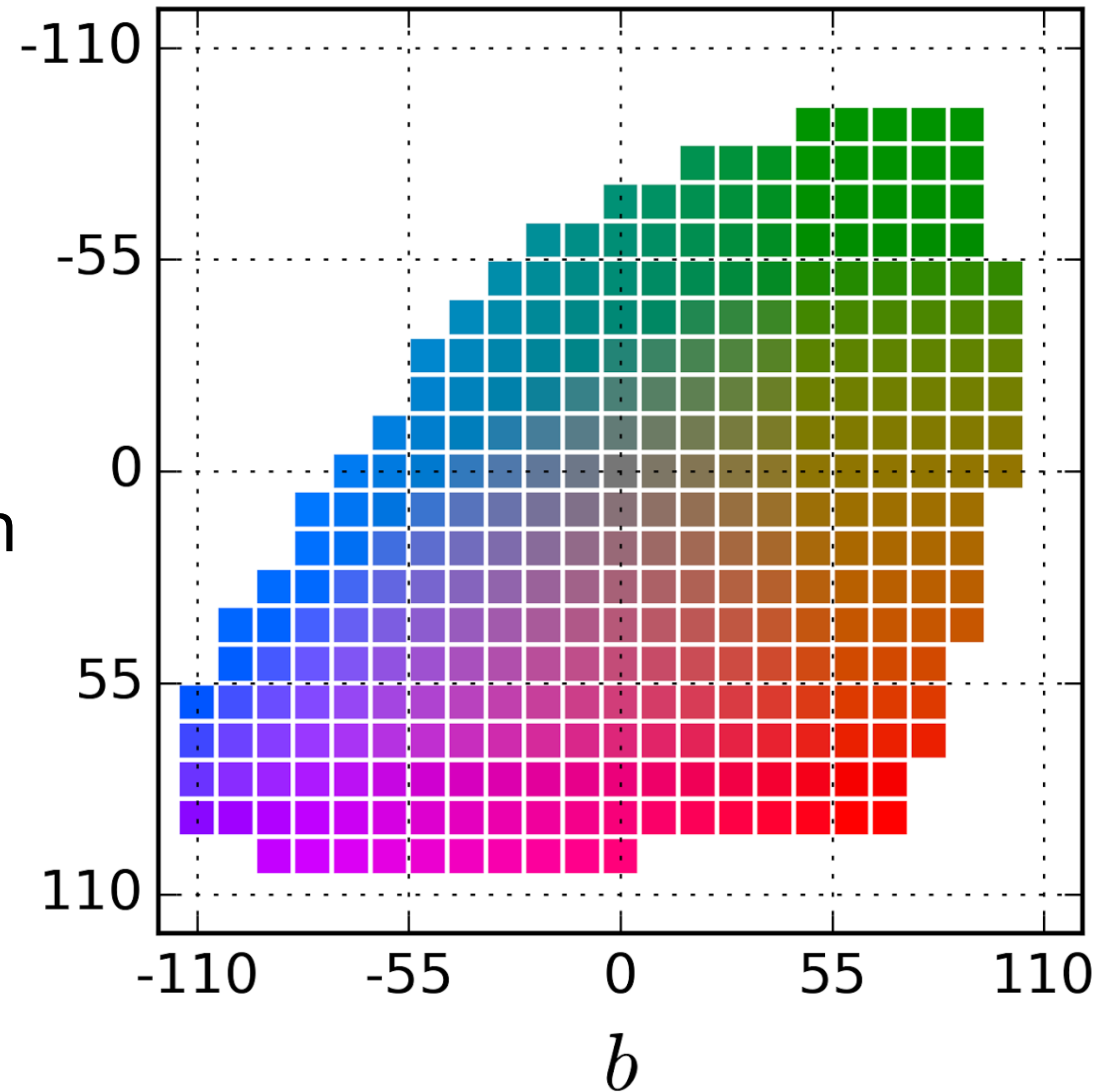
$$\theta^* = \arg \min_{\theta} \ell(\mathcal{F}_{\theta}(\mathbf{X}), \mathbf{Y})$$

- Regression with L2 loss inadequate

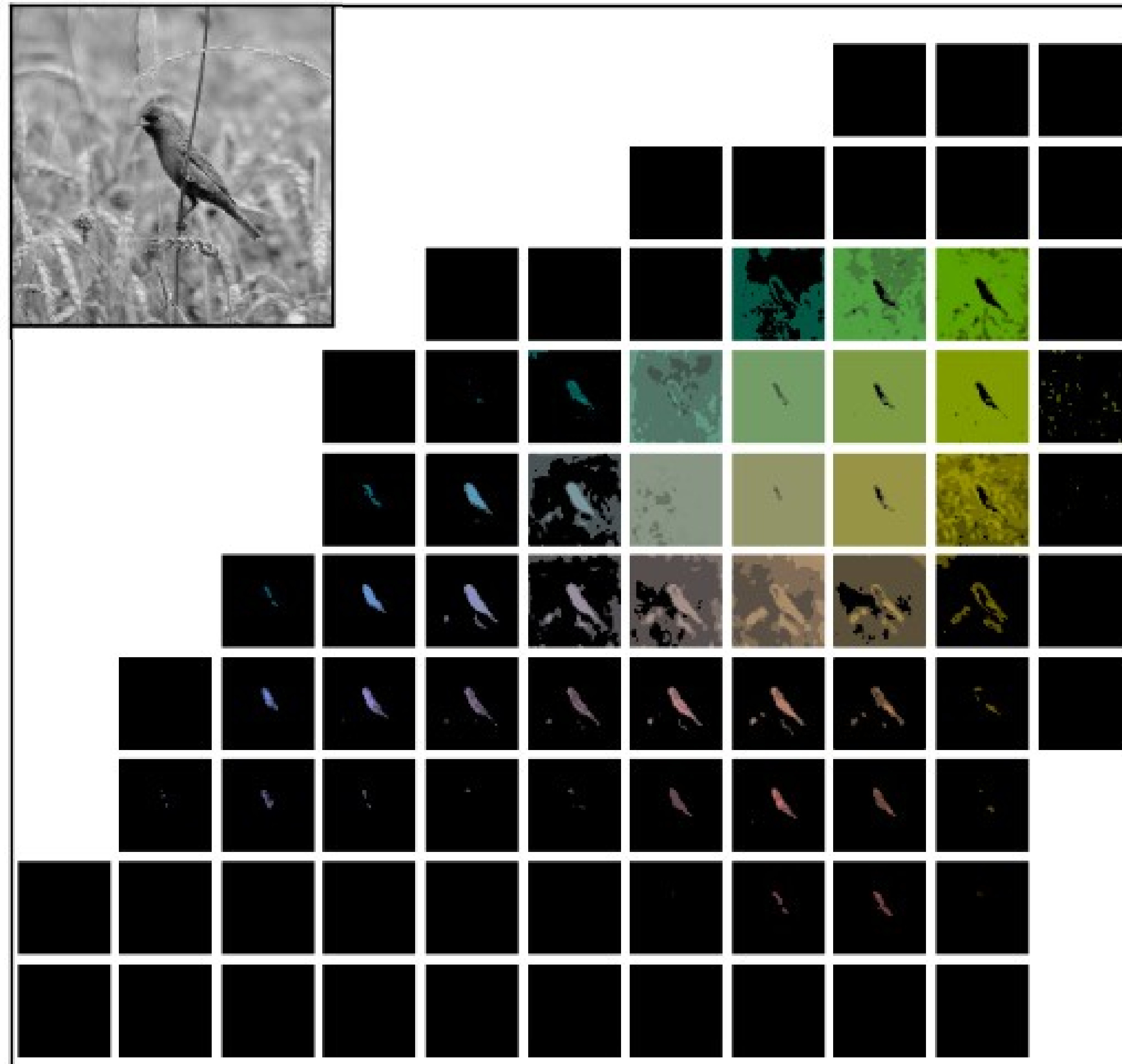
$$L_2(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \hat{\mathbf{Y}}_{h,w}\|_2^2$$

- Use per-pixel multinomial classification^a

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



a



b

Designing pixel loss functions

Input

Zhang et al.
2016

Ground
truth



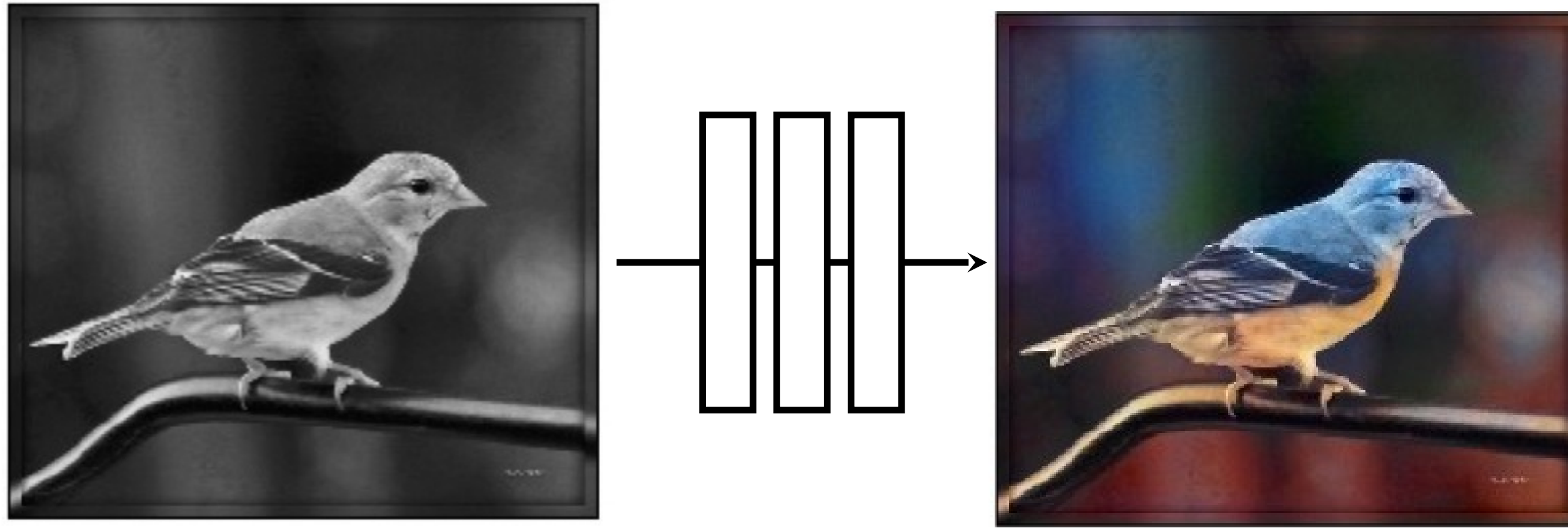
Color distribution cross-entropy loss with colorfulness enhancing term.

[Zhang, Isola, Efros, ECCV
2016]



Designing pixel loss functions

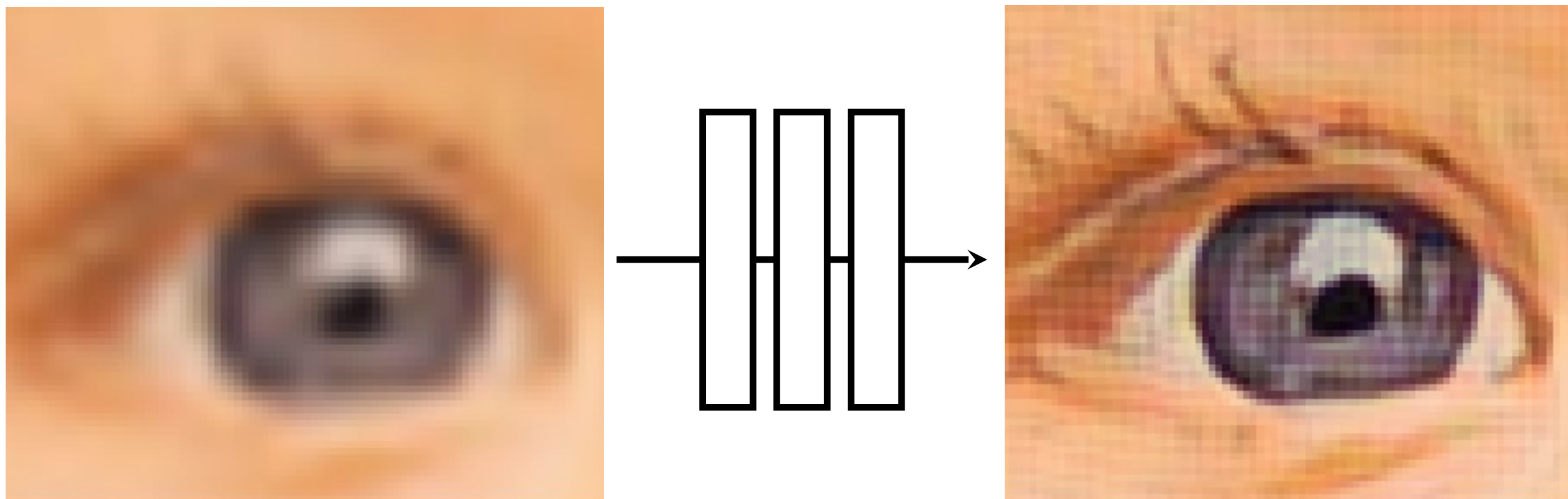
Image colorization



[Zhang et al. 2016]

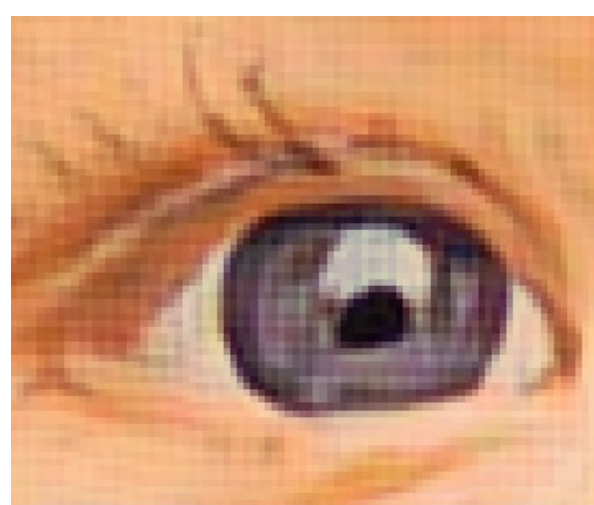
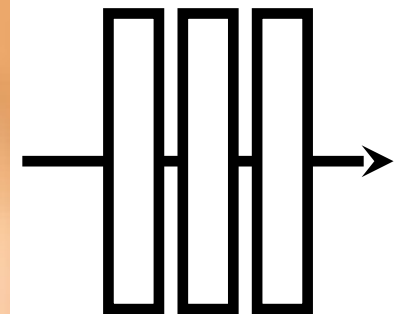
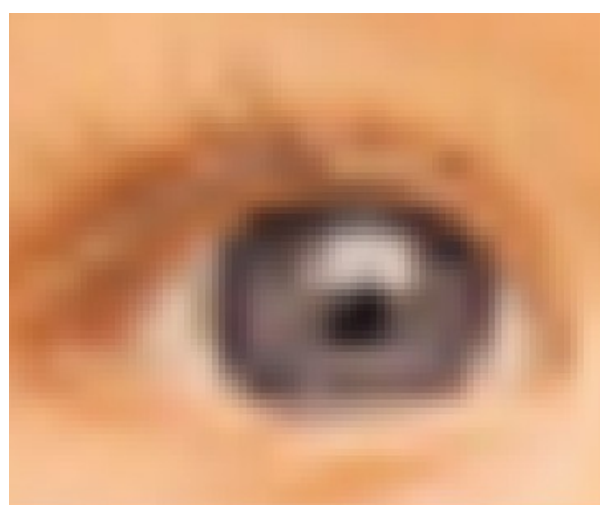
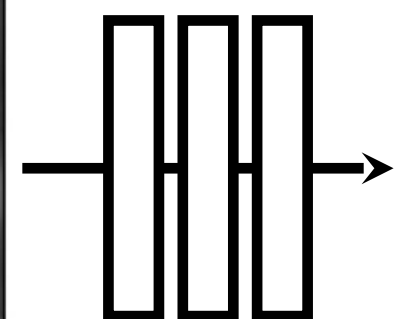
Cross entropy loss, with
colorfulness term

Super-resolution



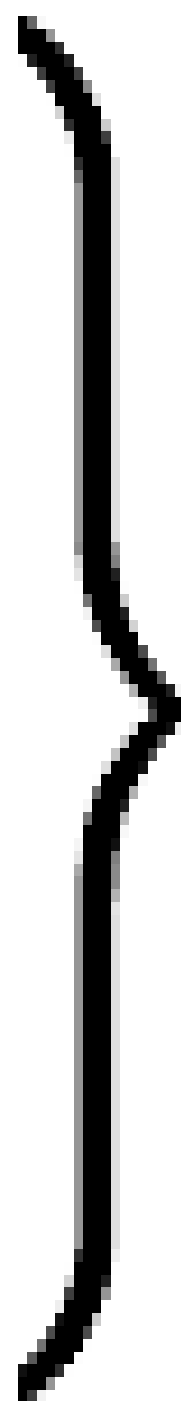
[Johnson et al. 2016]

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



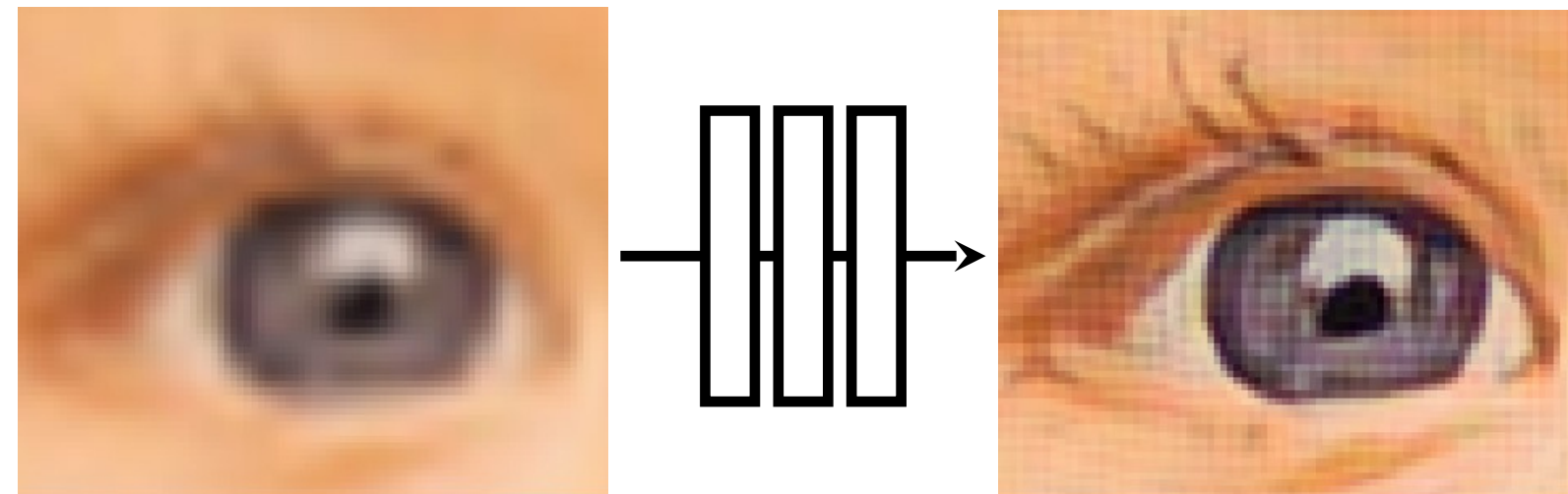
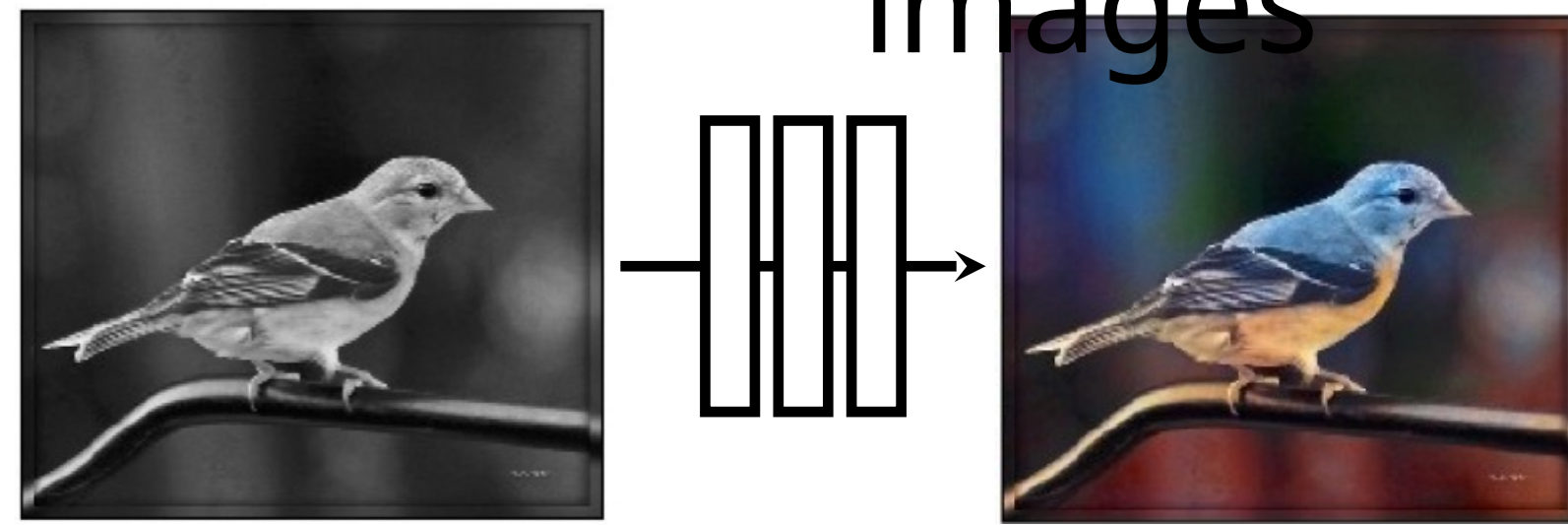
•
•
•

•
•
•



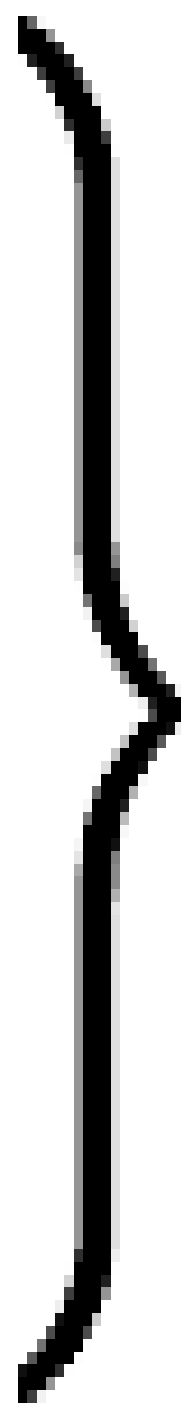
Universal
loss?

Generated
images

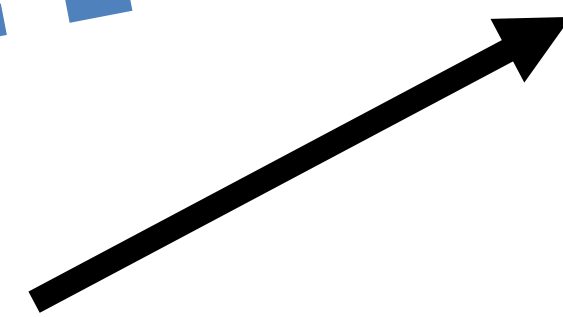
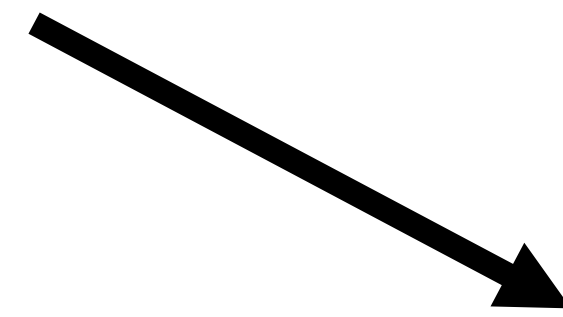


⋮

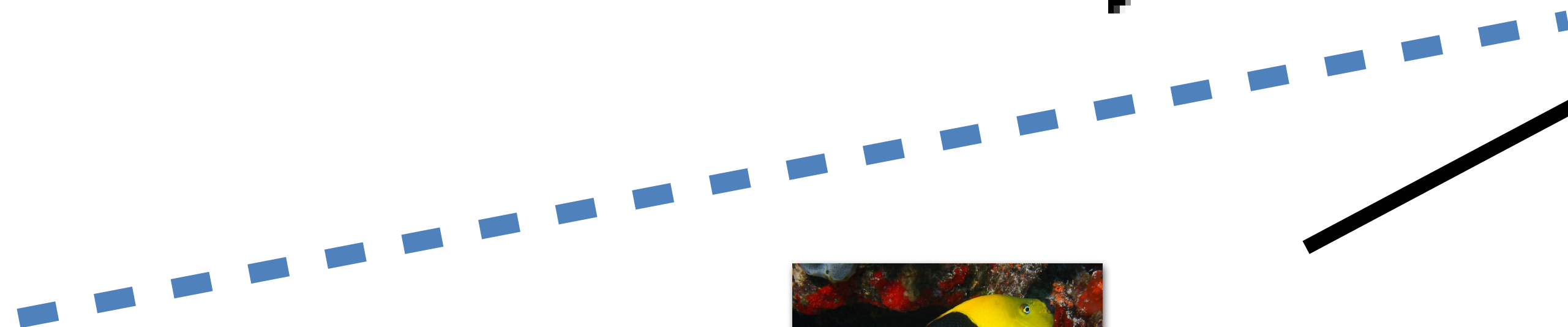
⋮



Generative Adversarial Network (GANs)



Generated
vs Real
(classifier)



Real
photos



...

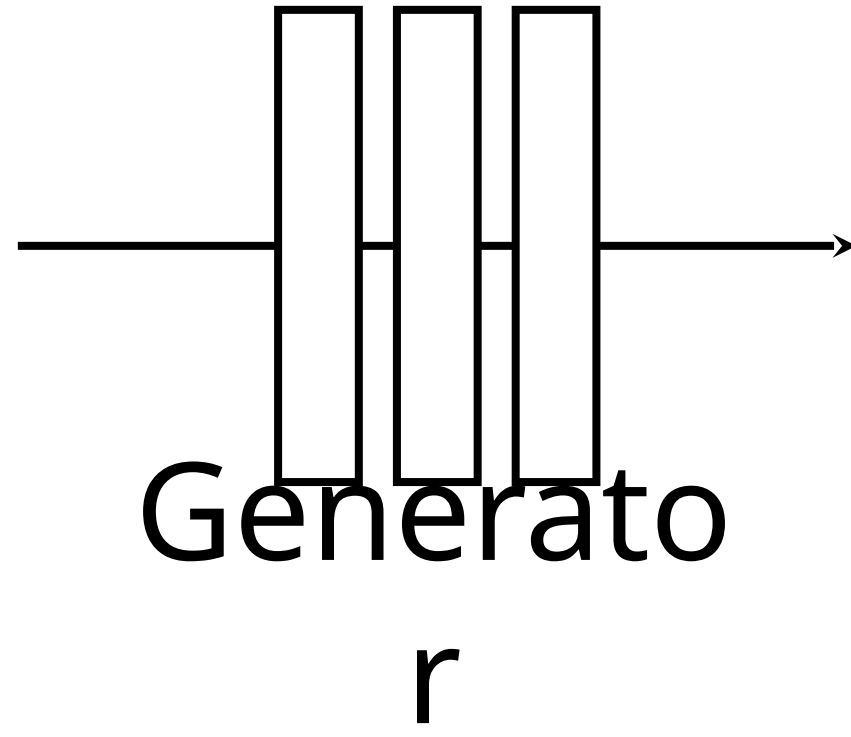


[Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, Bengio 2014]

$\mathbf{x}$



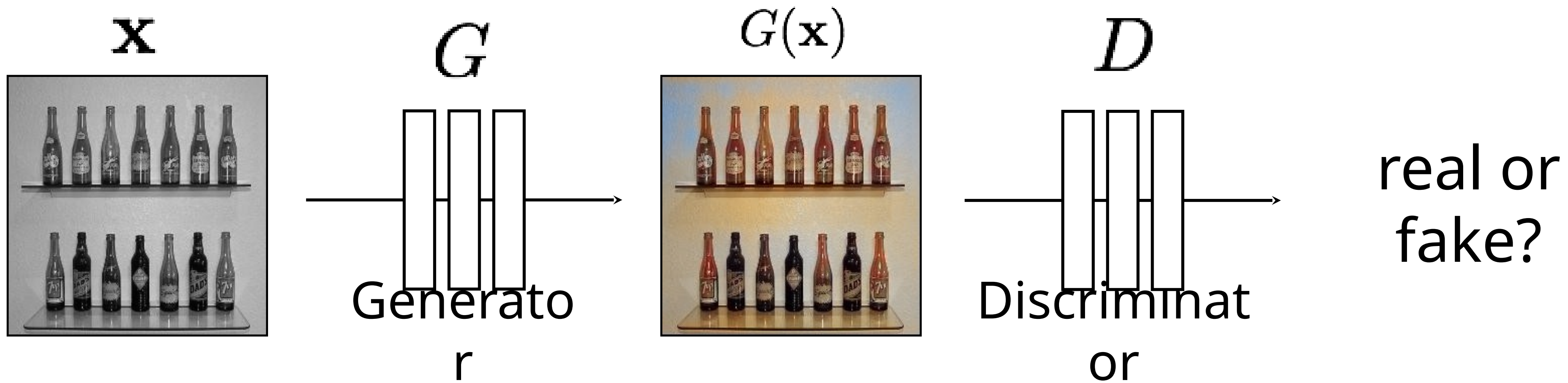
G



$G(\mathbf{x})$

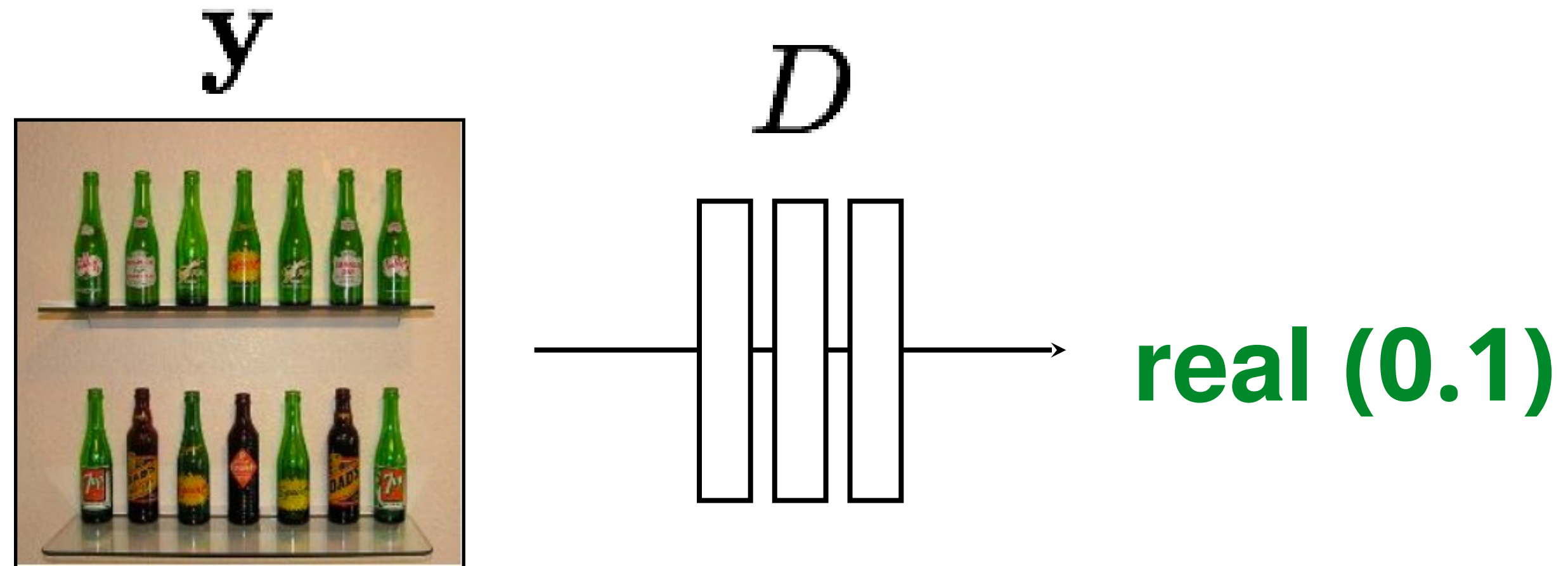


[Goodfellow et al.,
2014]



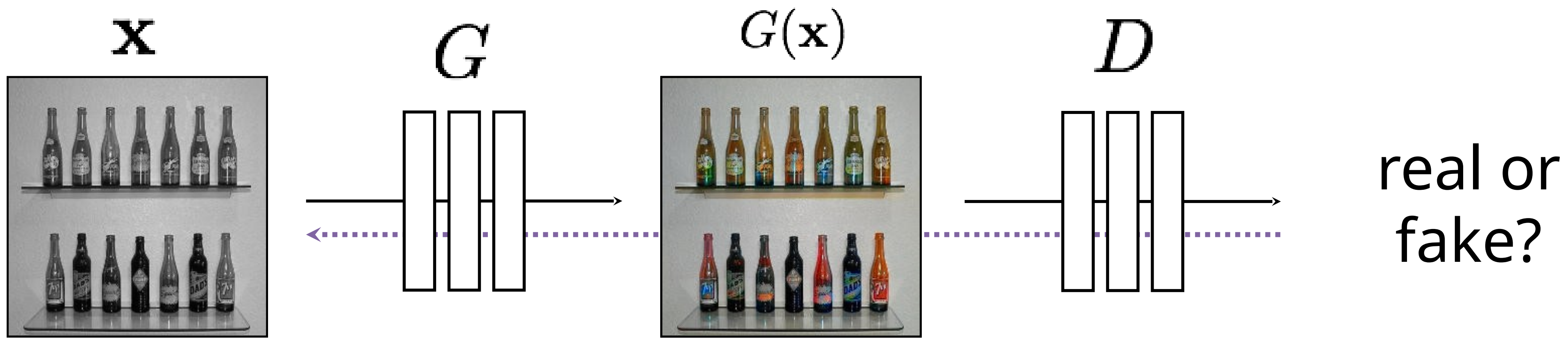
G tries to synthesize fake images that fool D

D tries to identify the fakes



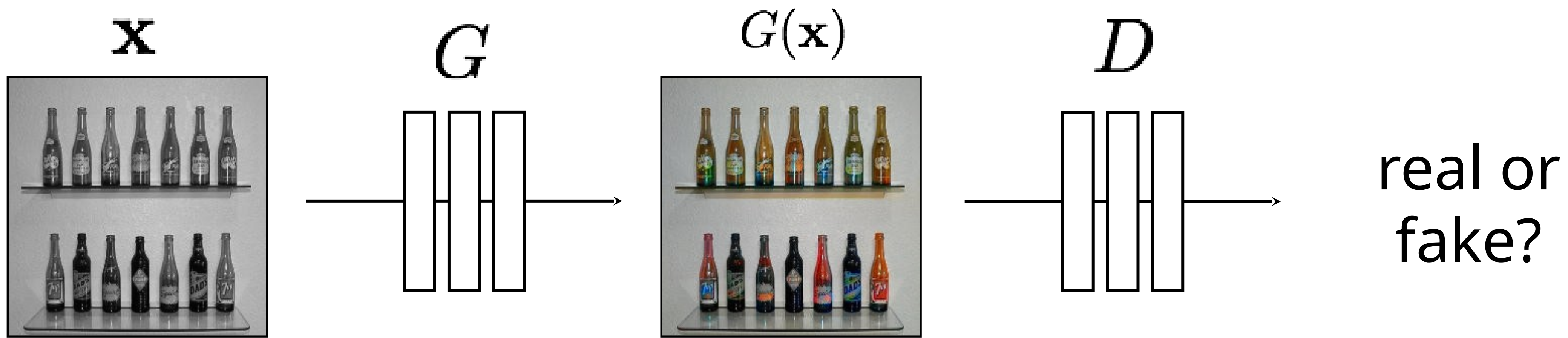
$$\arg \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$

[Goodfellow et al.,
2014]



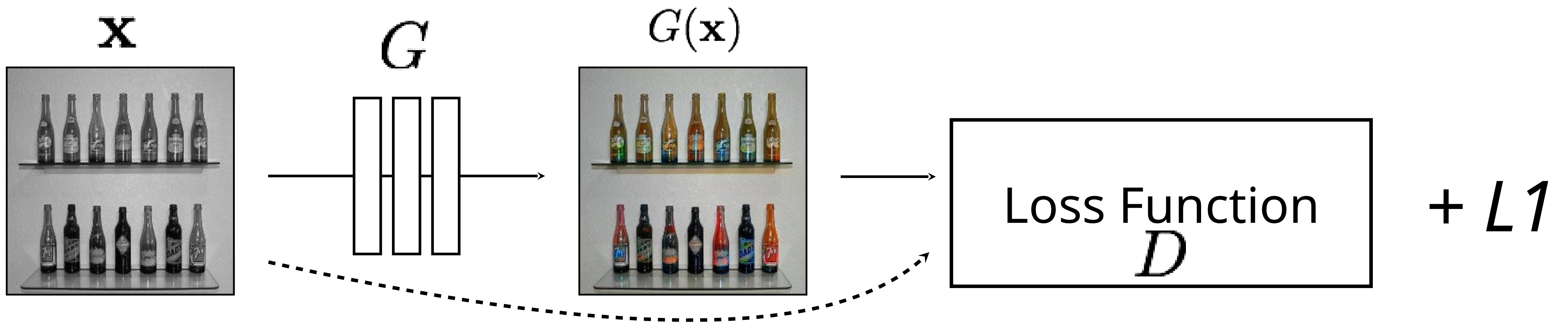
**G tries to synthesize fake images that
fool D:**

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



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

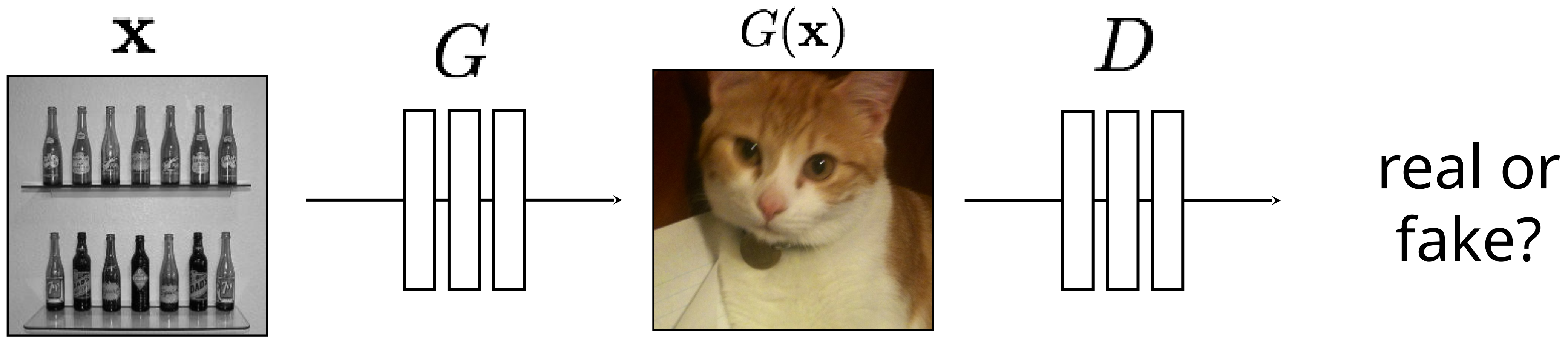
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$



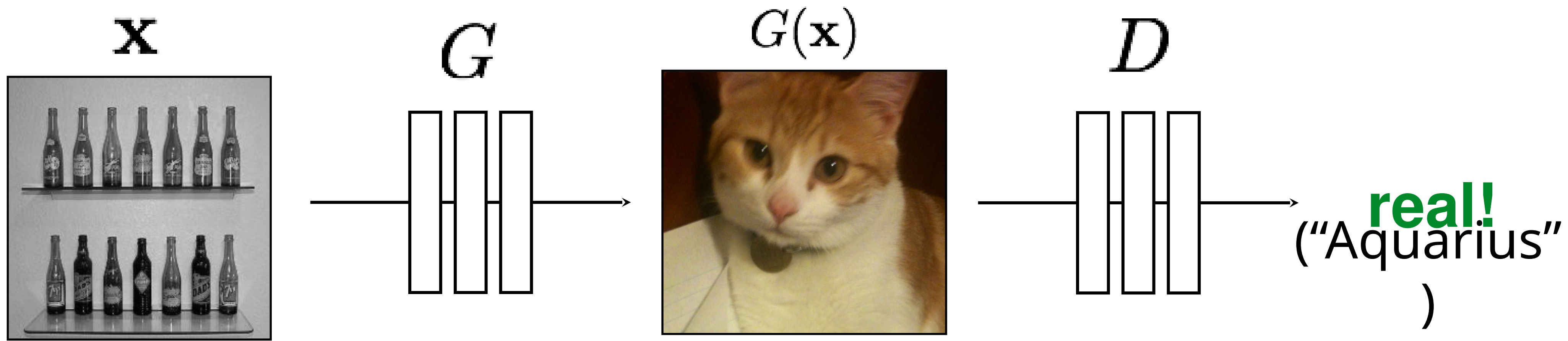
G's perspective: D is a loss function.

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

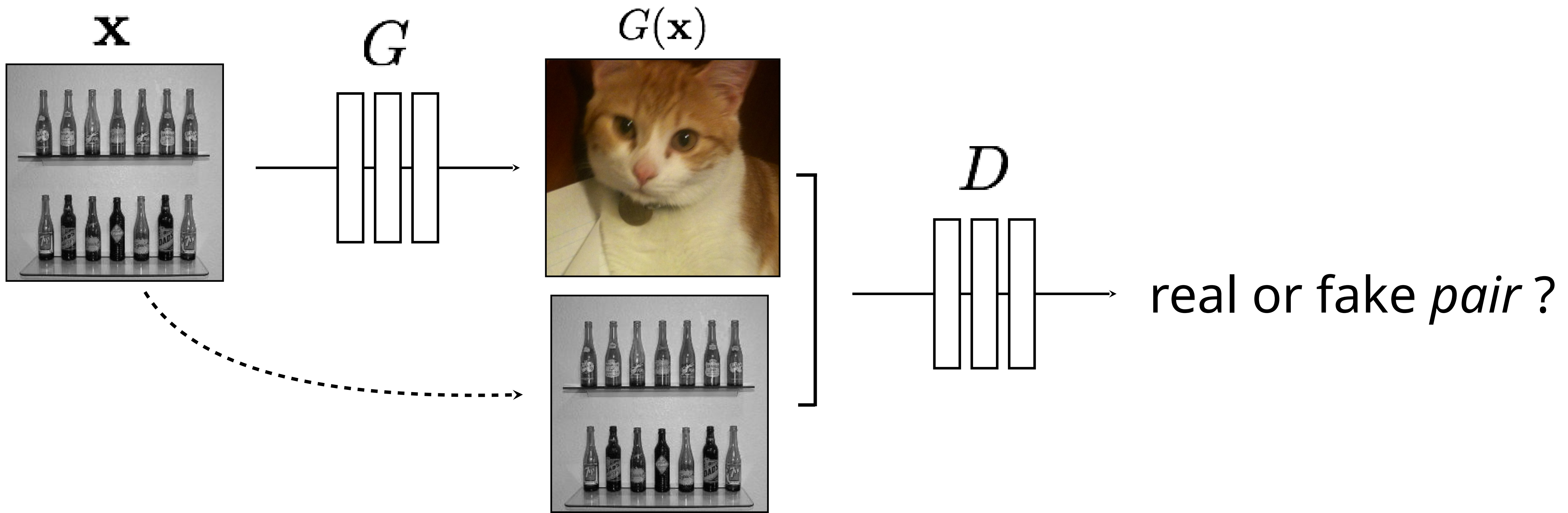
[Goodfellow et al.,
2014]
[Isola et al.,
2017]



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

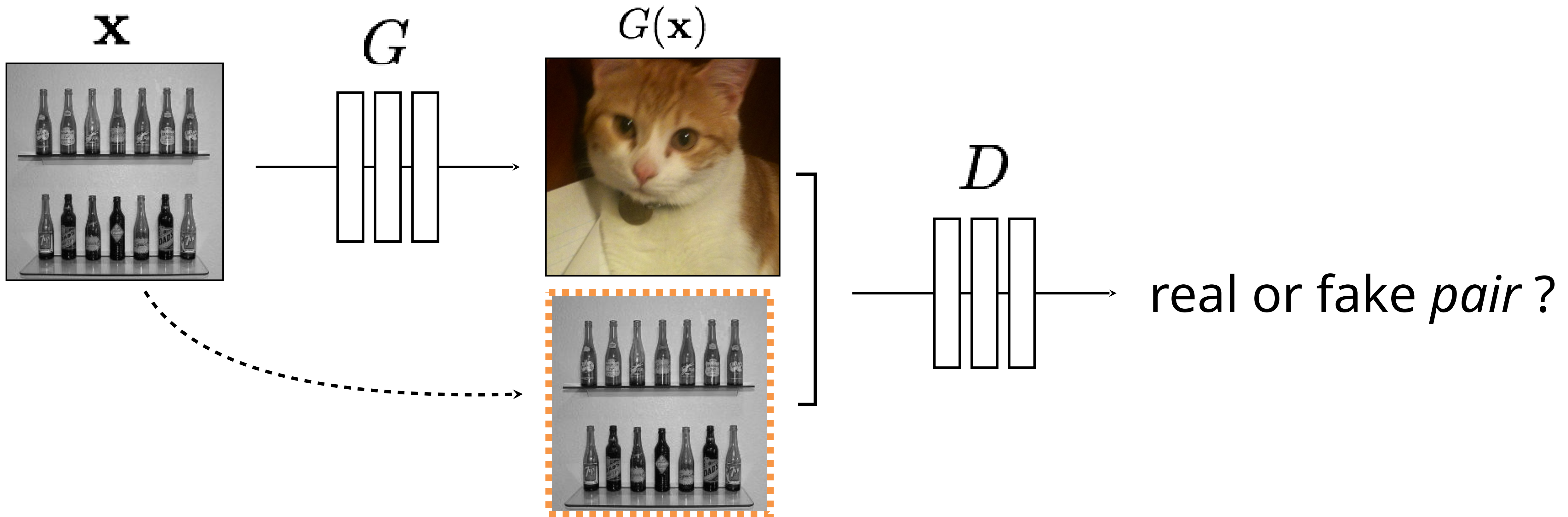


$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$

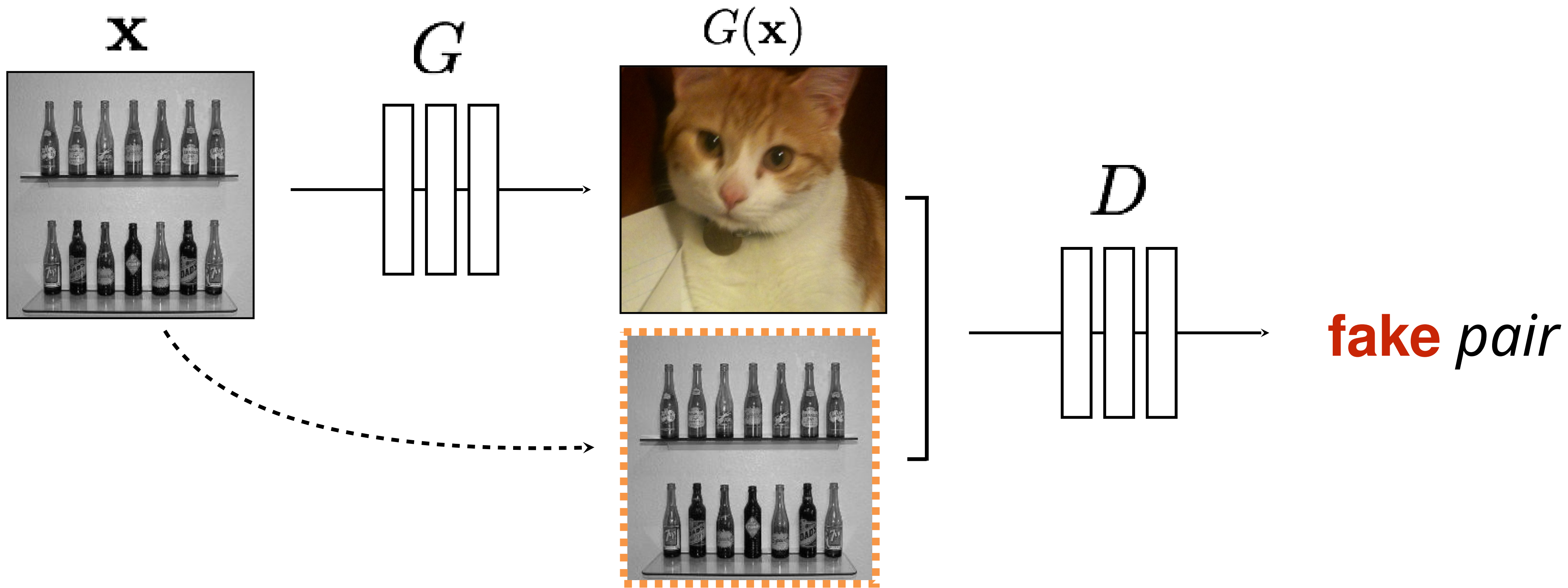


$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$

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

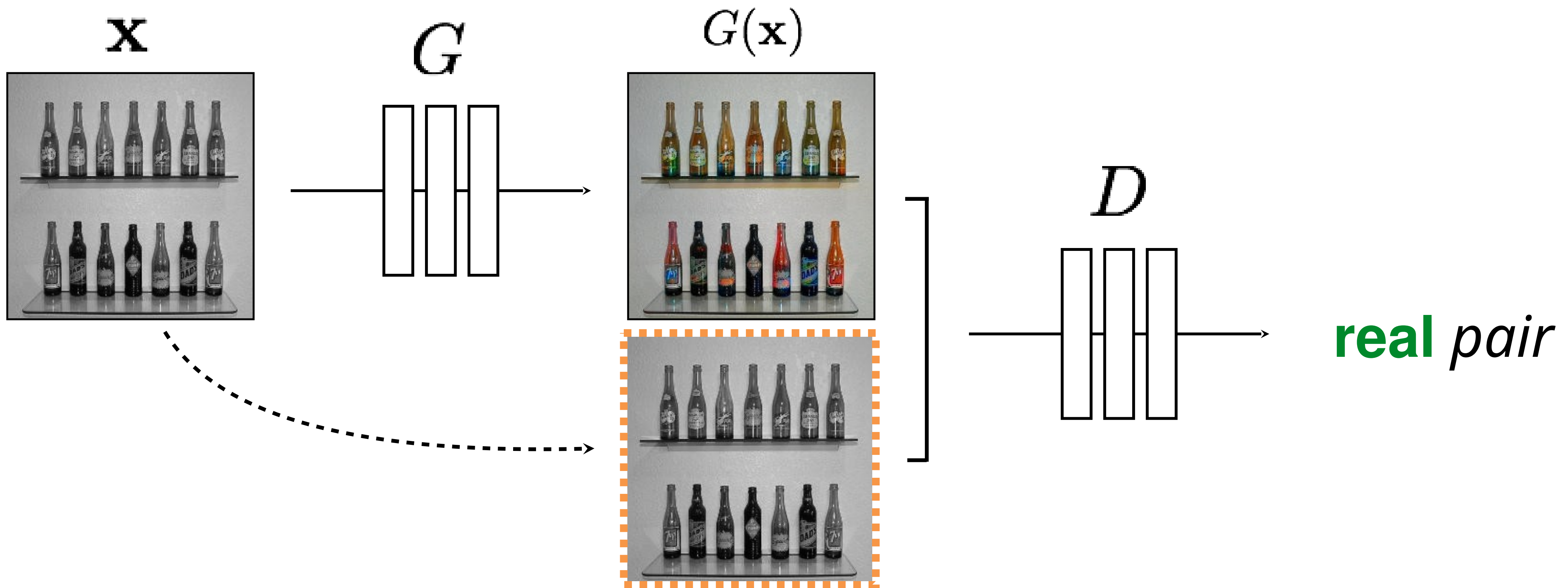


[Goodfellow et al.,
2014]
[Goodfellow et al.,
2017]



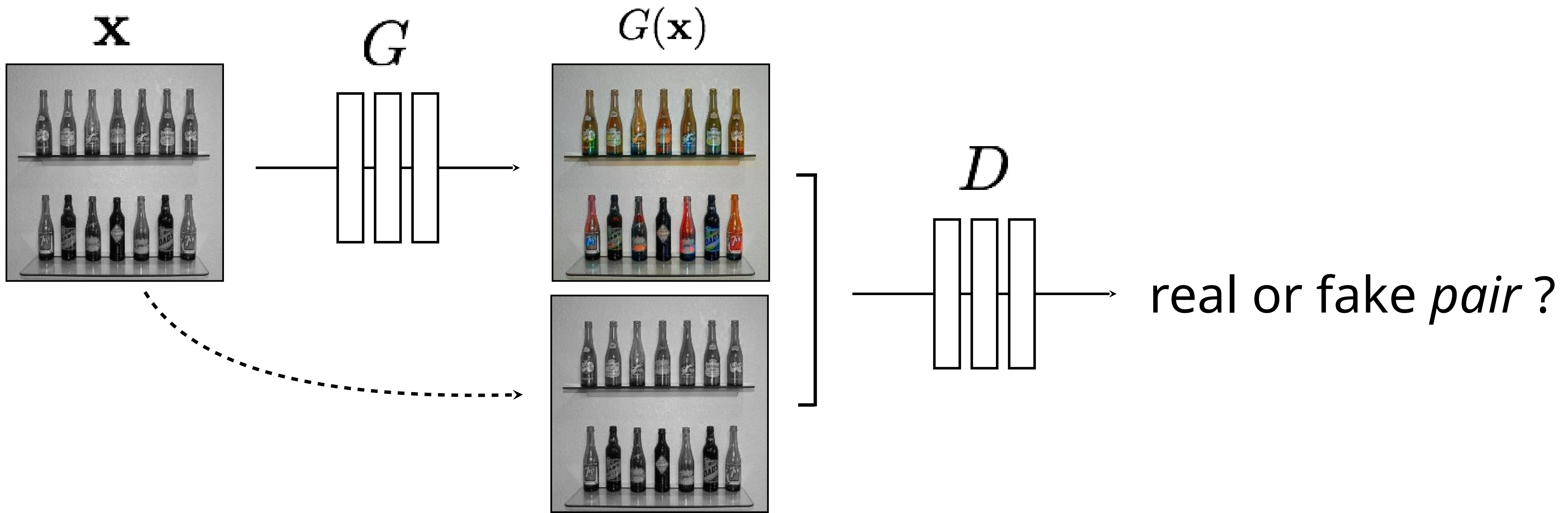
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))]$$

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



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

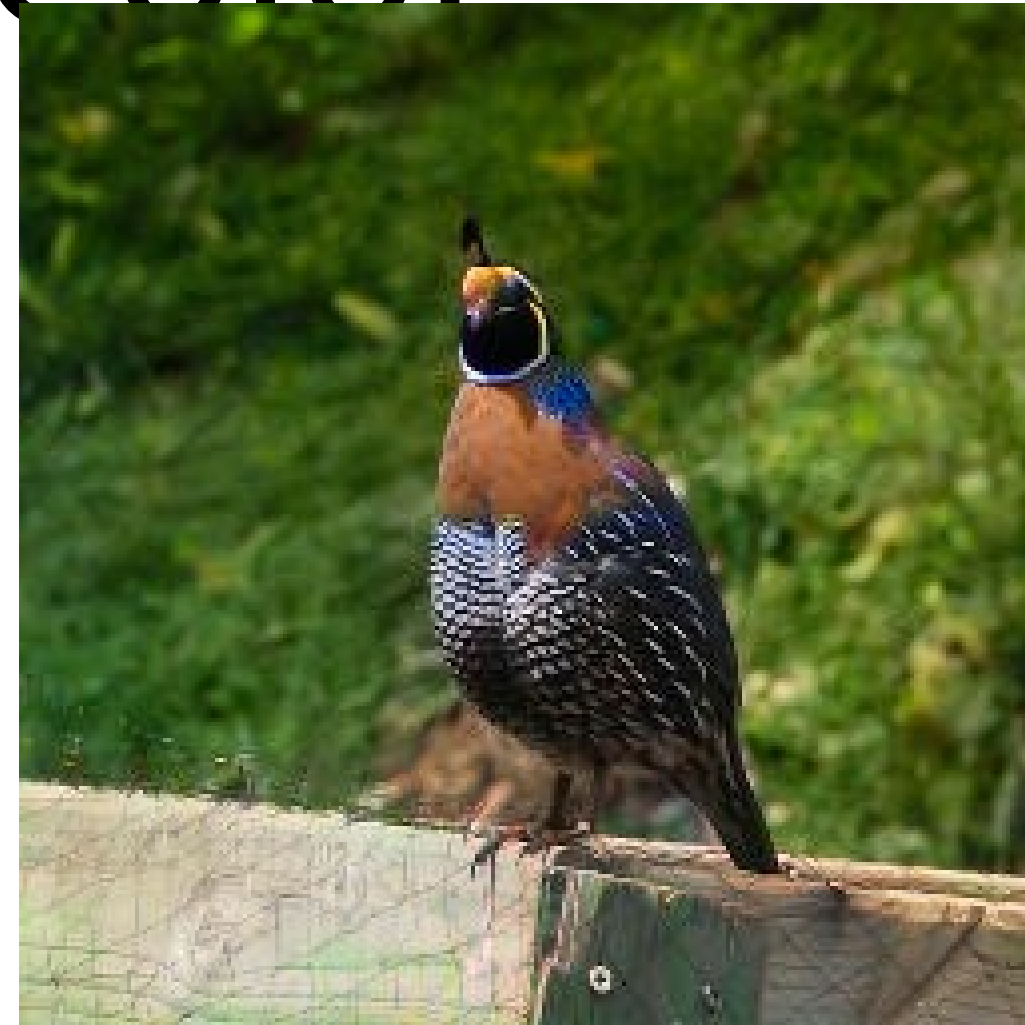
[Goodfellow et al.,
2014]
[Goodfellow et al.,
2017]

BW →
Color



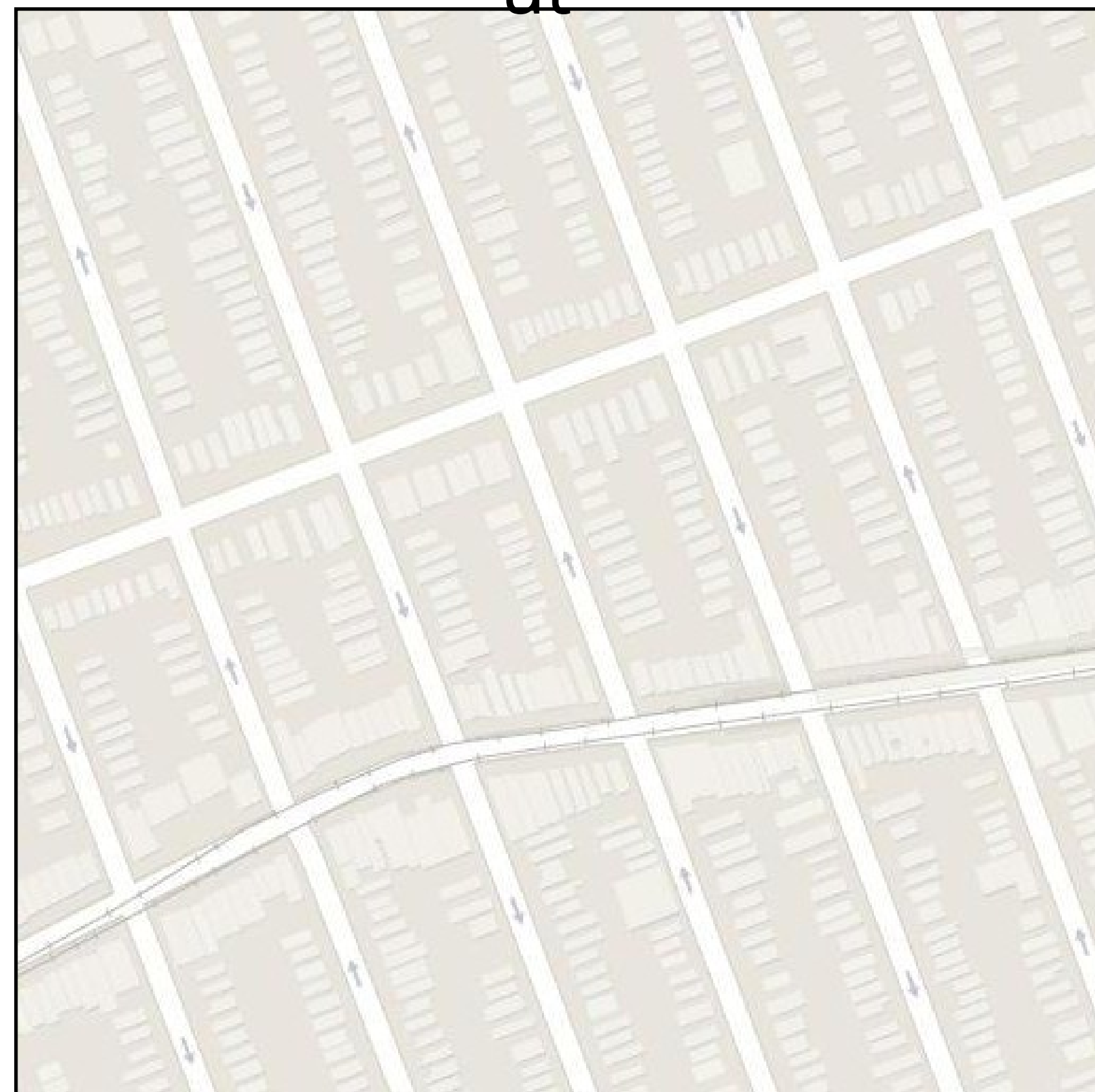
Data from [Russakovsky et al.
2015]

BW →
Color

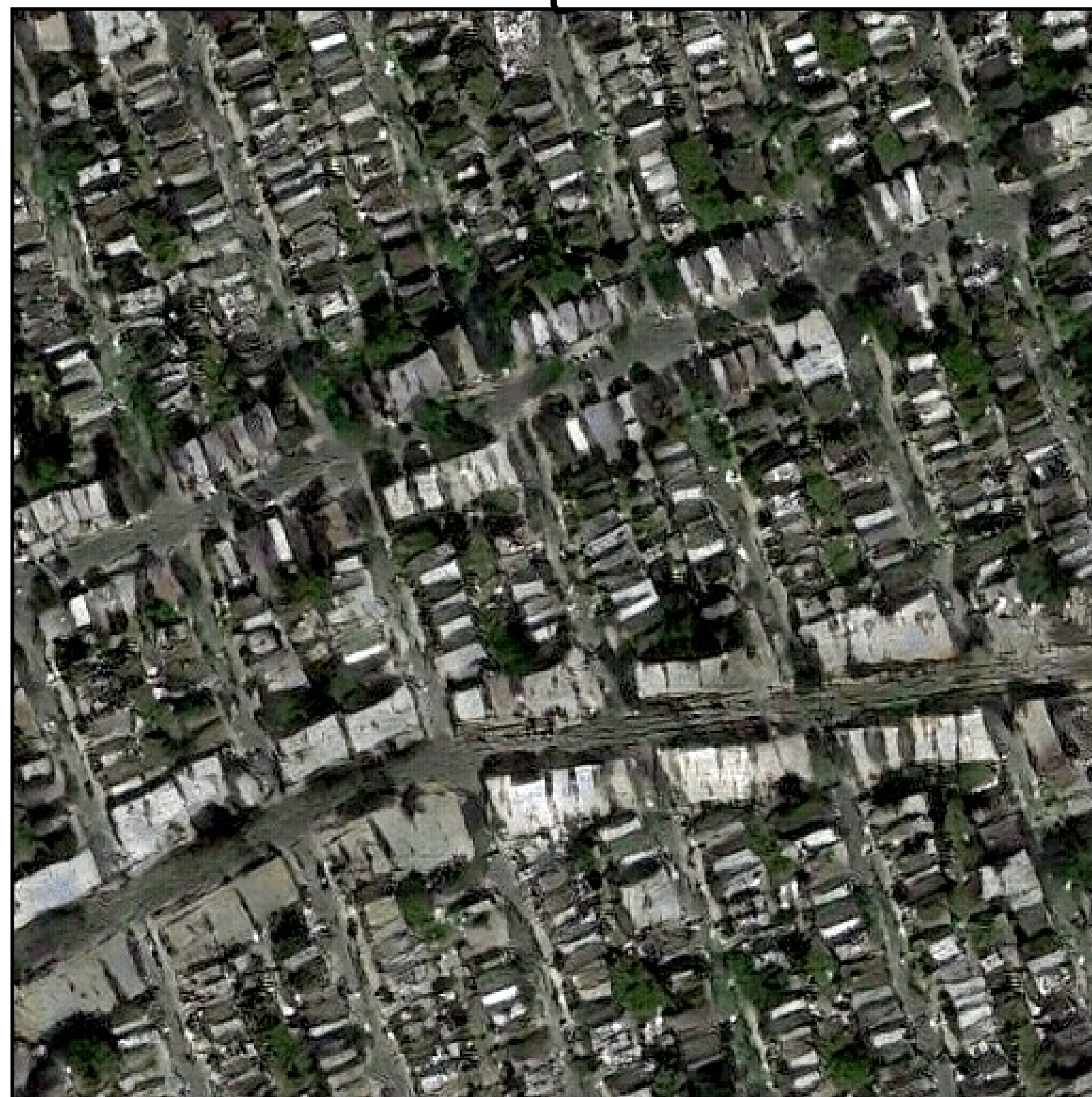


Data from [Russakovsky et al.
2015]

Input



Output



Groundtruth



Data from
[\[maps.google.com\]](https://maps.google.com)



Input

Output

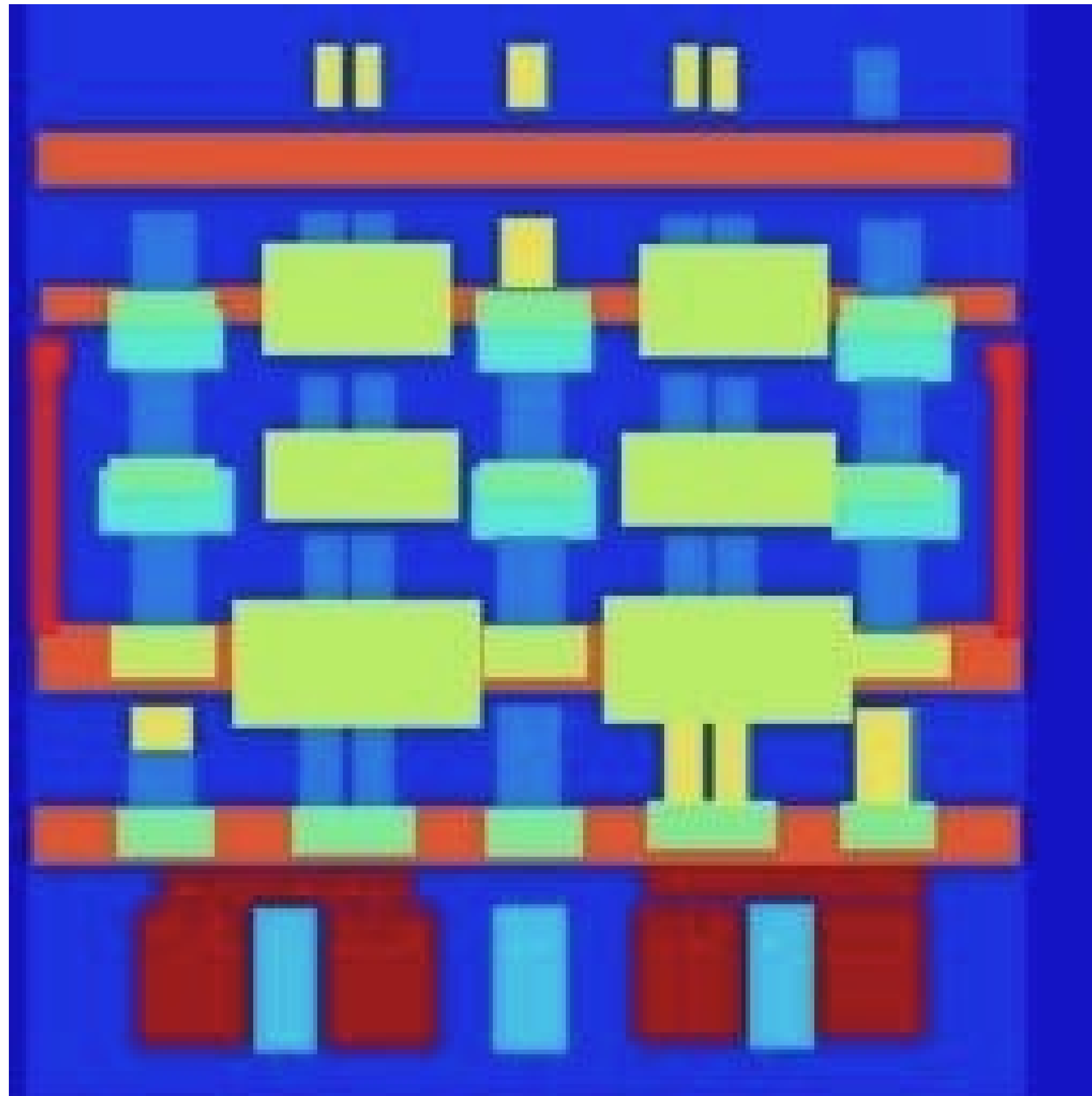
Groundtruth



Data from [[maps.google.com](https://www.google.com/maps)]

Labels → Facades

Input



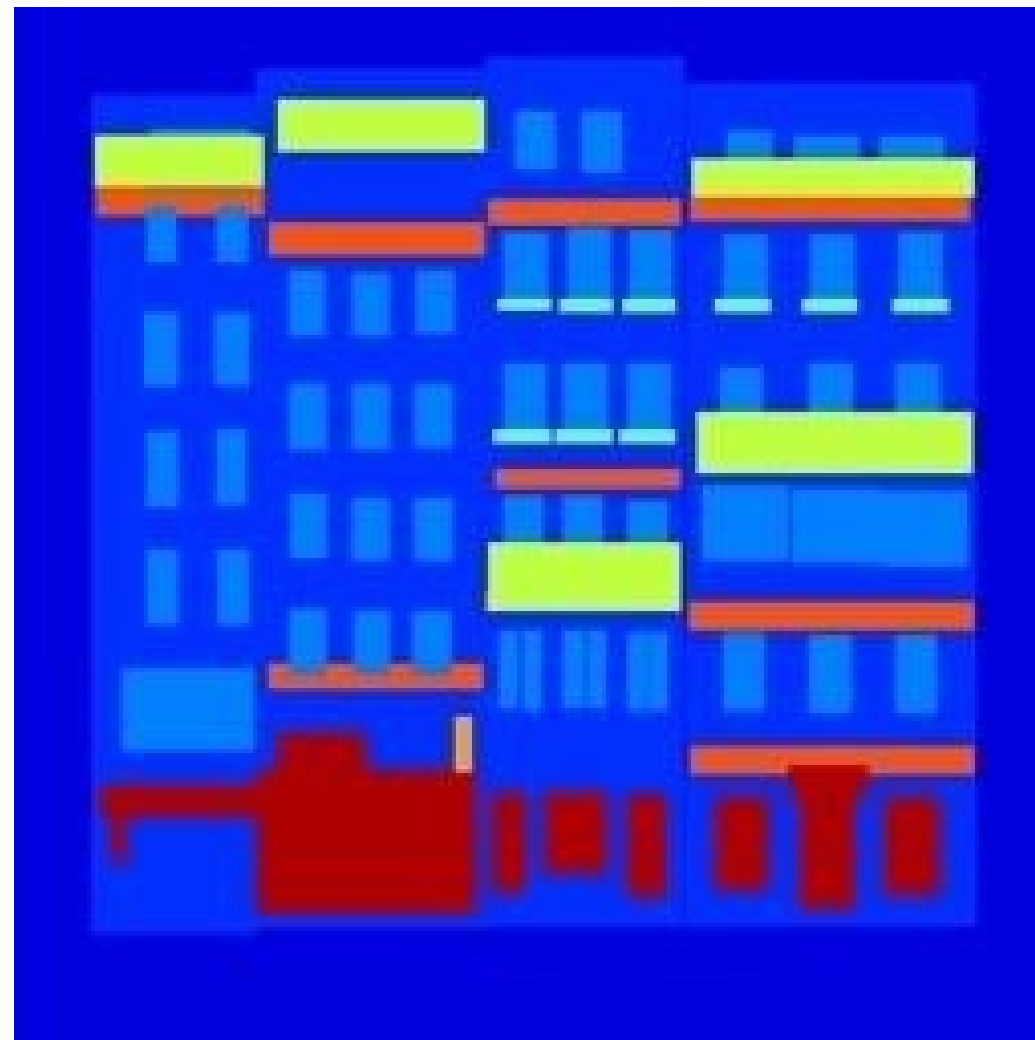
Output



Data from [Tylecek,
2013]

Labels → Facades

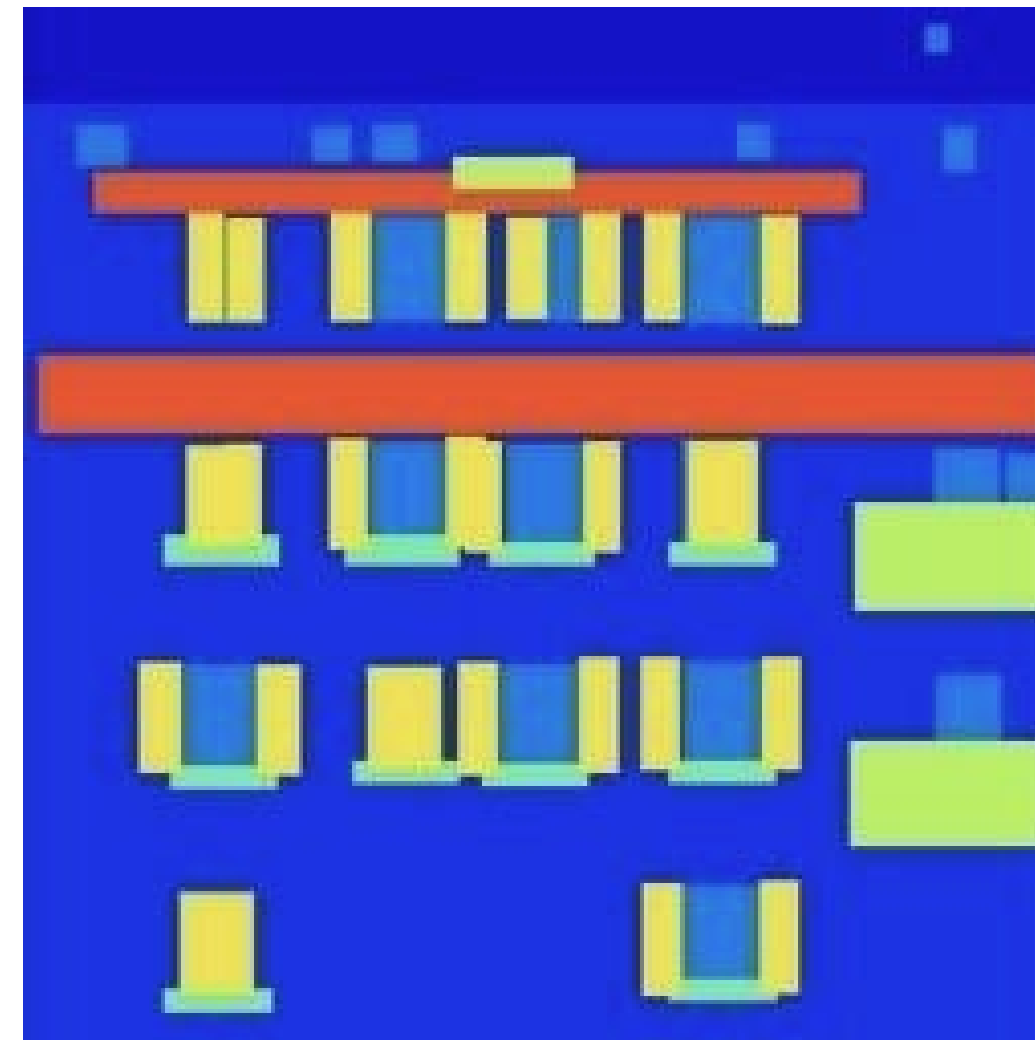
Input



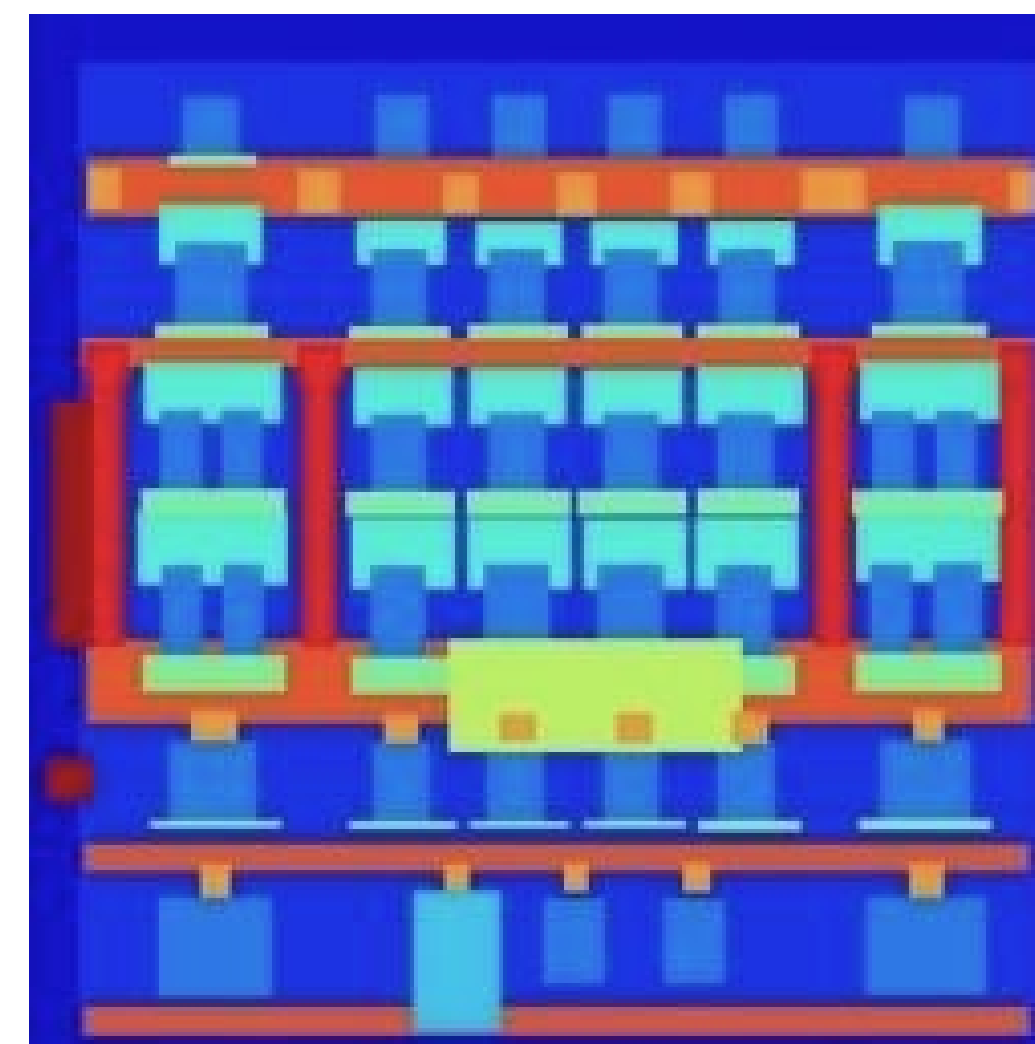
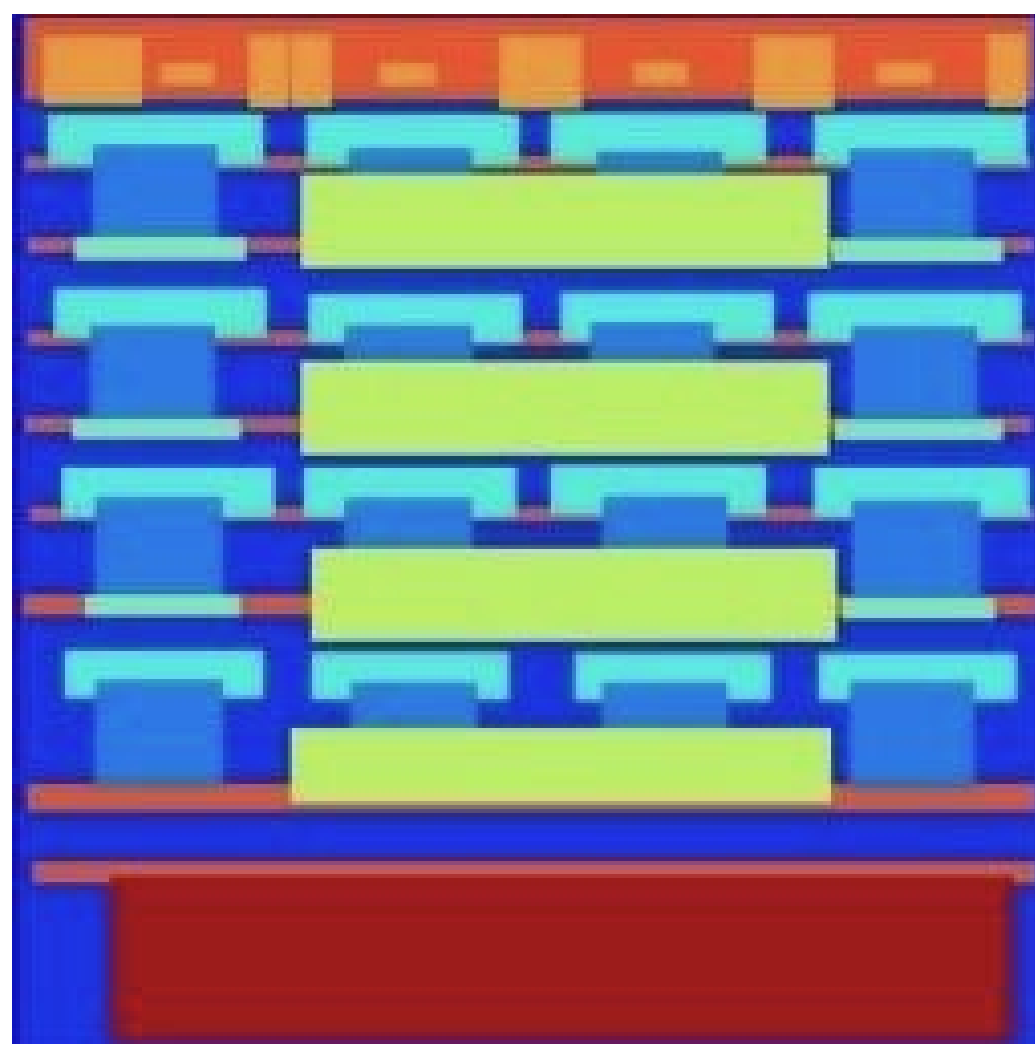
Output



Input



Output



Data from [Tylecek,
2013]

Day → Night

Input

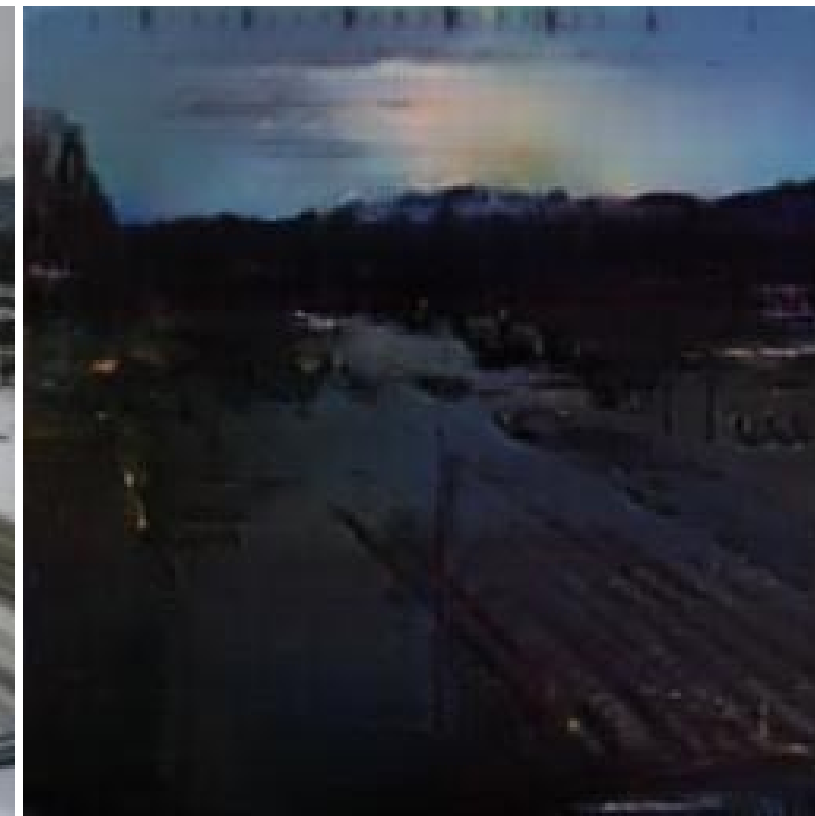
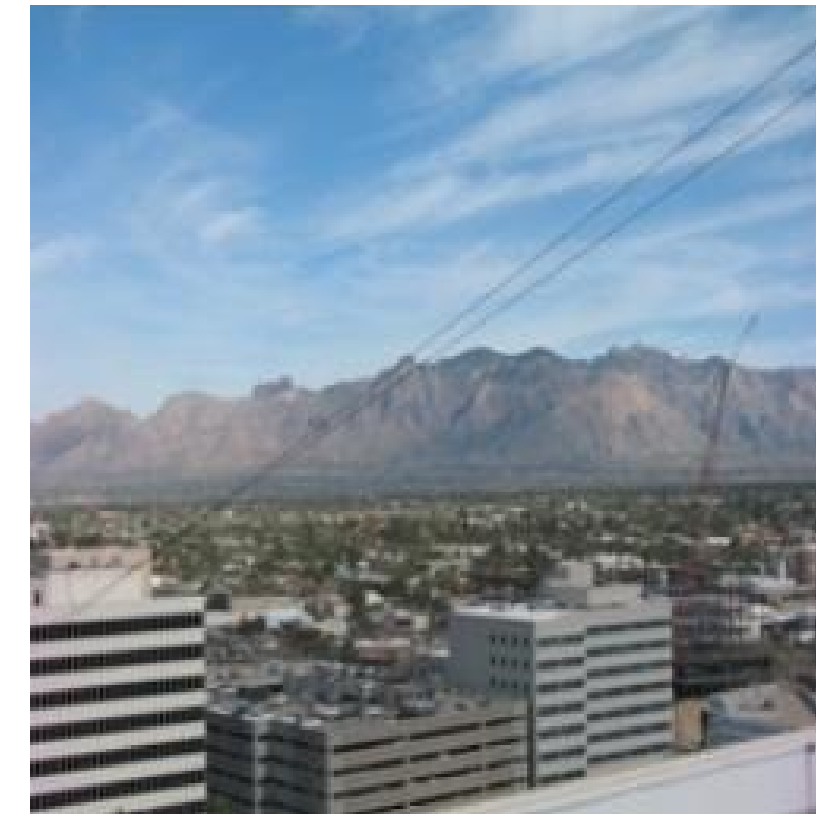
Output

Input

Output

Input

Output



Data from [Laffont et al.,
2014]

Thermal →

Input

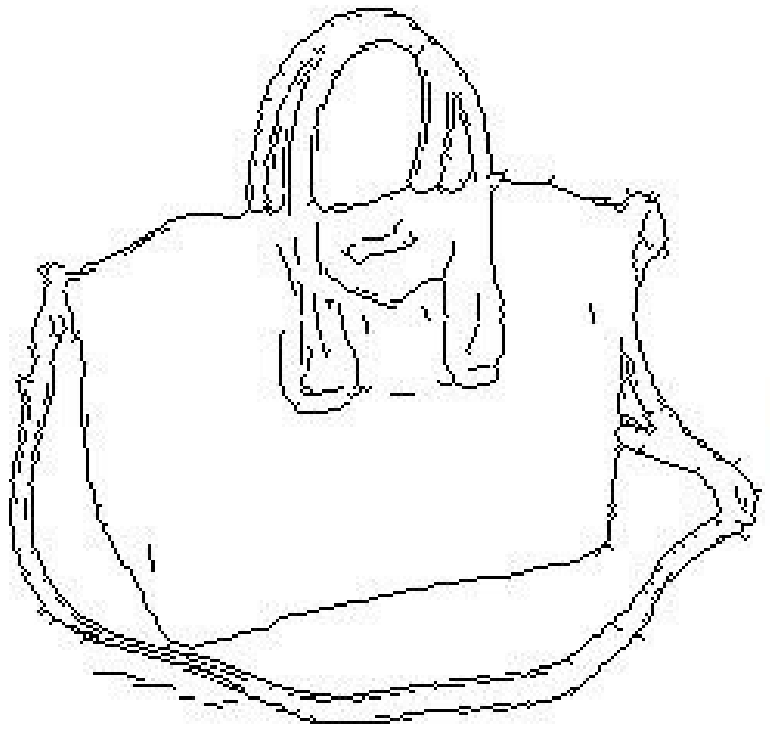
DCR
Ground-truth

Output



Edges → Images

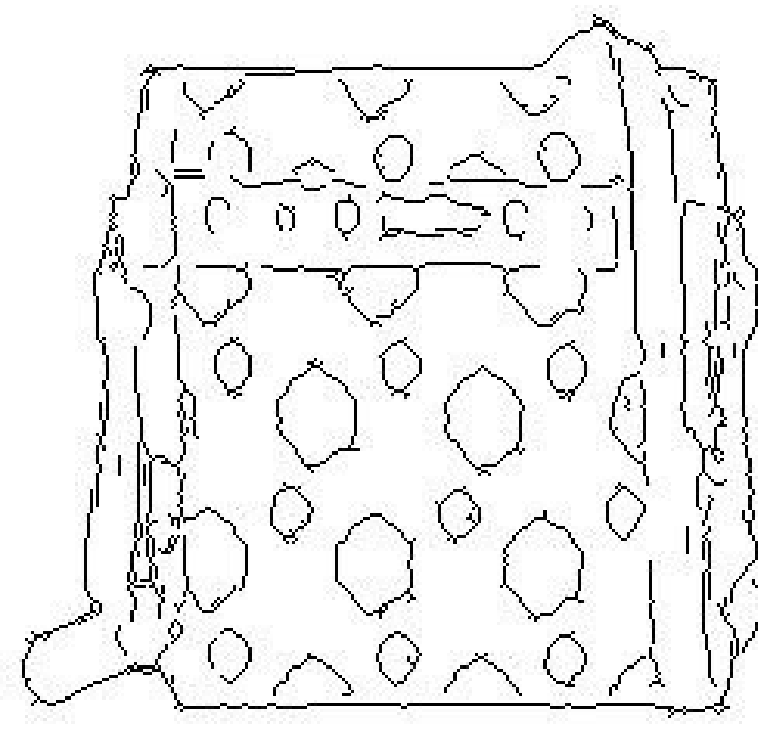
Input



Output



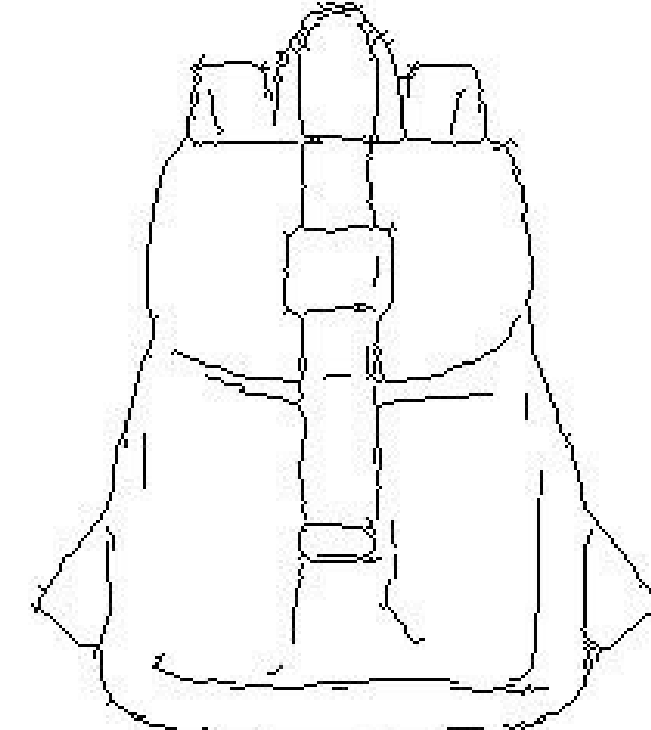
Input



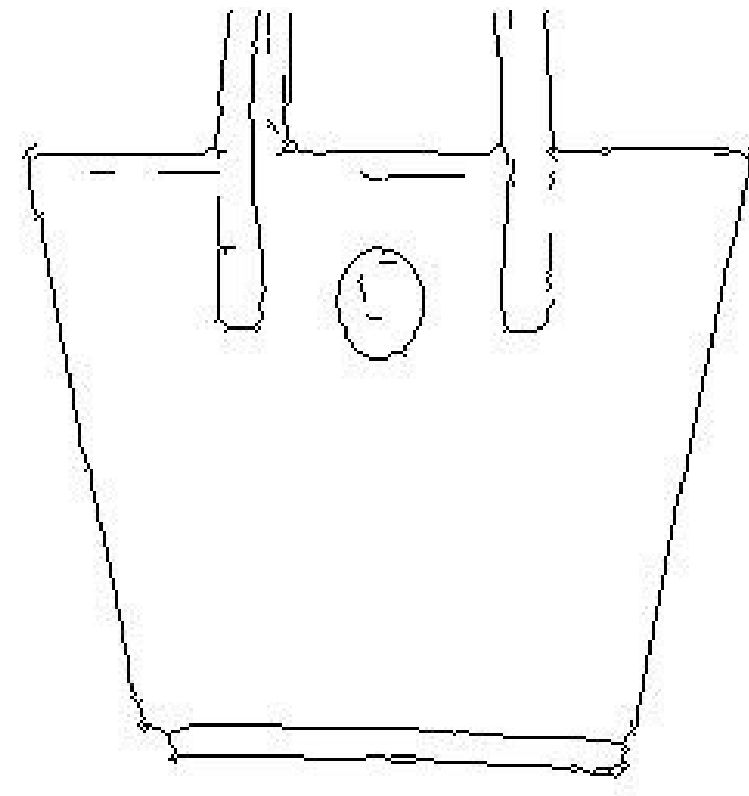
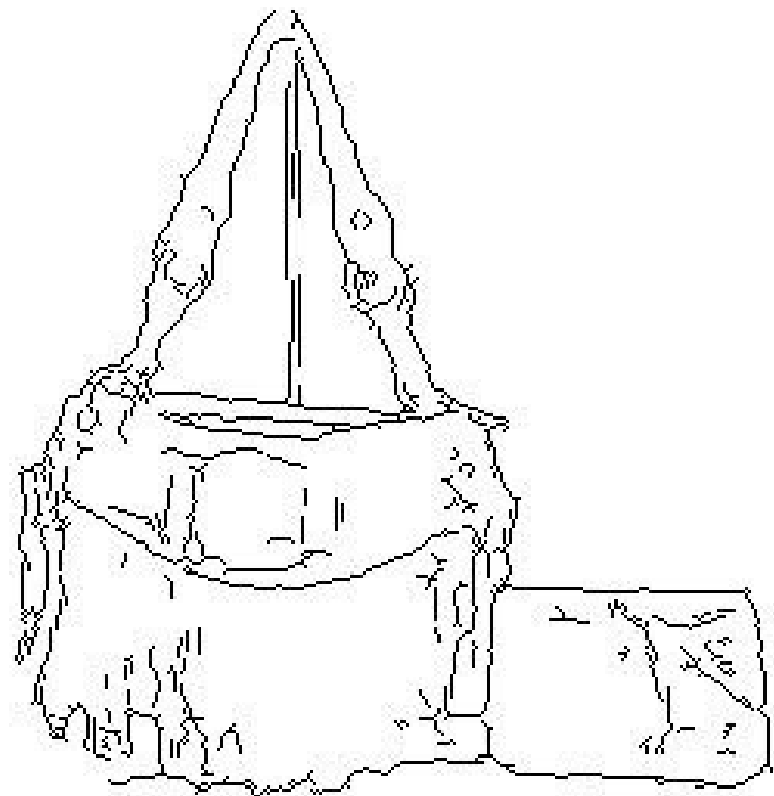
Output



Input

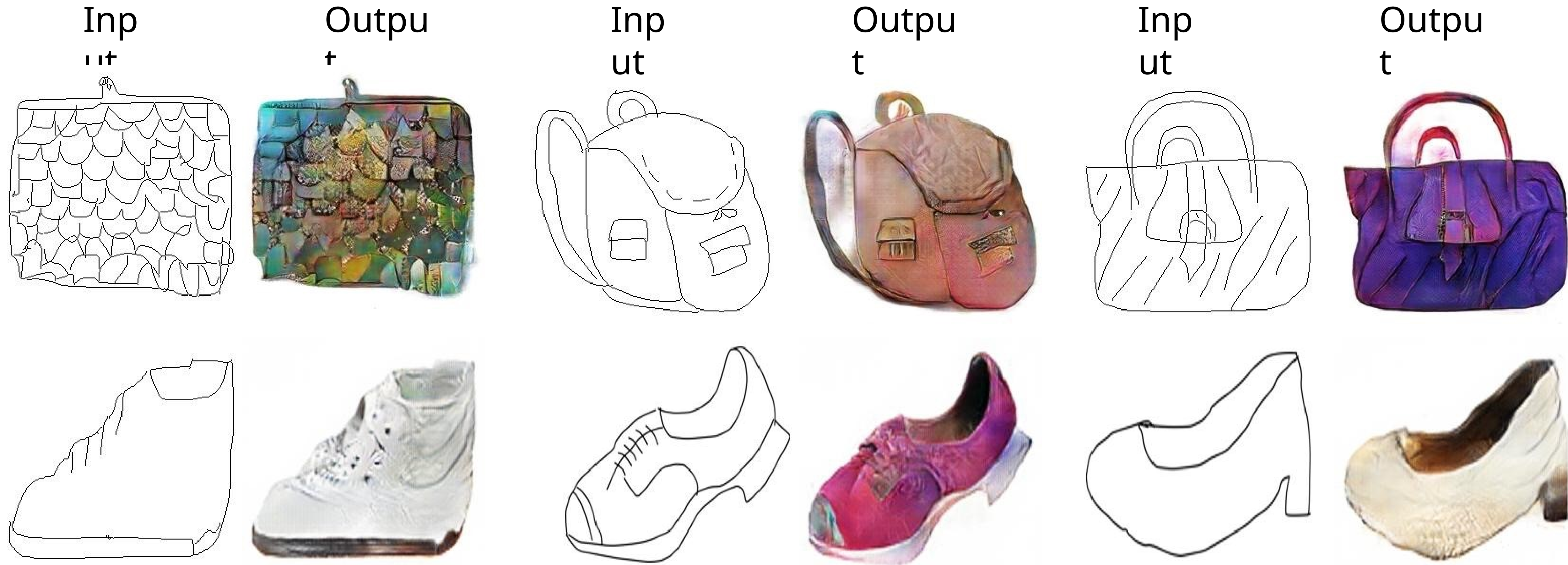


Output



Edges from [Xie & Tu,
2015]

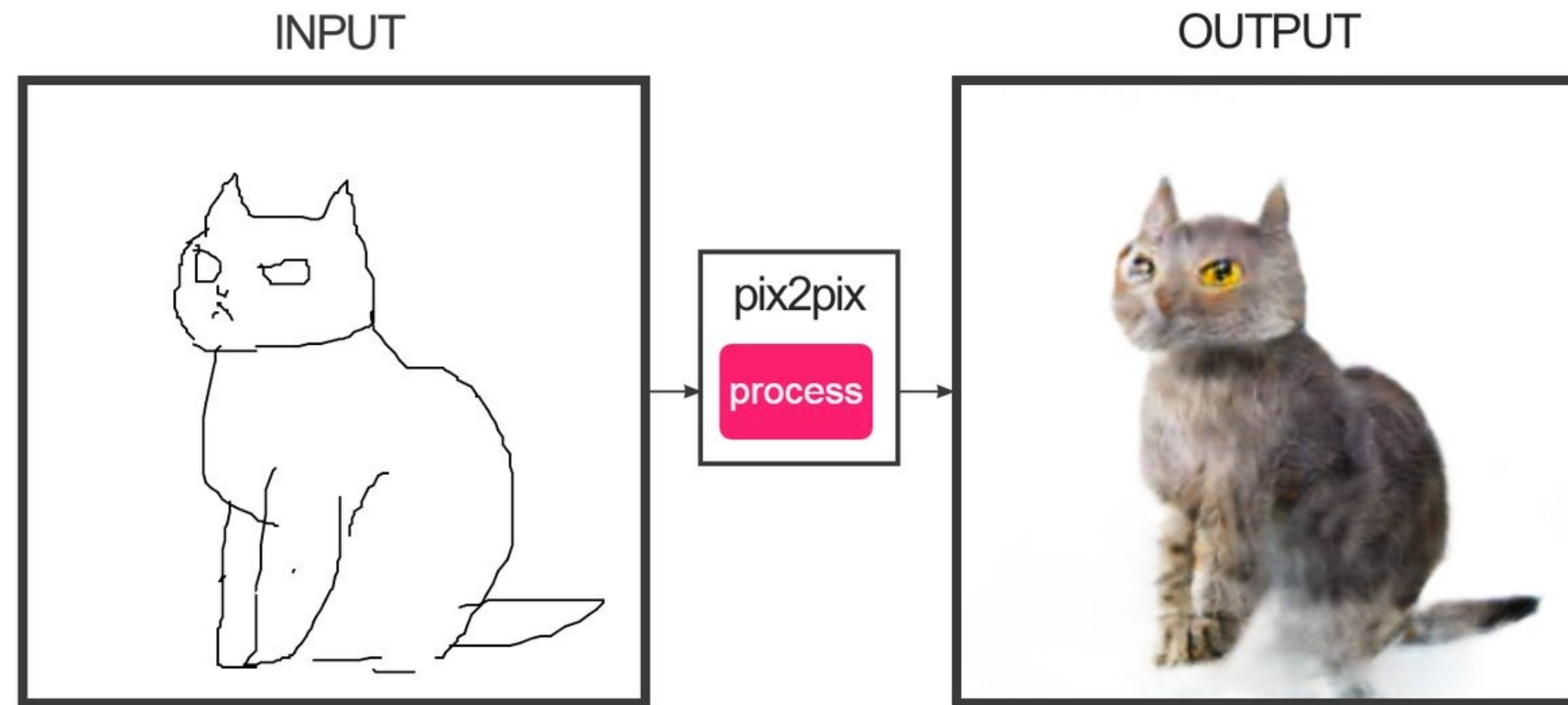
Sketches → Images



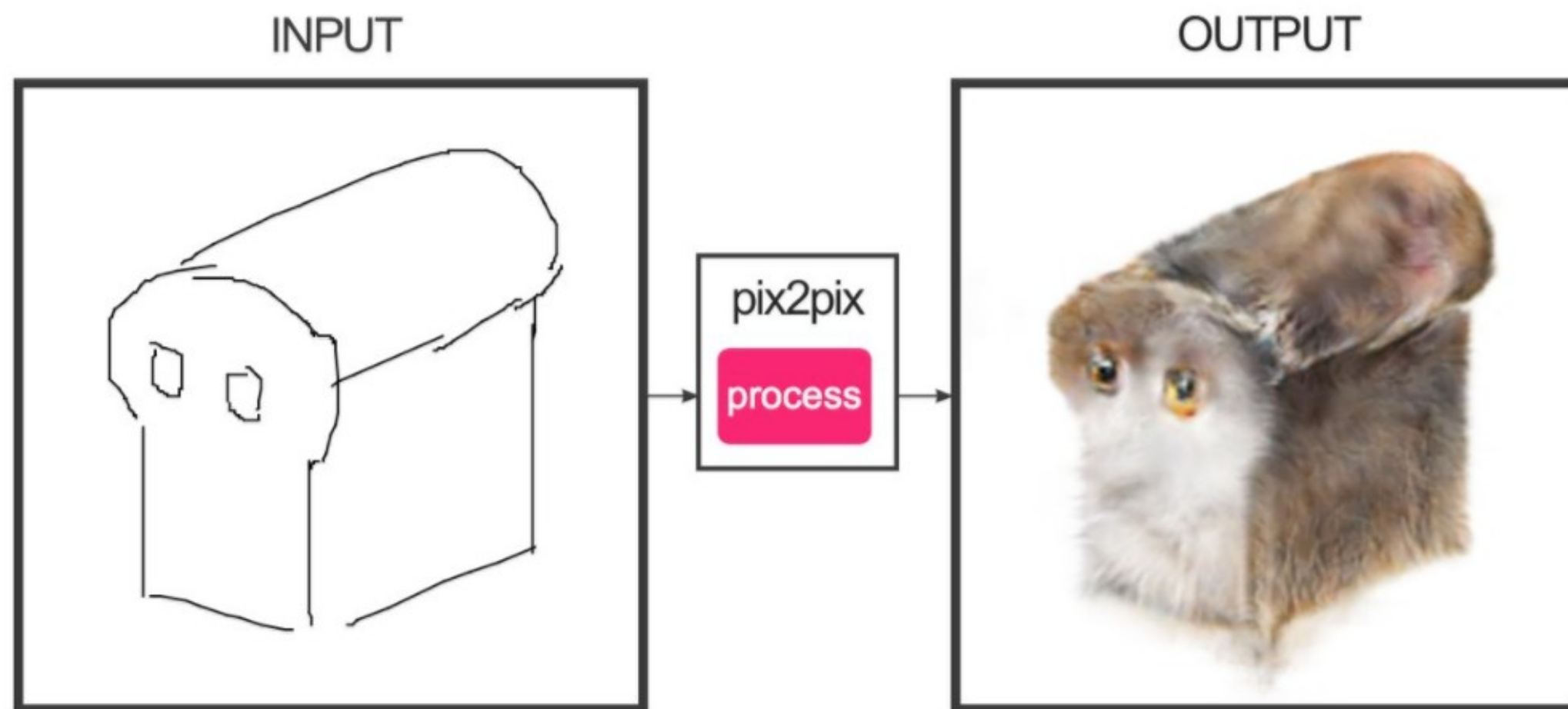
Trained on Edges →
Images

Data from [Eitz, Hays, Alexa,
2012]

#edges2cats [Christopher Hesse]



@gods_tail



Ivy Tasi @ivymyt

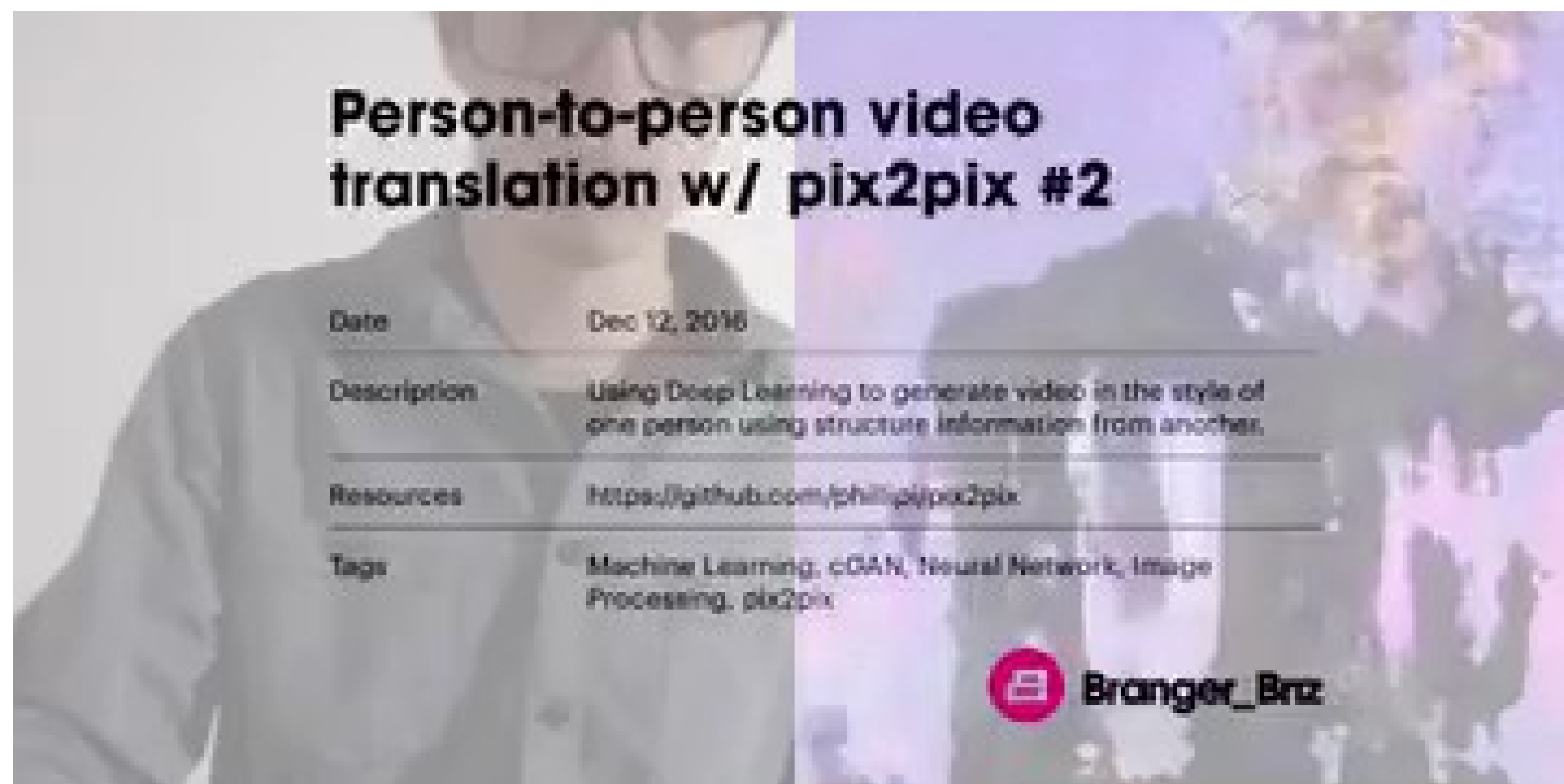


Vitaly Vidmirov @vvid



@ka92

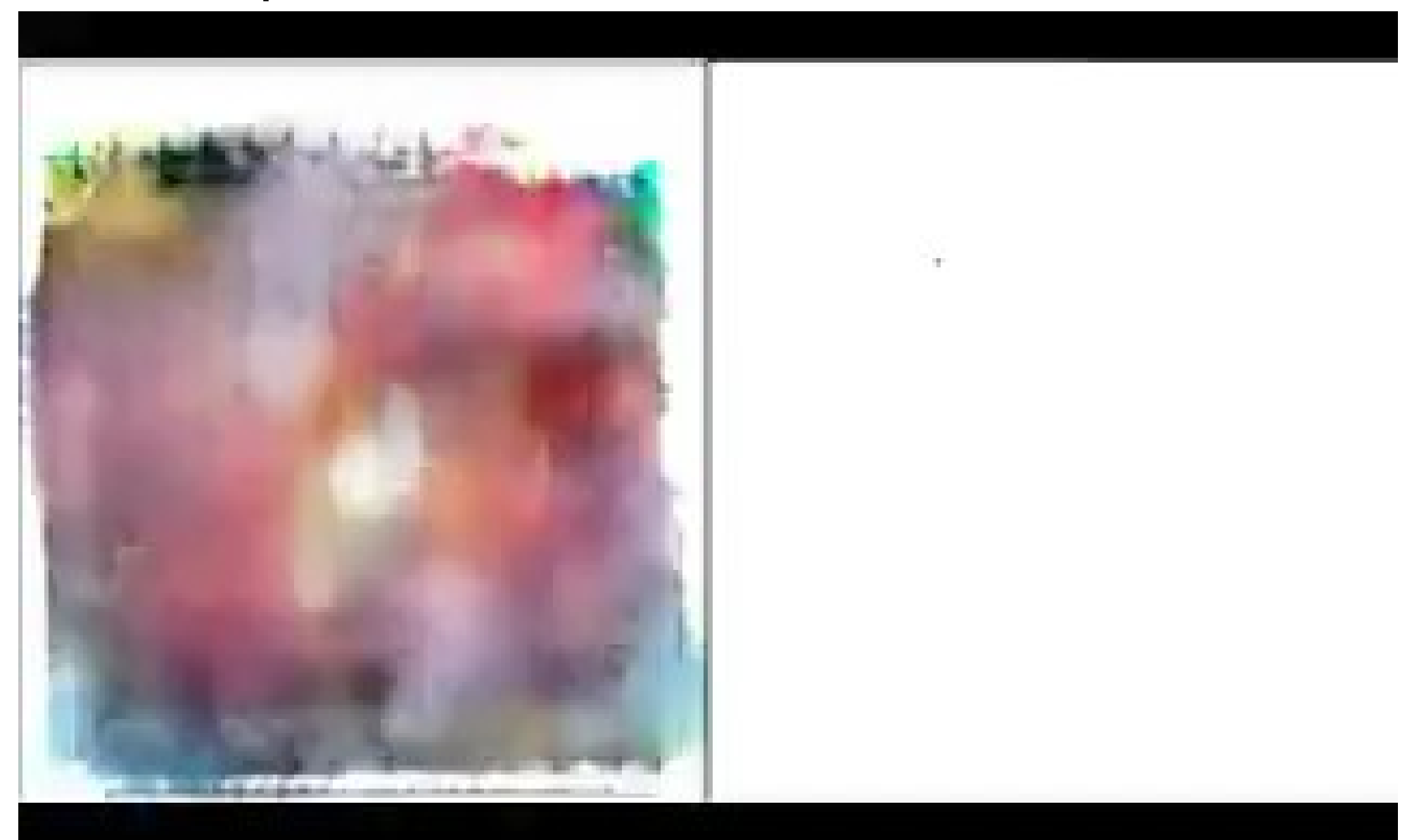
Twitter-driven research: #pix2pix



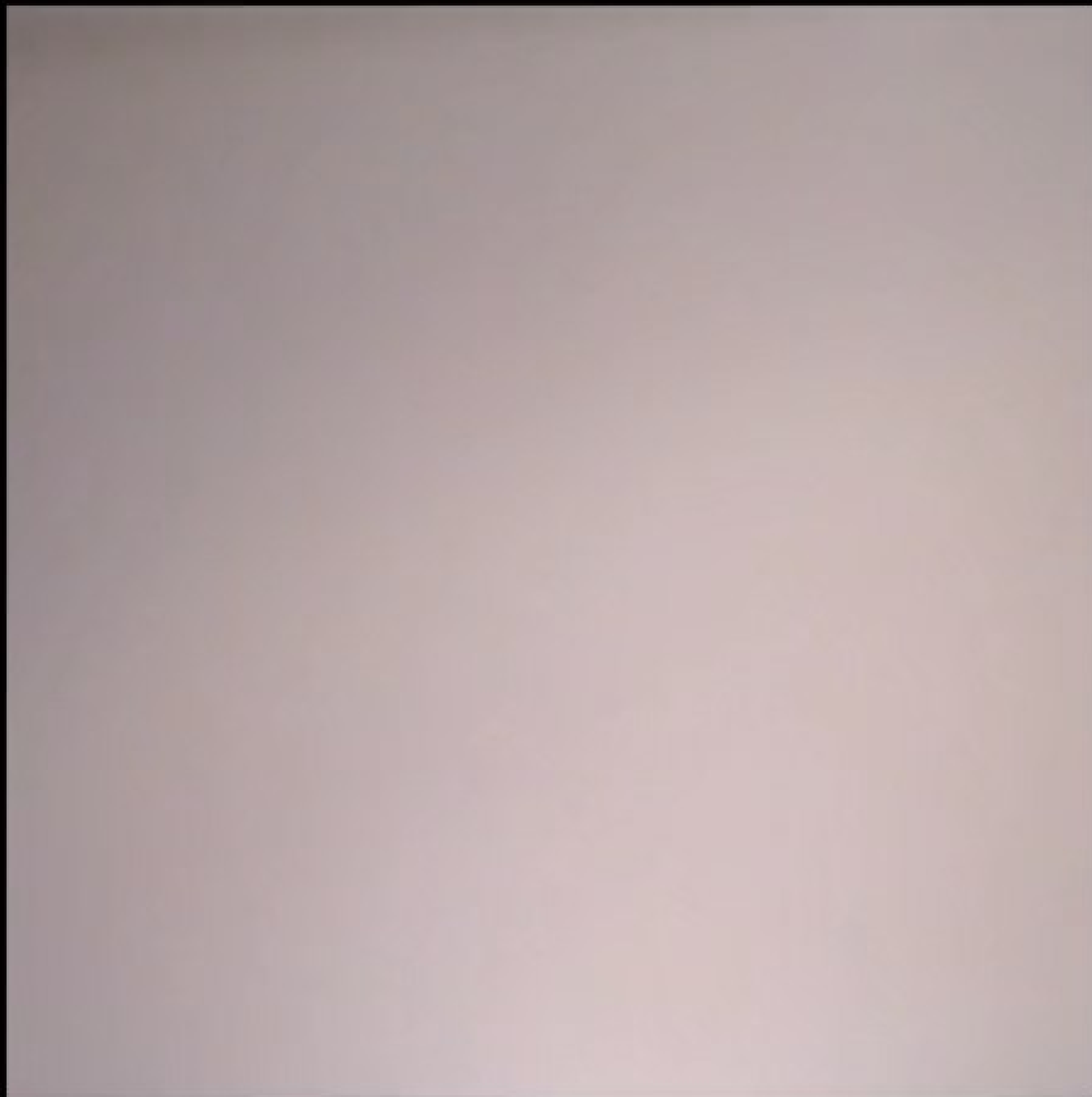
Brannon Dorsey @brannondorsey



Mario Klingemann
@quasimondo

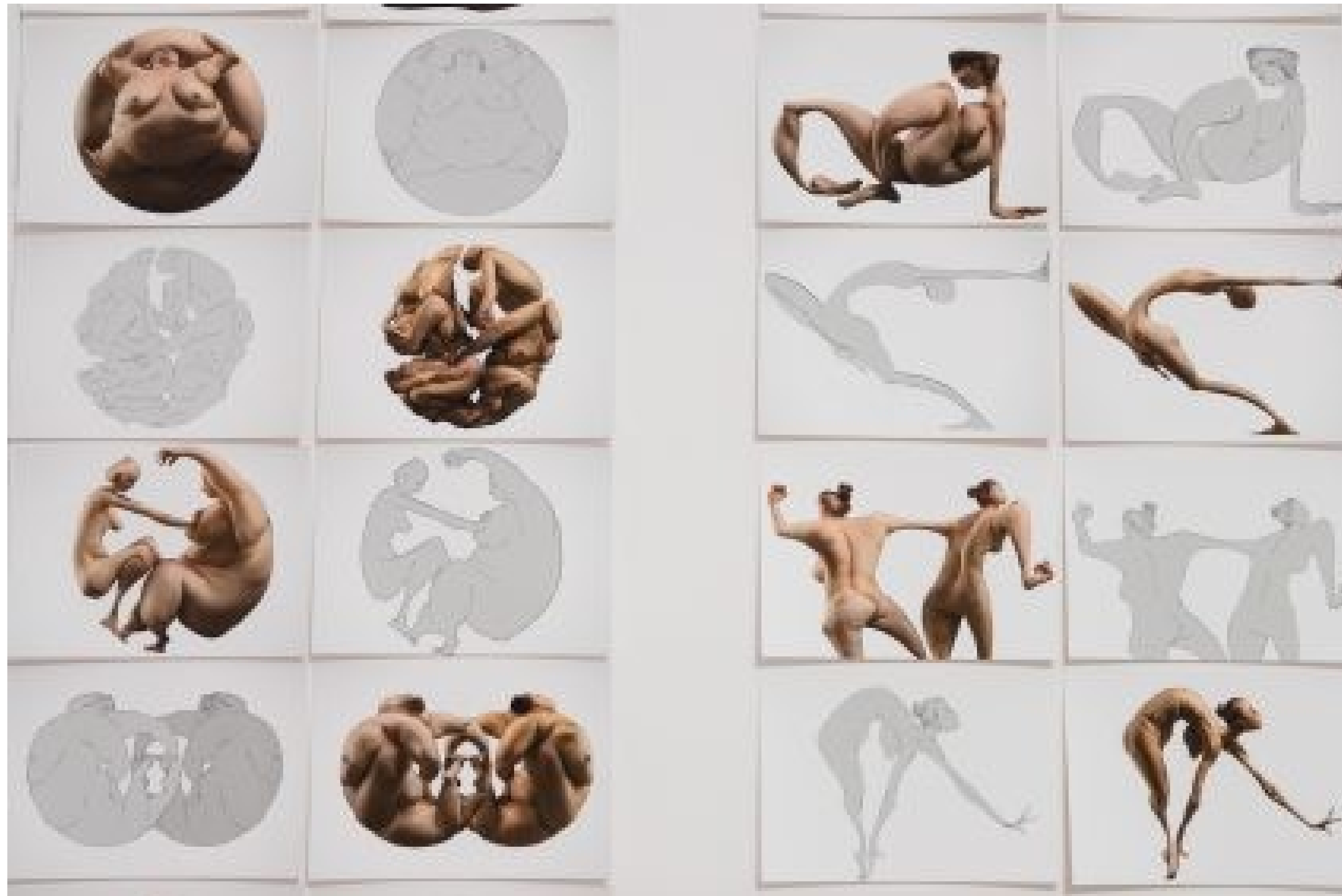


Bertrand Gondouin @bgondouin



© Memo Akten
“Learning to See:
Gloomy Sunday”

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





Paired training examples



Unpaired training examples

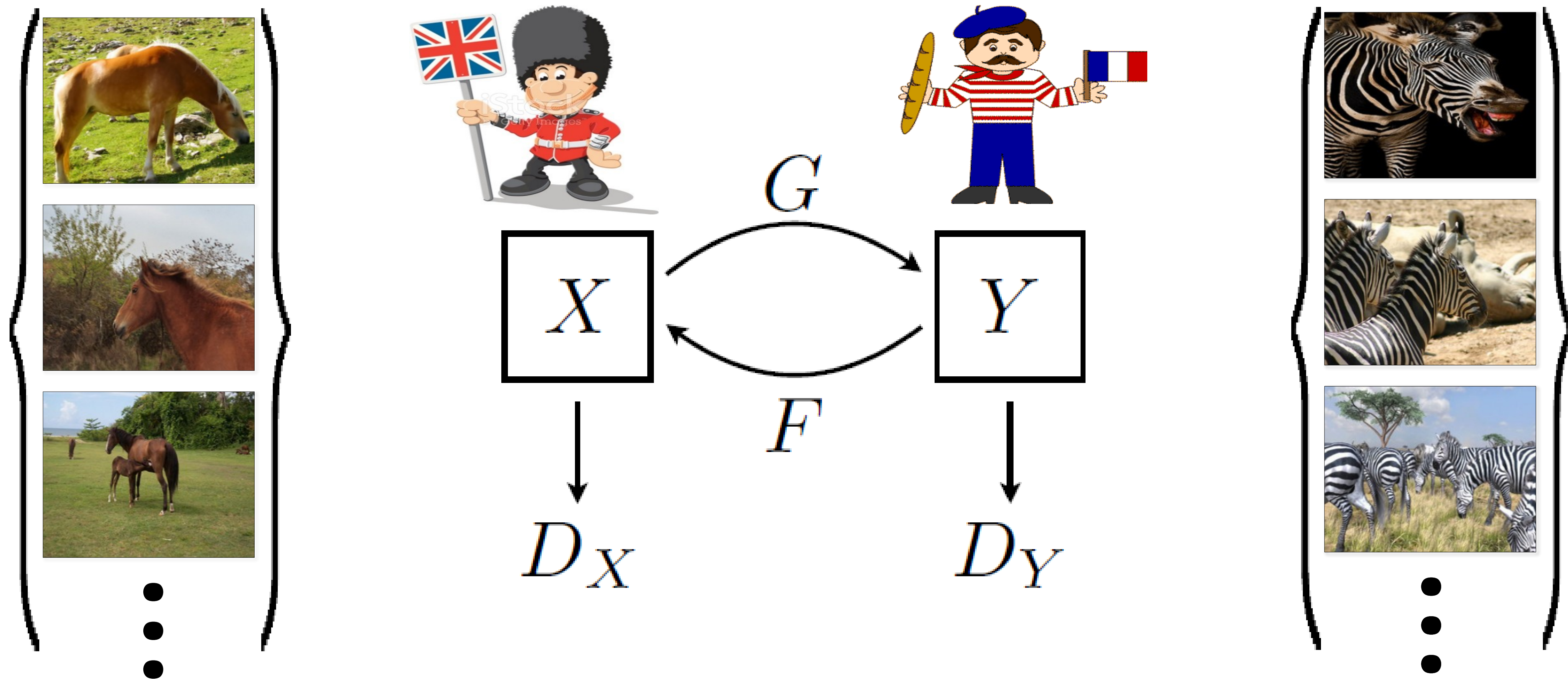
X



Y

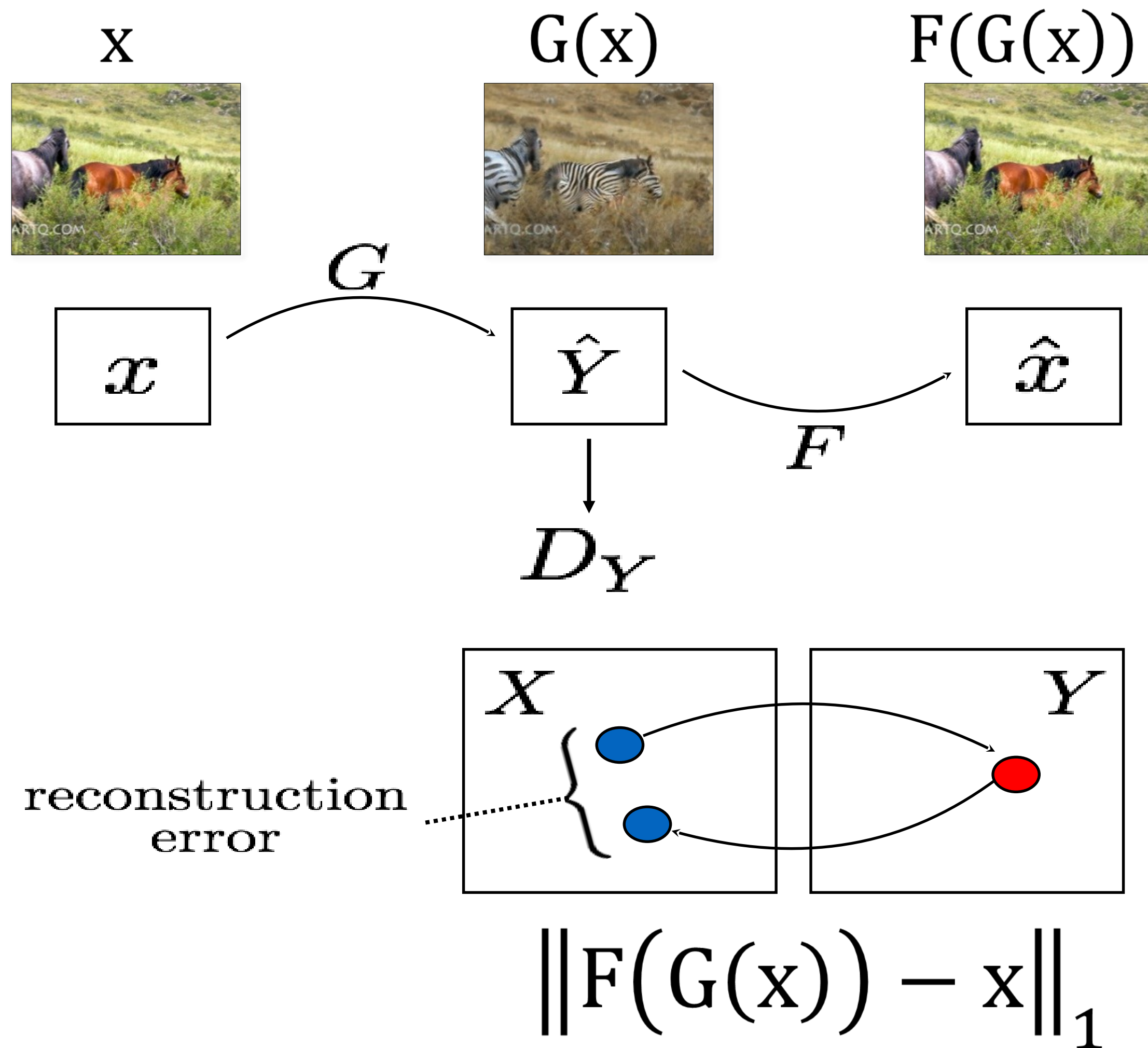


CycleGAN, or “there and back aGAN”

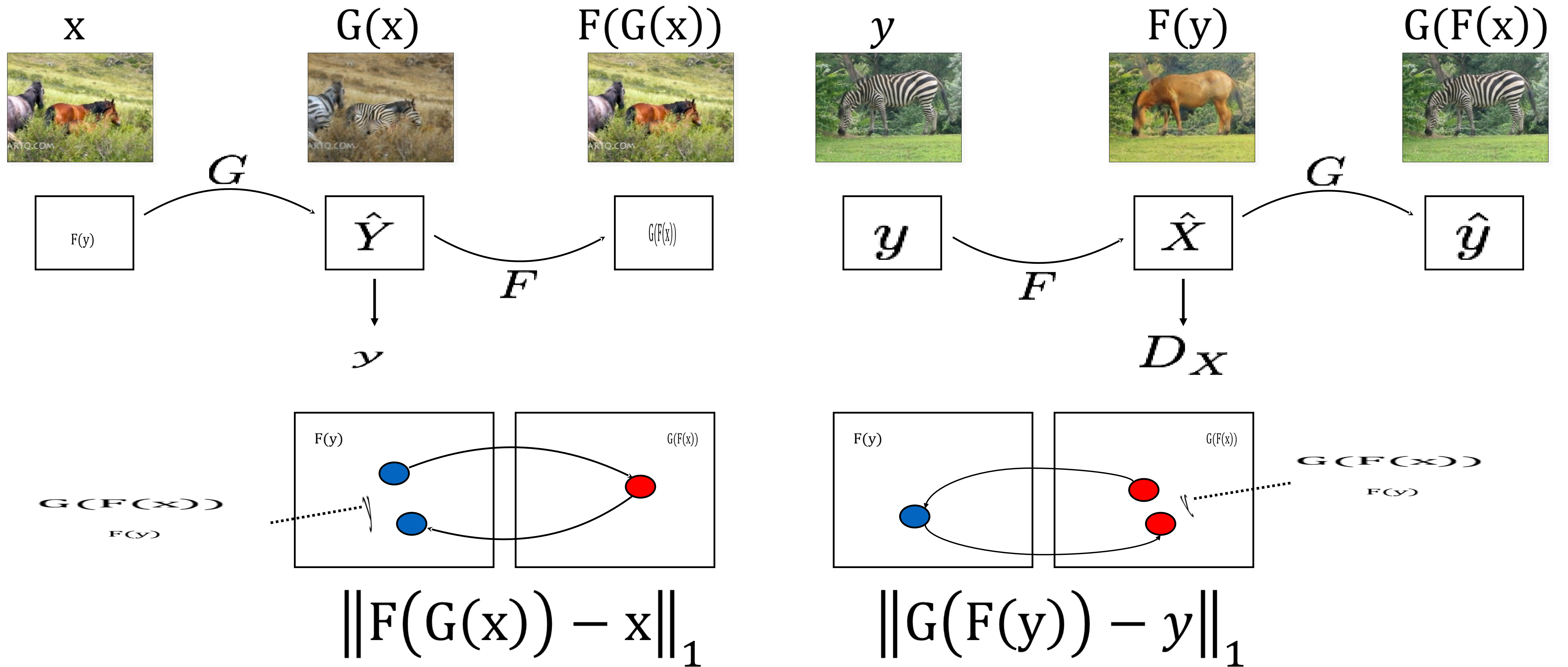


[Zhu*, Park*, Isola, Efros. ICCV 2017]

Cycle-Consistency Loss



Cycle-Consistency Loss





Video









Collection Style Transfer



Photograph
© Alexei Efros



Monet



Van Gogh



Cezanne



Ukiyo-e



CG to Real

Grand Theft
Auto



Real to CG



Shallower depth of field

Input

Output

Input

Output

Input

Output



Failure case

