Image Processing 2

Hakan Bilen
University of Edinburgh

Computer Graphics
Fall 2017

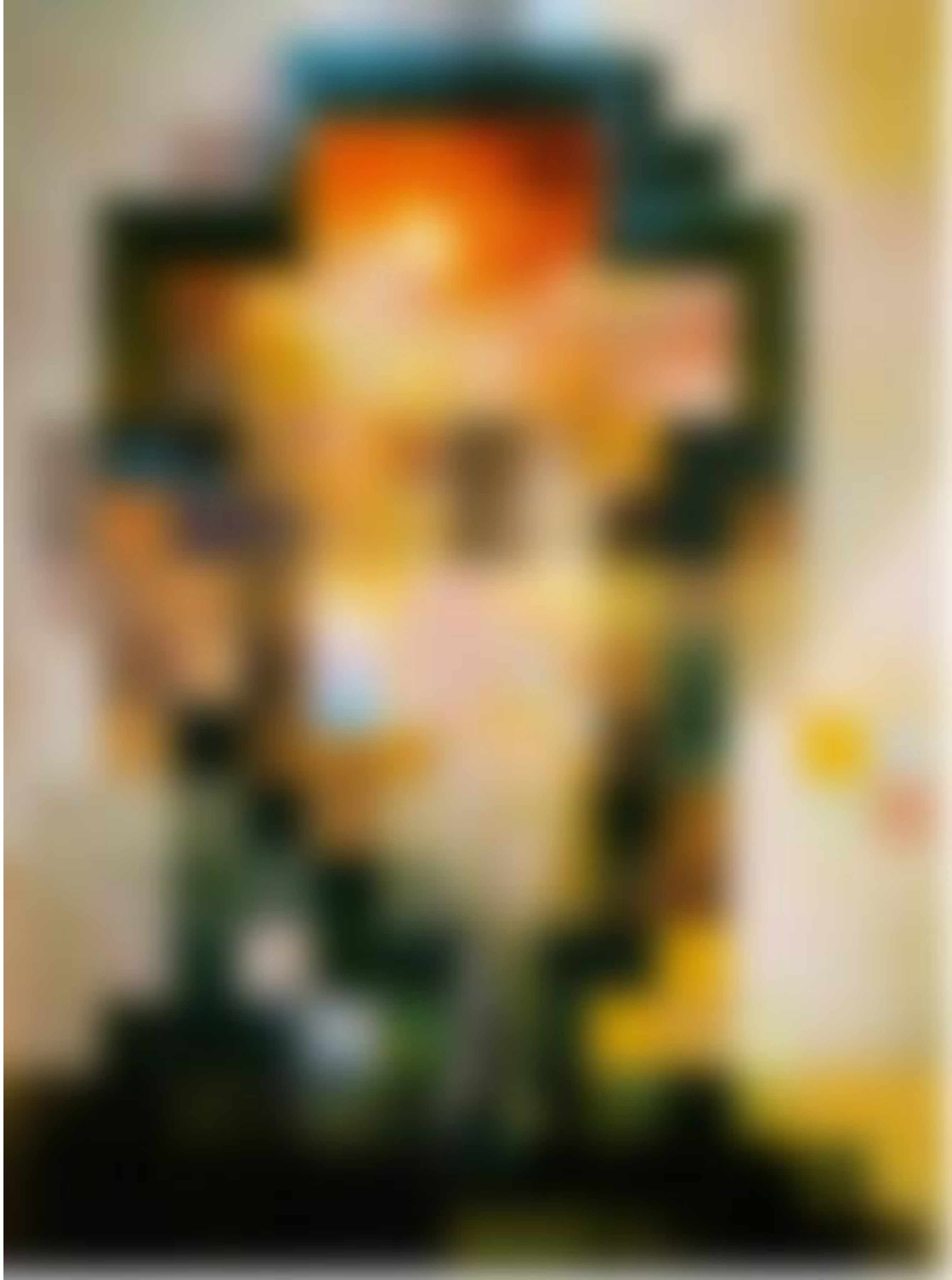
This week

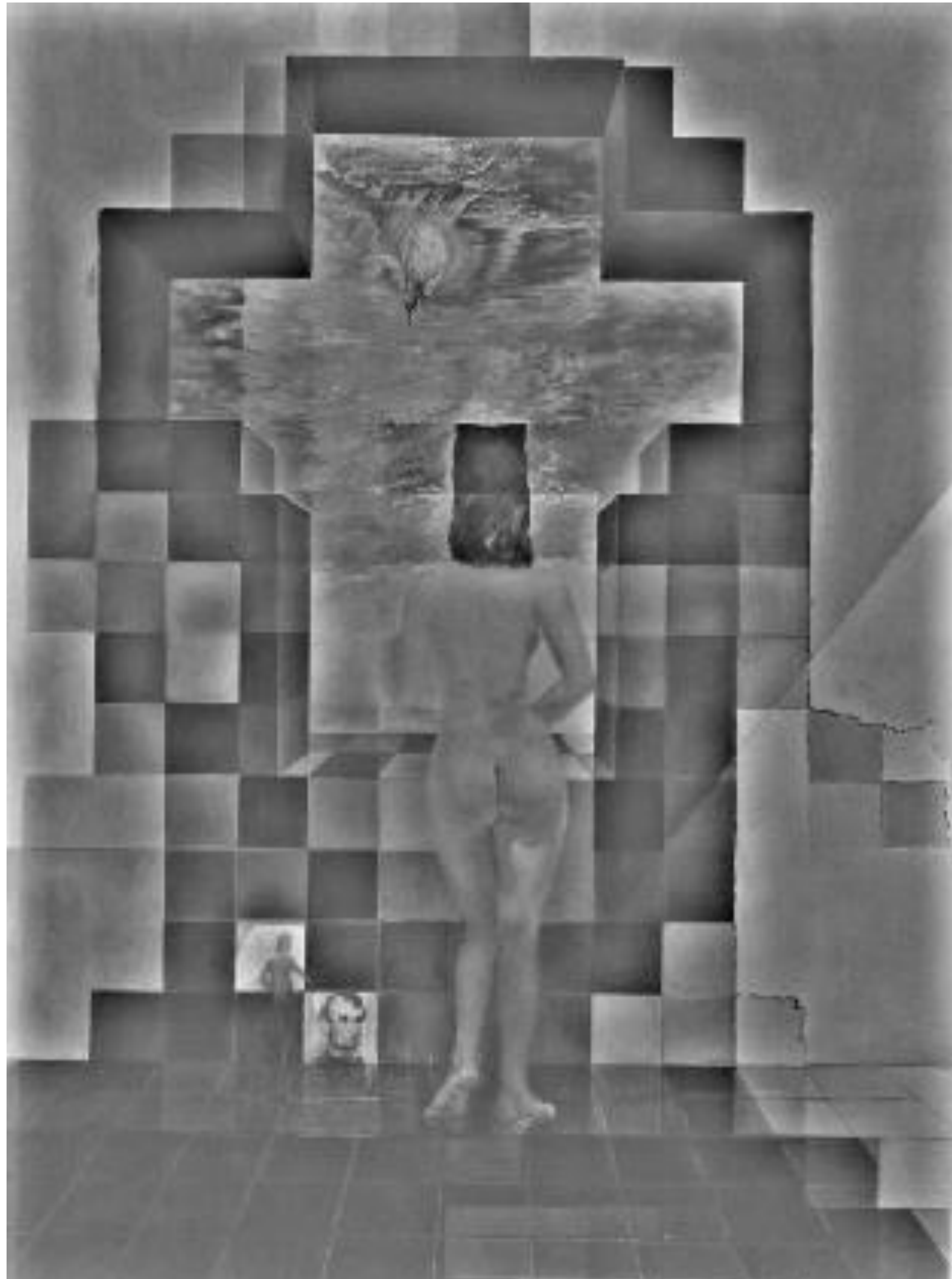
- What is an image?
- What is image processing?
- Point processing
- Linear (Spatial) filters
- **Frequency domain**
- **Deep image processing**

Salvador Dali
"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln", 1976

Slide credits: A. Efros

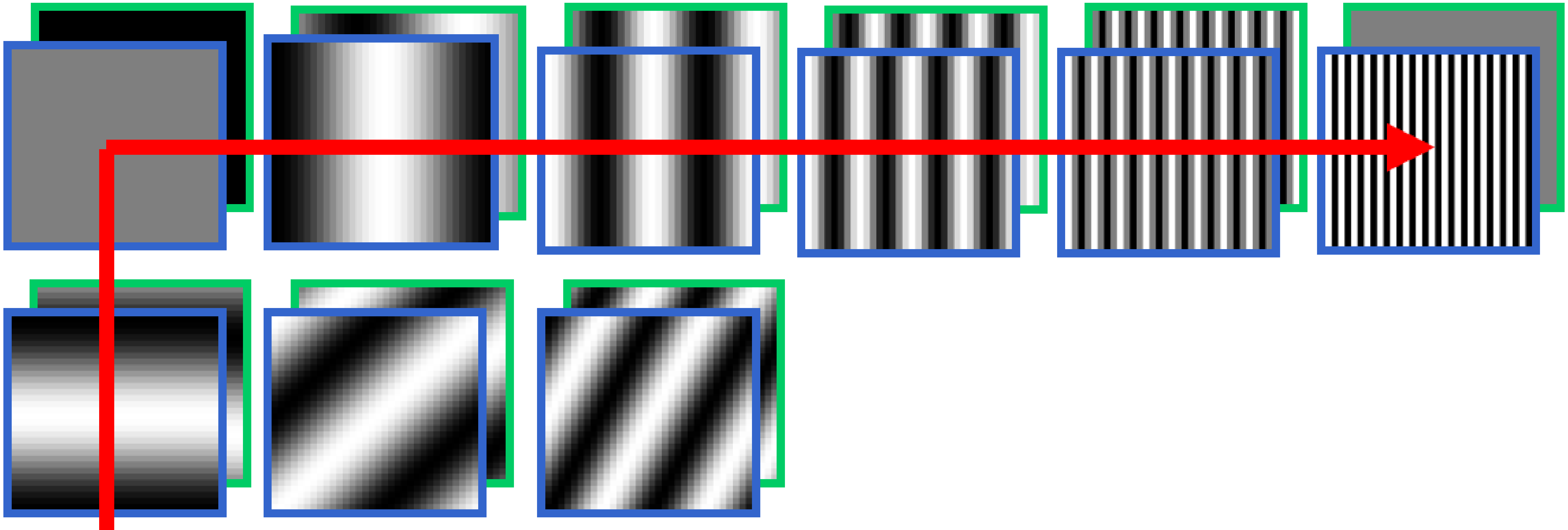
Why?

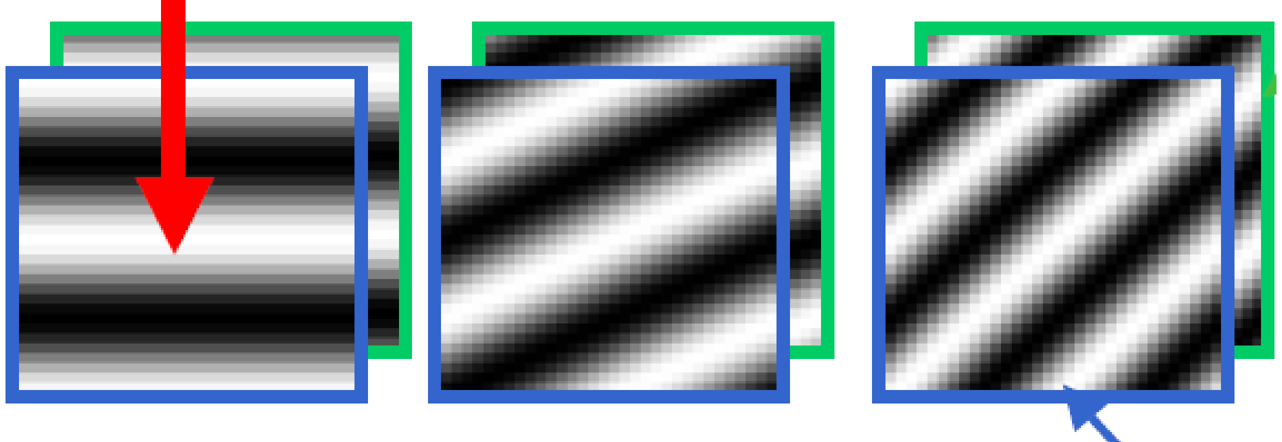




A set of basis functions

Teases away fast vs. slow changes in the image.





This change of basis has a special name…

Slide Credit: A. Efros

Jean Baptiste Joseph Fourier (1768-1830)

Fourier had a crazy idea (1807):

***Any** univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.*

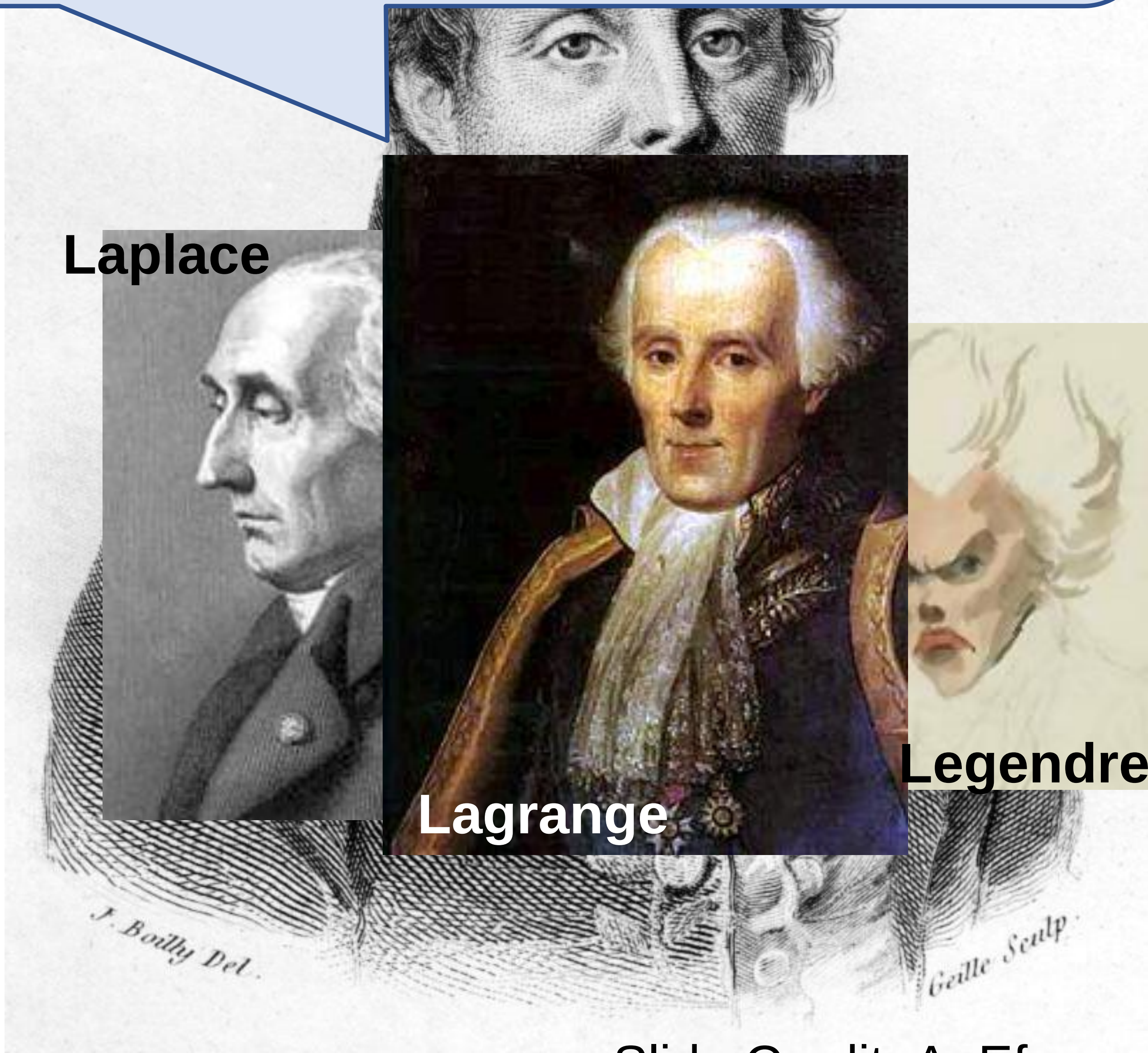
...the manner in which the author arrives at these equations is not exempt of difficulties and...his analysis to integrate them still leaves something to be desired on the score of generality and even rigour.

don't believe it?

- Neither did Lagrange, Laplace, Poisson and other big wigs
- Not translated into English until 1878!

but it's (mostly) true!

- called Fourier Series



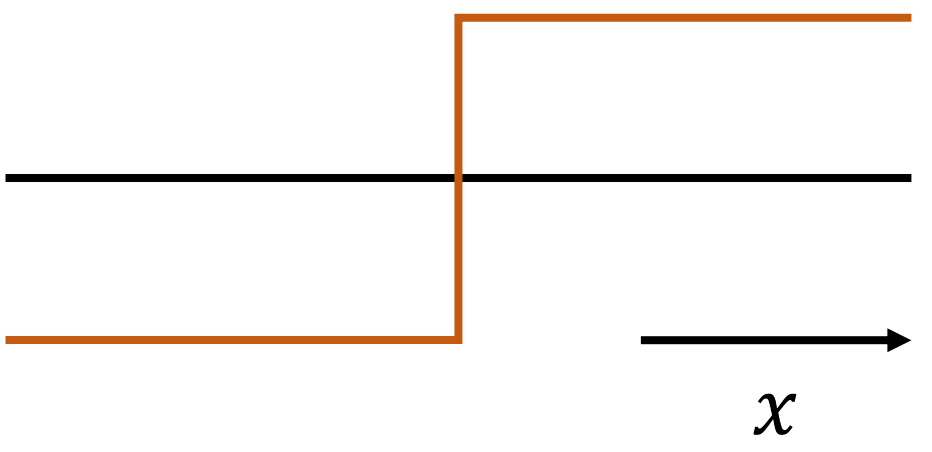
http://www-history.mcs.st-andrews.ac.uk/Biographies/Fourier.html

Slide Credit: A. Efros

1D Fourier Series

Example: A step edge

$$f(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases}$$



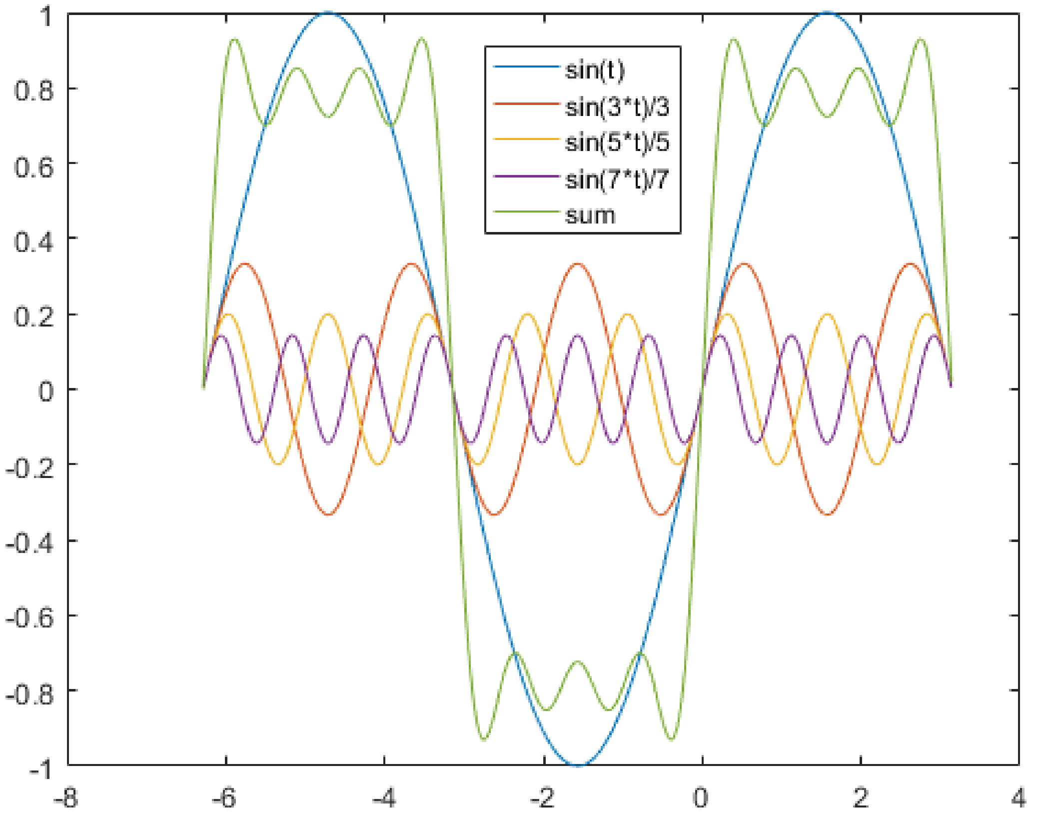
Fourier Decomposition

$$f(x) = \sum_{n=1} a_n \sin nx$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx \, dx$$

$$= \frac{2}{\pi} \int_0^{\pi} f(x) \sin nx \, dx = \begin{cases} \frac{4}{n\pi} & if \text{ n is odd} \\ 0 & \text{otherwise} \end{cases}$$

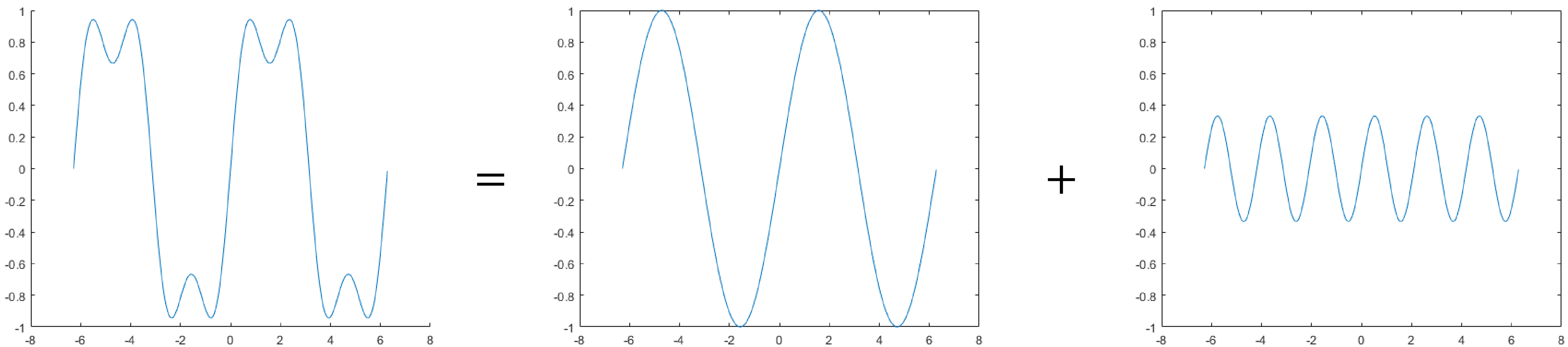
$$f(x) = \sum_{n=1} \frac{4}{(2n-1)\pi} \sin(2n-1)\, x$$



Slide Credit: A. Zisserman

1D Fourier Series

Example: A square wave

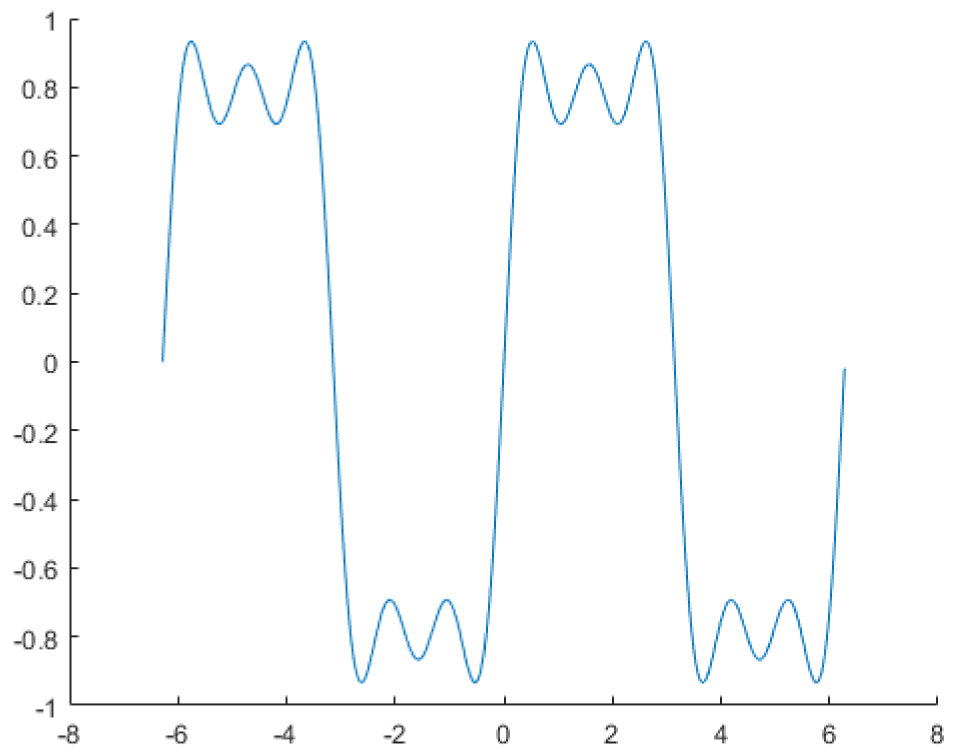


$$f(x) = sinx + \frac{1}{3}\sin 3x$$

Slide Credit: A. Zisserman

1D Fourier Series

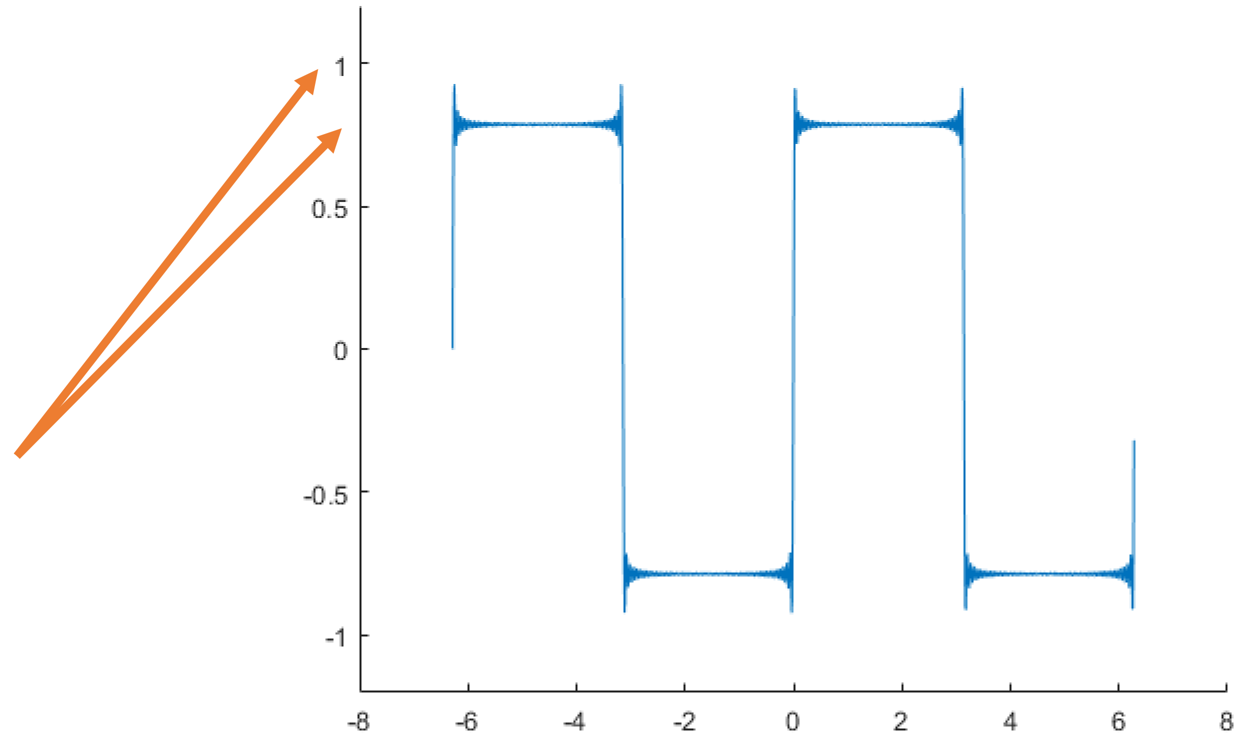
Example: A square wave



$$f(x) = sinx + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x$$

1D Fourier Series

Example: A square wave

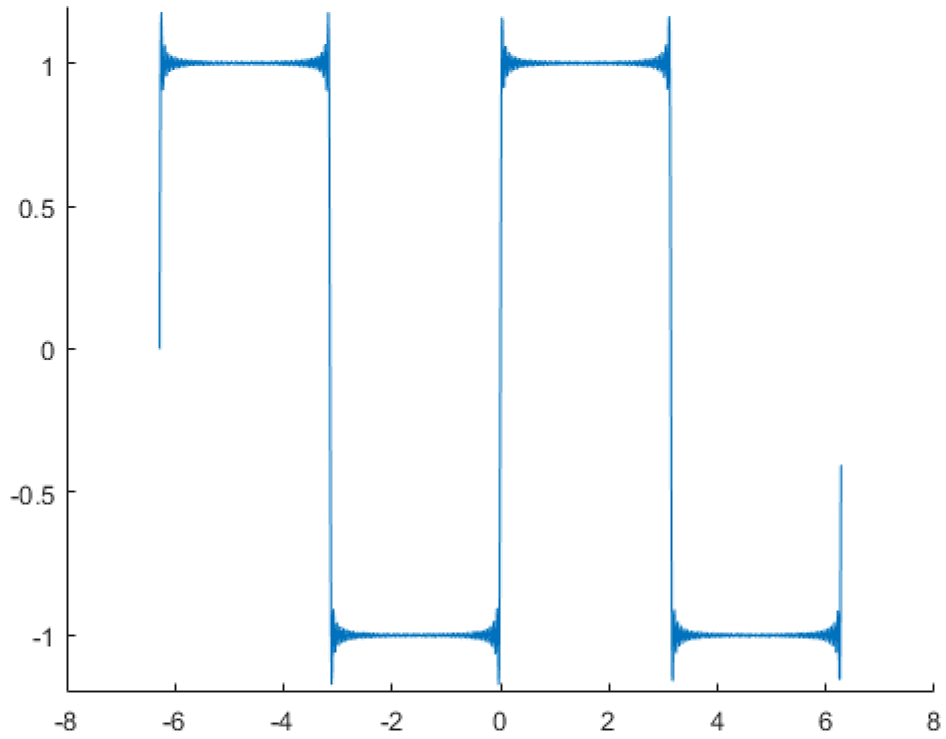


$$f(x) = sinx + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \cdots$$

Slide Credit: A. Zisserman

1D Fourier Series

Example: A square wave

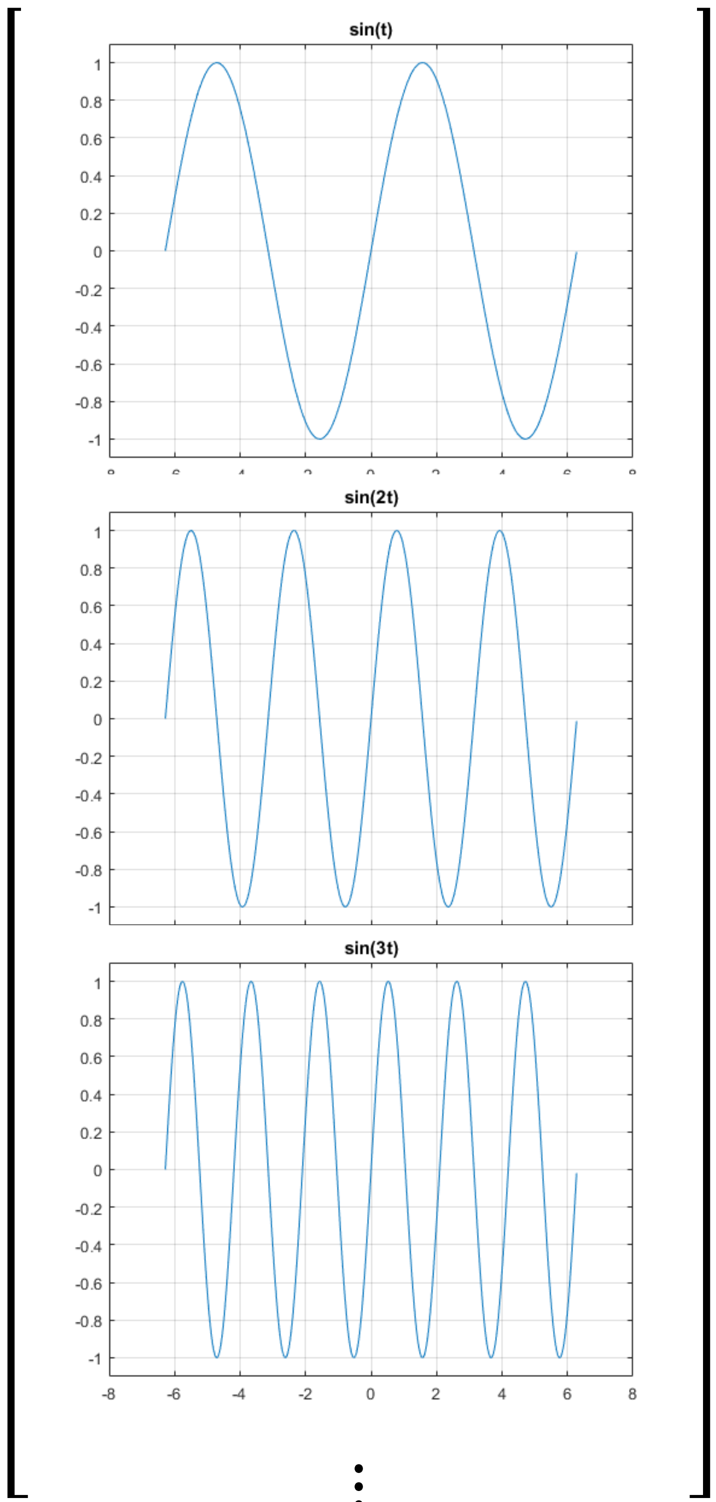


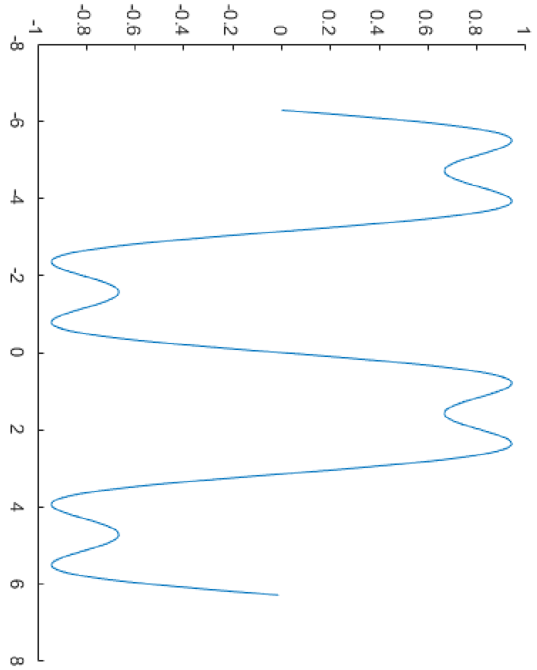
$$f(x) = \frac{4}{\pi}(sinx + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \cdots)$$

Slide Credit: A. Zisserman

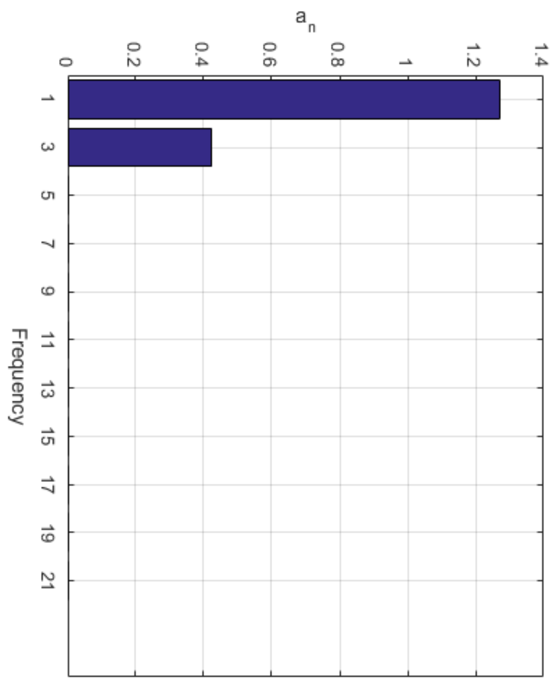
Fourier series: just a change of basis

$$Mf(x) = F(\omega)$$





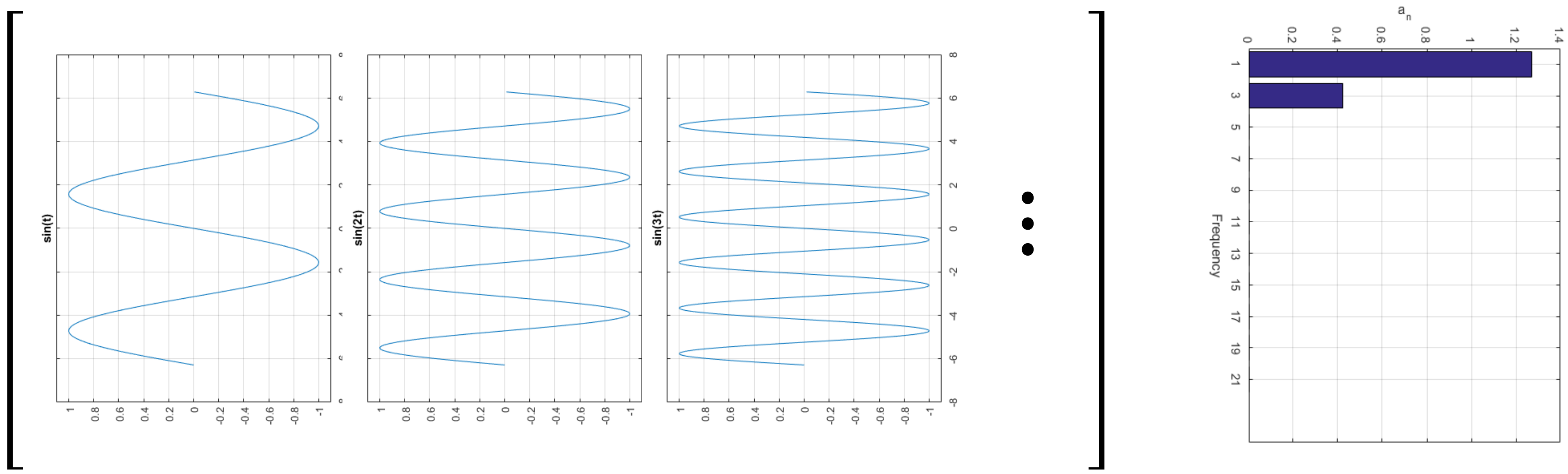
=

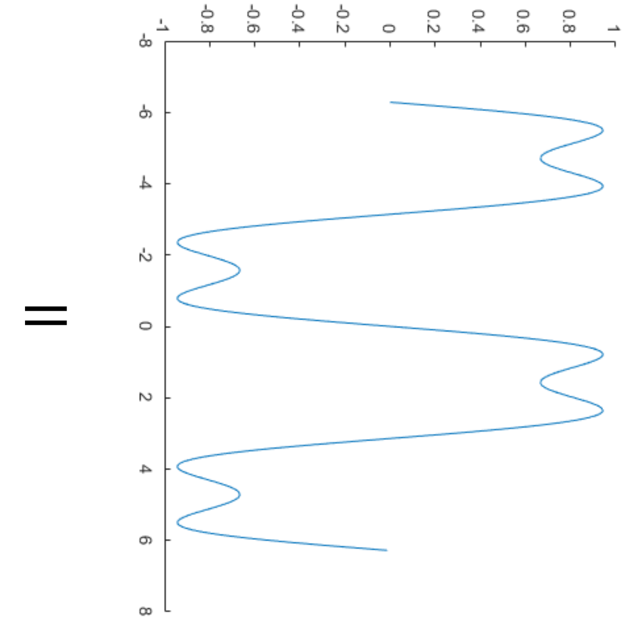


Slide Credit: A. Zisserman

Fourier series: just a change of basis

$$M^{-1}F(\omega) = f(x)$$





Slide Credit: A. Zisserman

1D/2D Fourier Transform

Transform pair for 1D

$$\mathrm{F}(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx,$$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

$$e^{j2\pi ux} = \cos(2\pi ux) + j sin(2\pi ux)$$

Transform pair for 2D

$$\mathrm{F}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dxdy,$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} dudv$$

- $\mathrm{F}(u, v)$ is complex in general,
  - $\mathrm{F}(u, v) = F_R(u, v) + \mathrm{j}\mathrm{F}_\mathrm{C}(u, v)$
  - $|\mathrm{F}(u, v)|$ is the magnitude spectrum
  - $\arctan(\frac{F_C(u,v)}{F_R(u,v)})$ is the phase spectrum

Slide Credit: A. Zisserman

Sinusoidal Waves

Fourier Transform is based on a decomposition into orthogonal basis functions:

$$e^{j2\pi ux} = \cos(2\pi ux) + jsin(2\pi ux)$$

Orthogonality: http://ms.mcmaster.ca/courses/20102011/term4/math2zz3/Lecture1.pdf

The maxima and minima occur when

$$2\pi(ux + vy) = n\pi$$

write $ux + vy$ in vector notation with $\mathbf{u} = (\mathrm{u}, v)^T$ and $\mathbf{x} = (\mathrm{x}, y)^T$ then

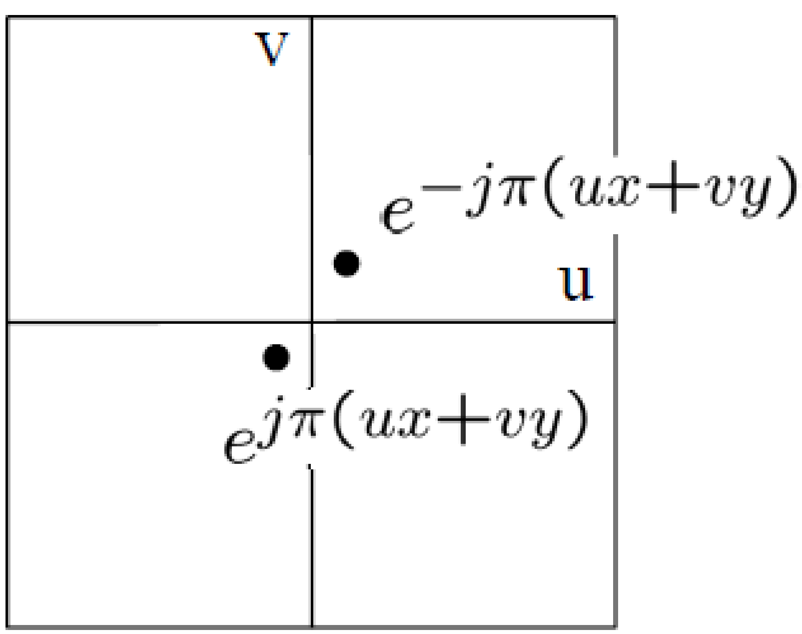
$$2\pi(ux + vy) = 2\pi\boldsymbol{u} \cdot \boldsymbol{x} = n\pi$$

are sets of equally spaced parallel lines with normal $\boldsymbol{u}$ and wavelength $\frac{1}{\sqrt{u^2+v^2}}$

Slide Credit: A. Zisserman

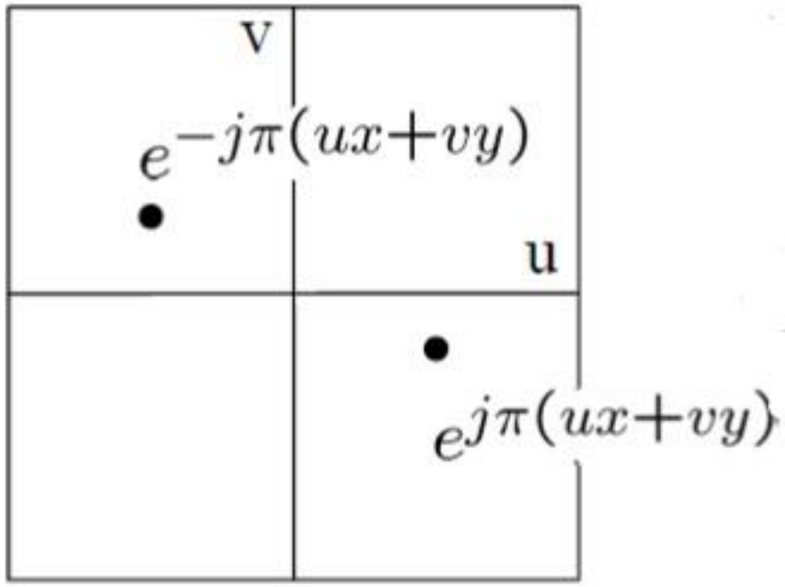
The real component of Fourier basis elements shown as intensity images

- Plot a basis element --- or rather, its real part --- as a function of x,y for some fixed (u,v). We get a function that is constant when (ux+vy) is constant.
- The magnitude of the vector (u, v) gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.



Slide Credit: B. Freeman

Here the magnitude of the vector (u, v) is larger than the previous one and angle is different



Slide Credit: B. Freeman

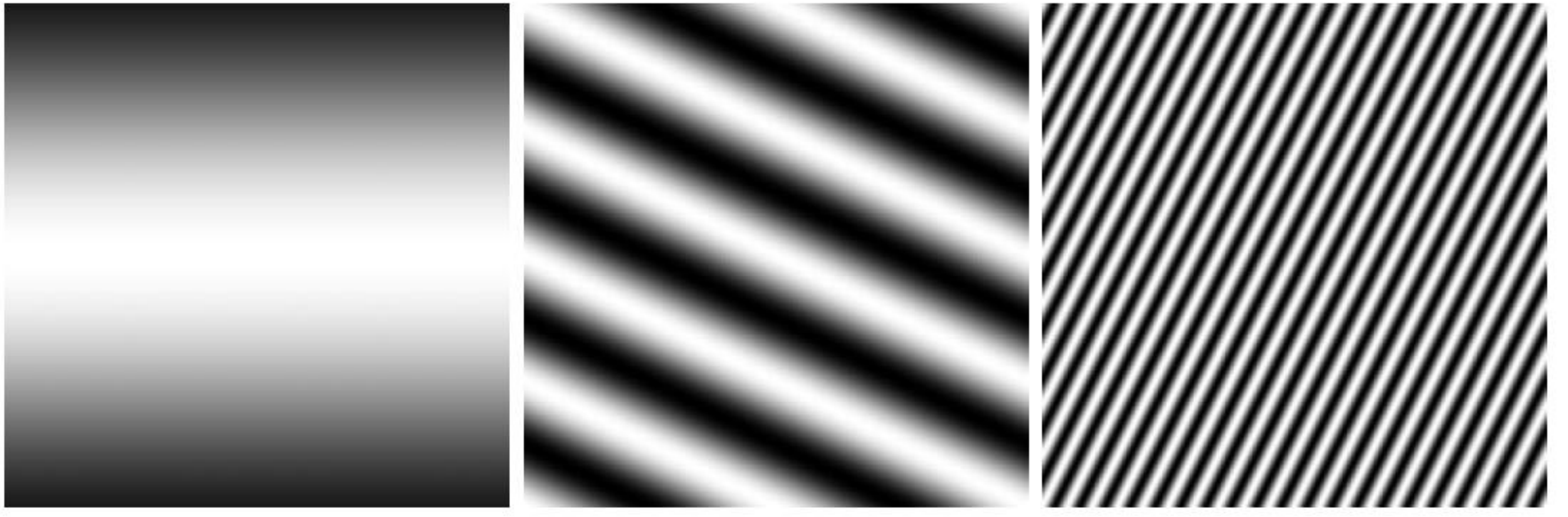
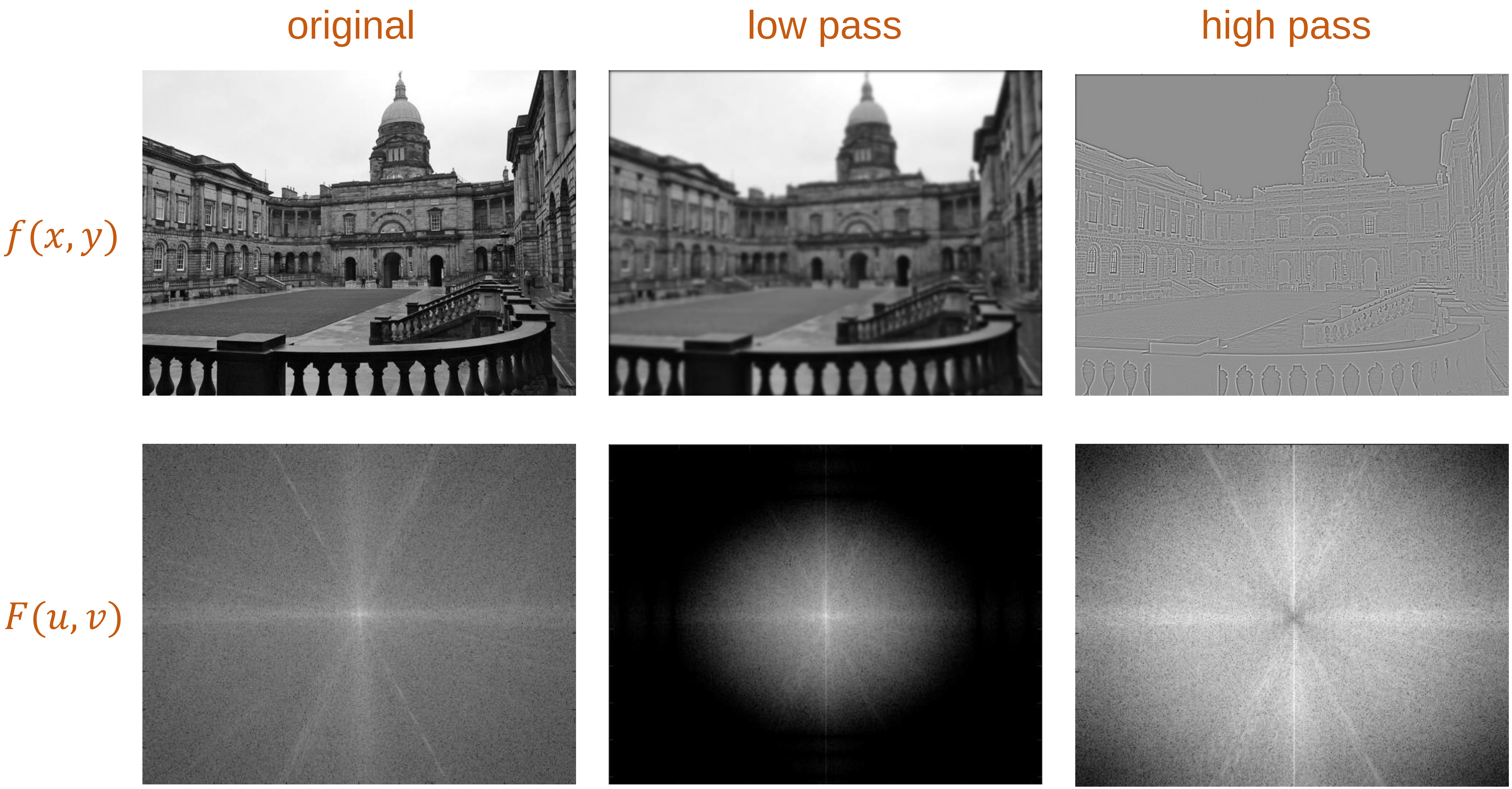


FIGURE 4.5: The real component of Fourier basis elements shown as intensity images. The brightest point has value one, and the darkest point has value zero. The domain is $[-1, 1] \times [-1, 1]$, with the origin at the center of the image. On the **left**, $(u, v) = (0, 0.4)$; in the **center**, $(u, v) = (1, 2)$; and on the **right** $(u, v) = (10, -5)$. These are sinusoids of various frequencies and orientations described in the text.

Image Credit: Forsyth & Ponce



original
low pass
high pass
$f(x, y)$
$F(u, v)$

Convolution Theorem

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

Space convolution = frequency multiplication

In words: the Fourier transform of the convolution of two functions is the product of their individual Fourier transforms

Why is it so important?

Because linear filtering operations can be carried out by simple multiplications in the Fourier domain

Slide Credit: A. Zisserman

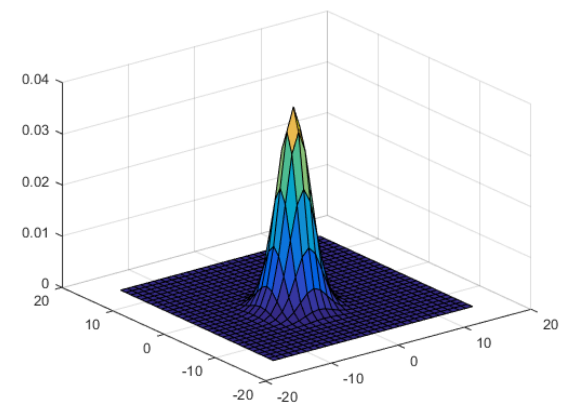
Convolution Theorem

The link between operations in the action of linear filters and frequency domain

Example

$f(x,y)$

$*$



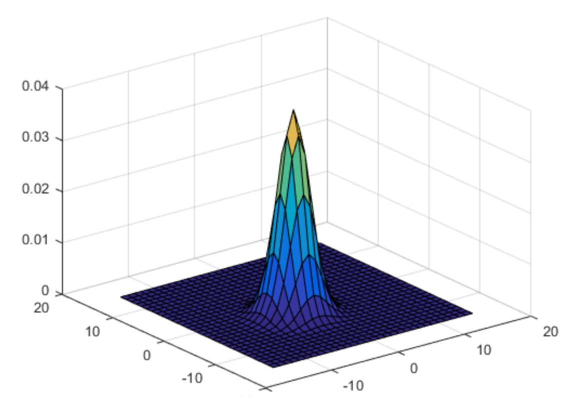
$g(x,y)$

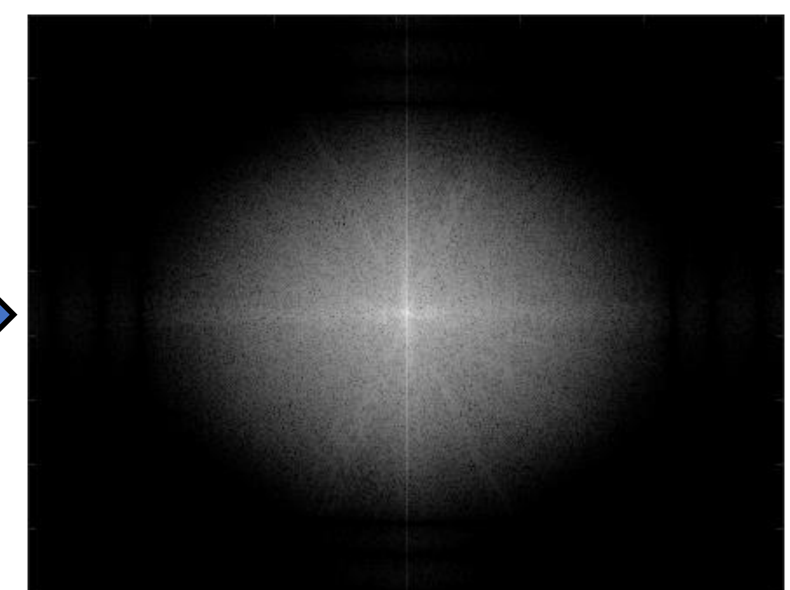
Fourier transform

inverse Fourier transform

$|F(u,v)|$

$\times$

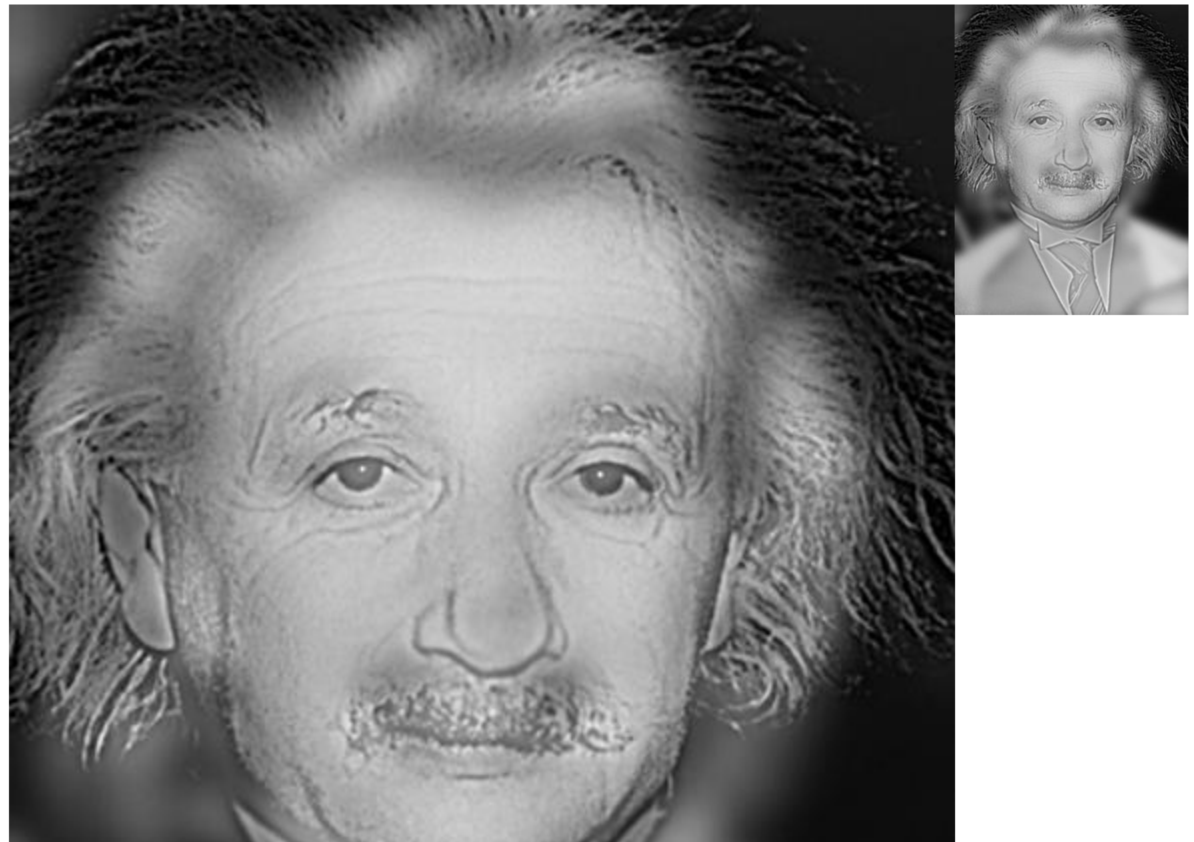


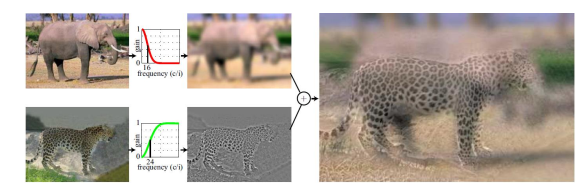


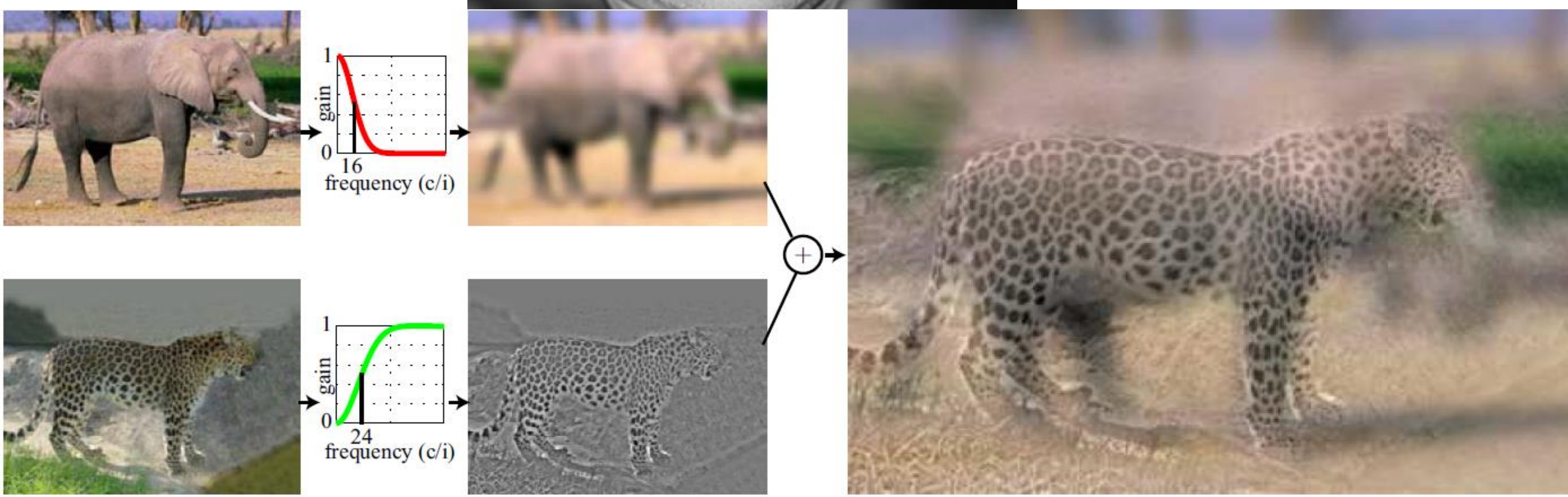
$|G(u,v)|$

Fourier Transform: Applications

Image Blending (Oliva etal 2006)

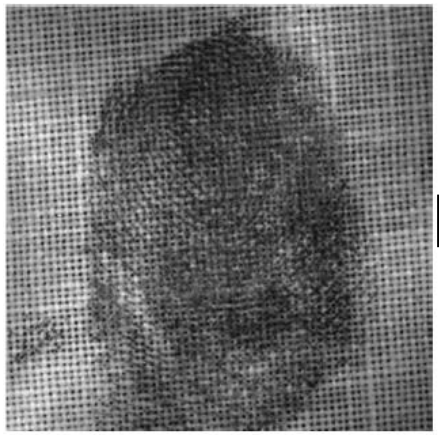


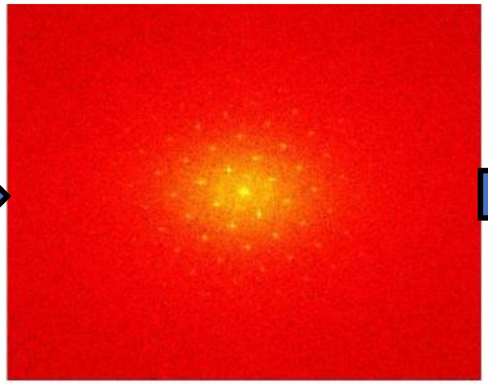




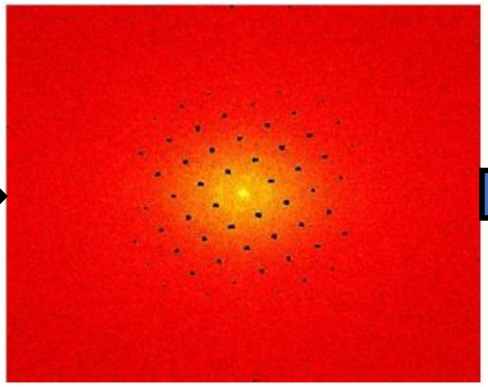
Fourier Transform: Applications

Forensic application

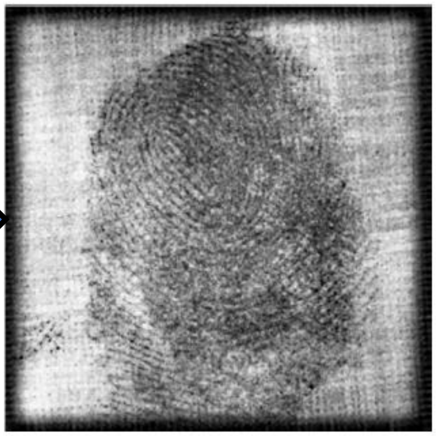




$|F(u,v)|$

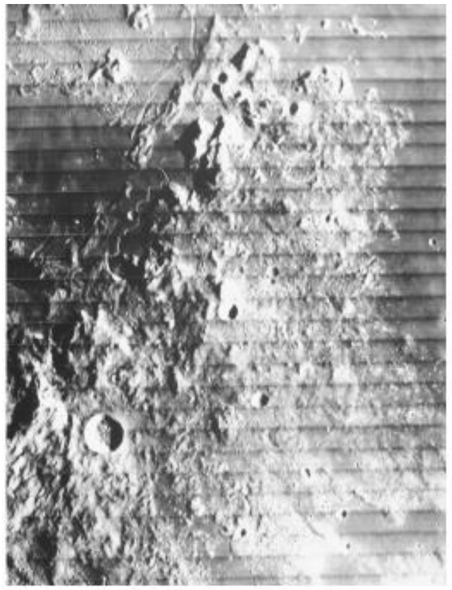


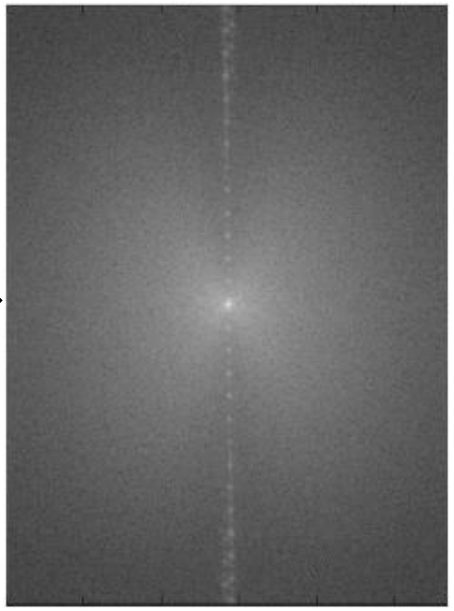
Remove peaks



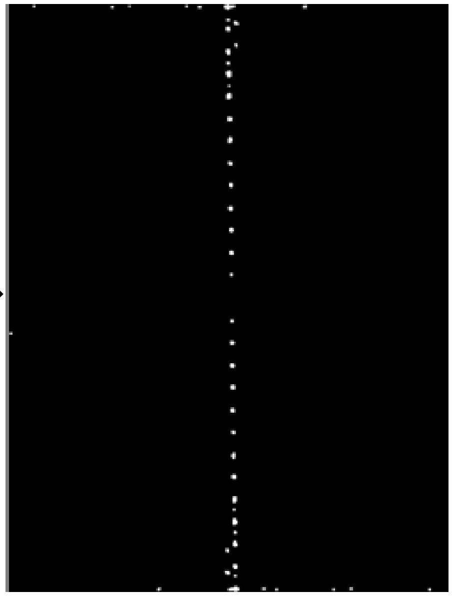
Periodic bg removed

Image enhancement: Lunar orbital image





$|F(u,v)|$



Remove peaks

Lines removed

Slide Credit: A. Zisserman

More advanced applications

Deep image processing

So far, we assumed that we have the

- ideal filters to enhance images (to blur, sharpen, resize)
- ideal models and priors (to super-resolve, remove noise)

What if we had a “learning machine” that can learn to map an input to output?

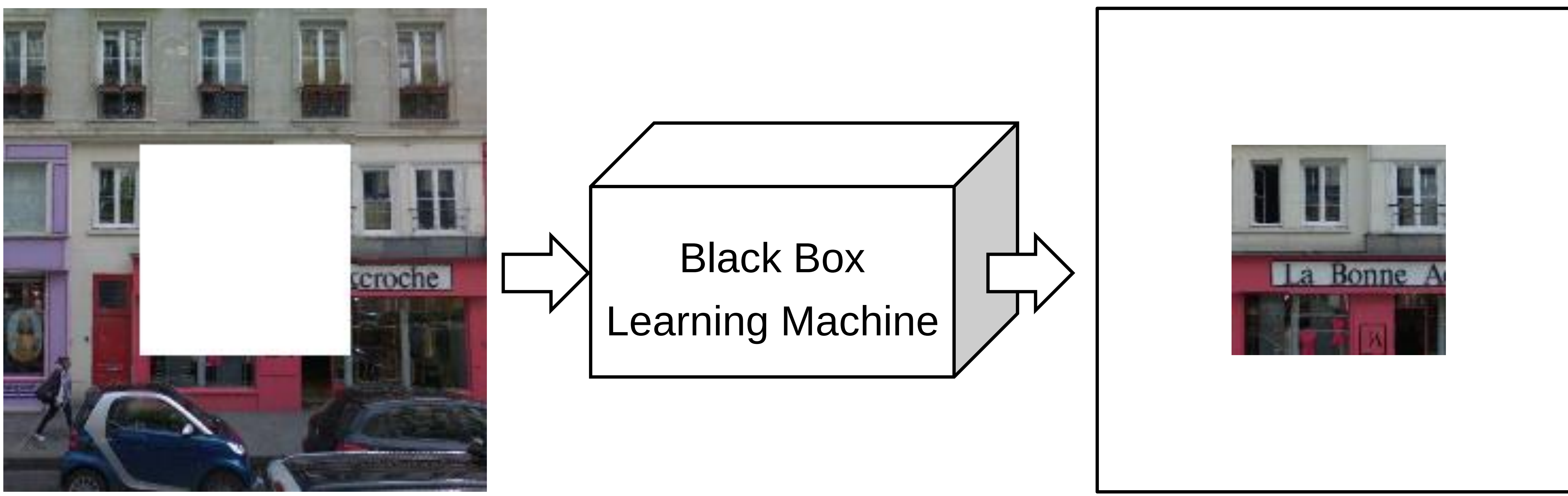


Image Credit: Pathak et al.

Perceptron [Rosenblatt 1957]

$$f(x) = \sum_{i=1}^{n} w_i x_n + b\,, \qquad y = \begin{cases} 1 & \text{if } f(x) > 0 \\ 0 & \text{otherwise} \end{cases}$$

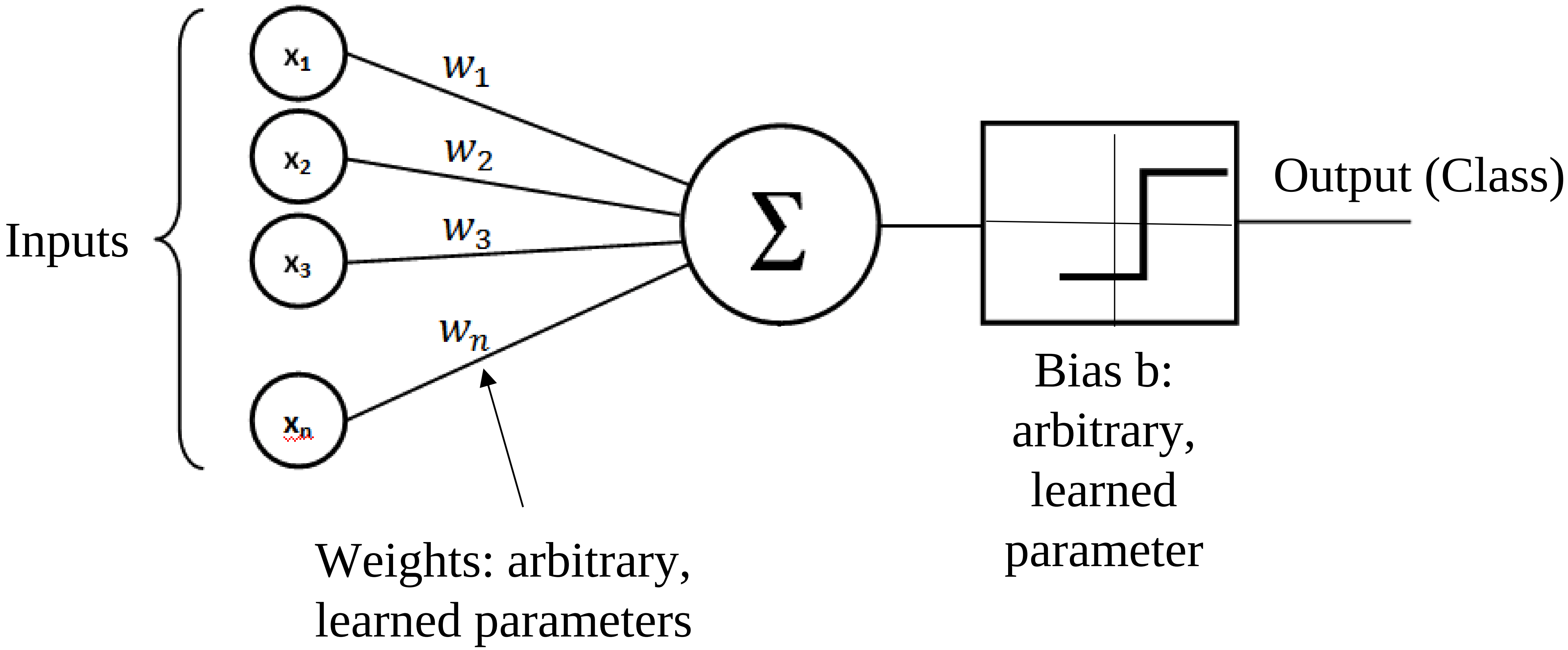


Image Credit: C. Barnes

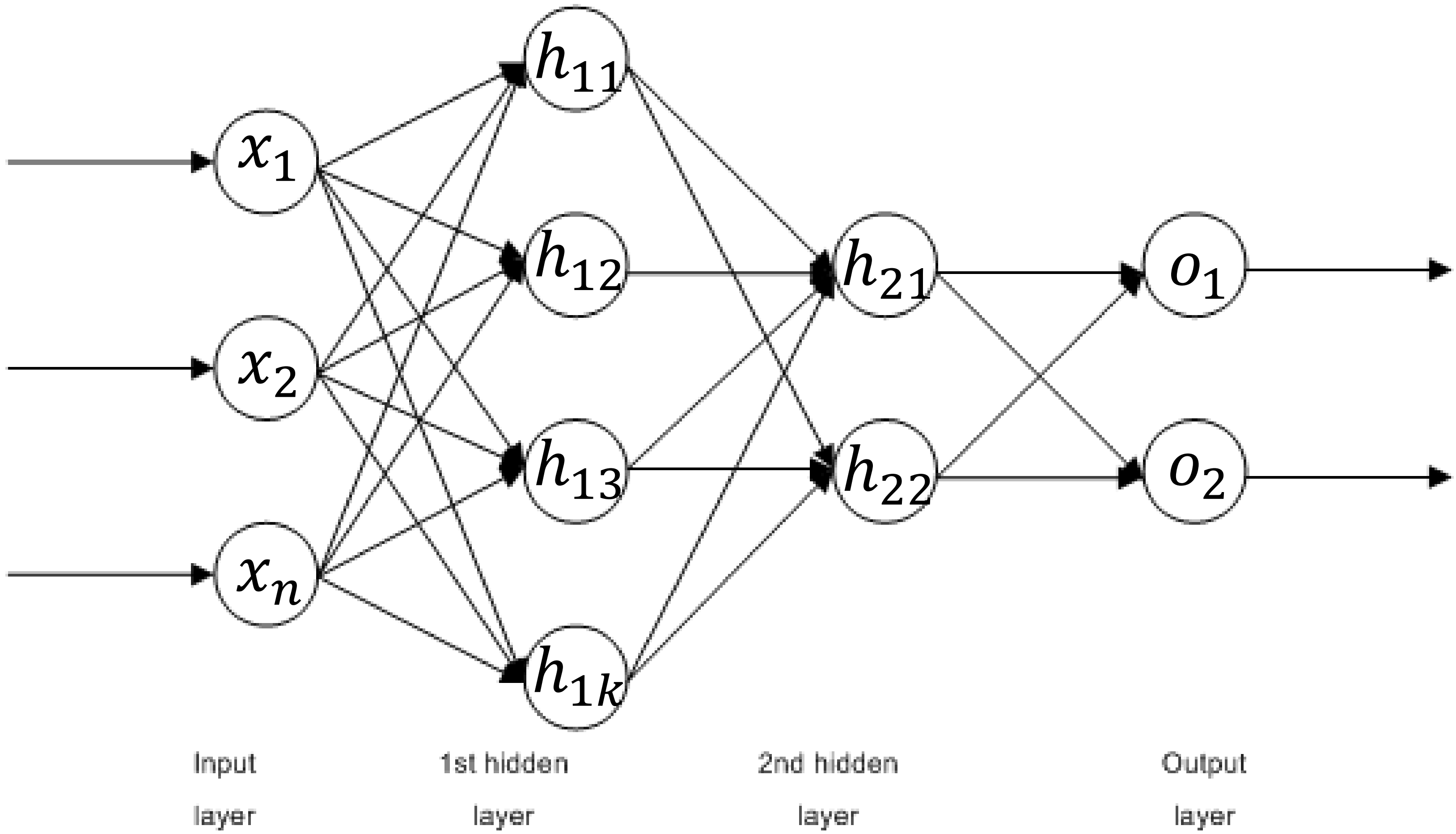
MultiLayer Perceptron (MLP)

Universal Approximation Theorem: a single hidden layer and linear output layer can approximate any continuous function.

$$H_0 = [h_{01}, h_{02}, \dots, h_{0n}] = [x_1, x_2, \dots, x_n]$$

$$H_1 = [h_{1.}] = [\sigma(w_{1.} \cdot h_{0.})]$$

$$O = \sigma(W_2 \cdot \sigma(W_1 \cdot H_0))$$



Activation functions (σ)

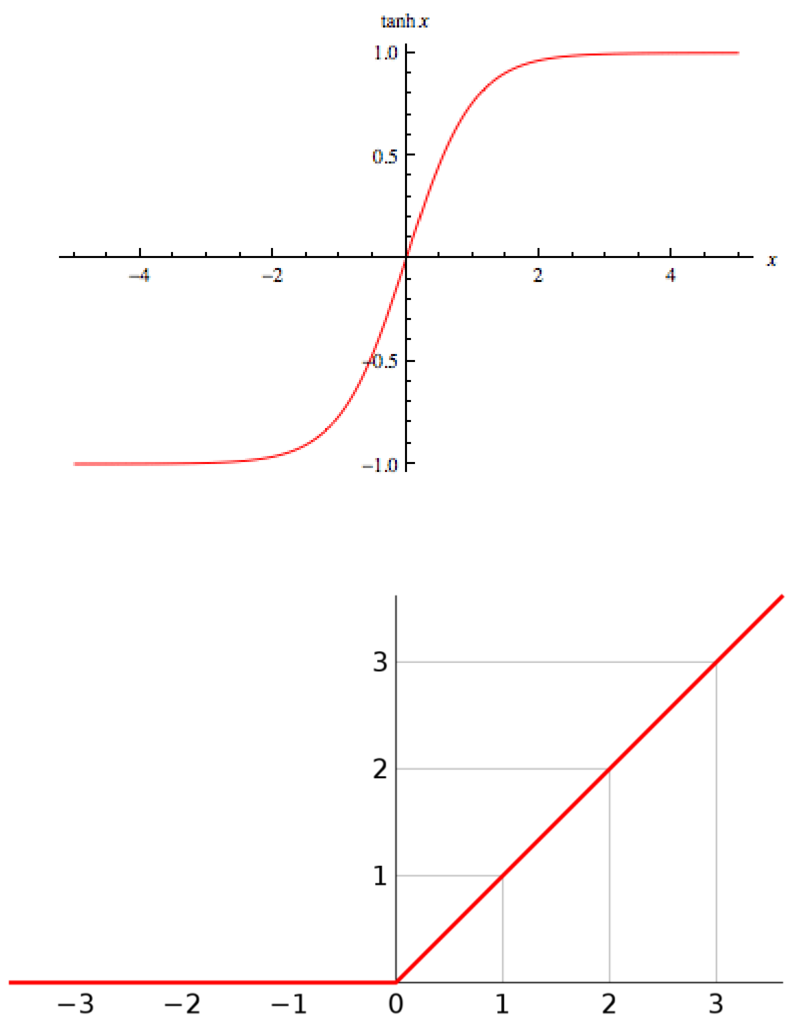


Image Credit: Wikipedia

Convolutional Neural Networks [Lecun et al 1998]

Activation function

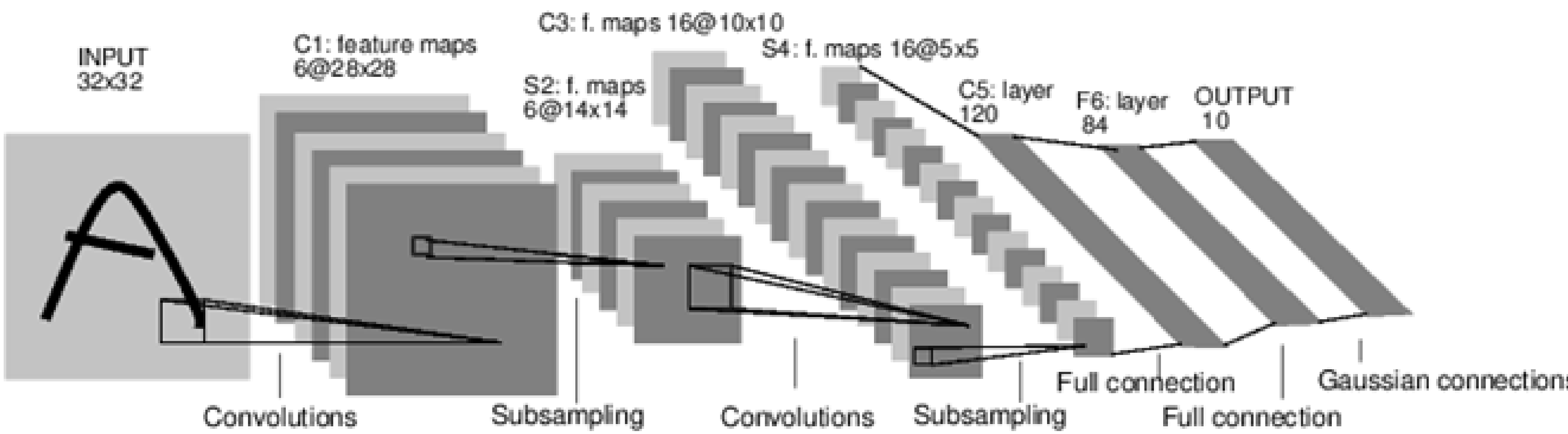
Layer (l) convolution filters

$$H_{l+1} = \sigma(H_l * [W_{l1}, W_{l2}, \dots, W_{lk}] + [b_{l1}, b_{l2}, \dots, b_{lk}])$$

Layer (l+1) feature map

Layer (l) feature map

Biases



http://www.deeplearningbook.org/

Deep learning guide

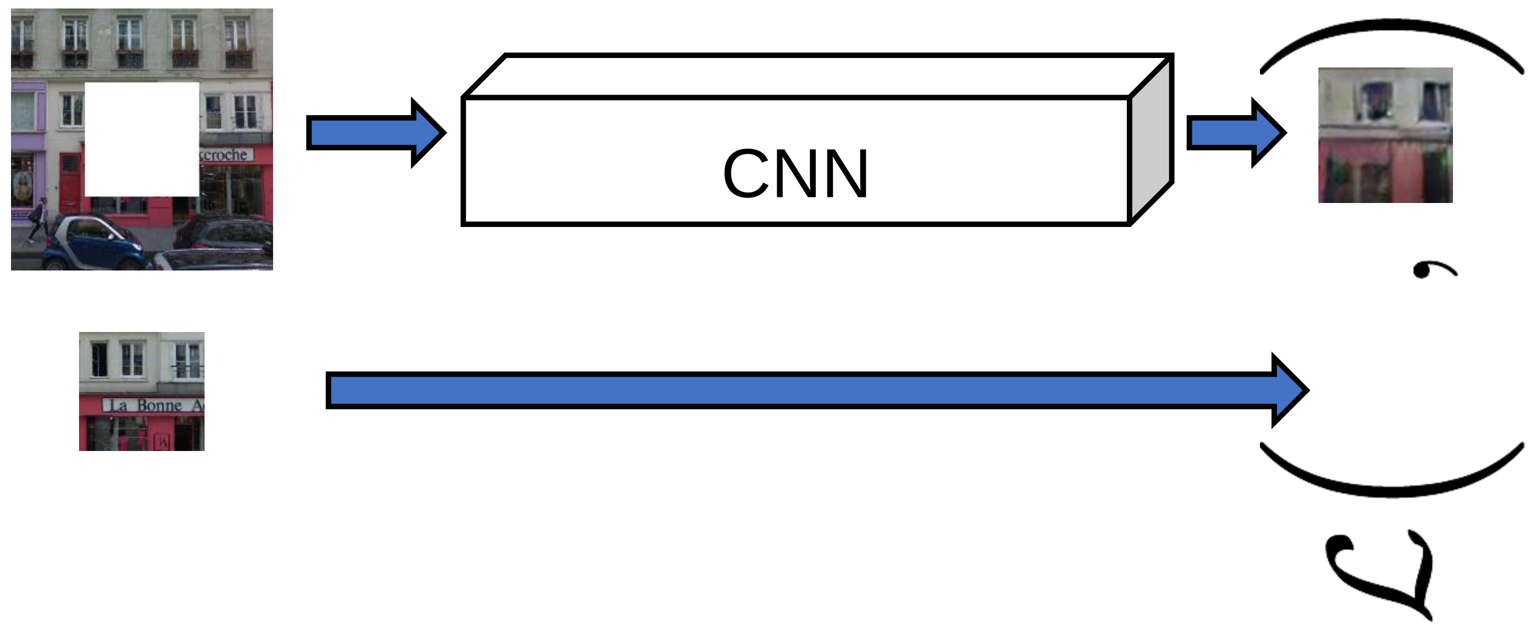
When to use it?

- Do you have plenty of input data (images, videos, text, audio)?
- Do you have supervisory signal for each input point (image and its label “cat”)?
- Do you have enough computation resources (this means expensive graphics cards)? Or are you working for Google, Facebook ☺ ?
- Can you specify a good compatibility function (loss) between output and desired output?

Back to applications

Image Inpainting (Pathak et al. 2016)

Goal: Generate the contents of an arbitrary image region conditioned on its surroundings

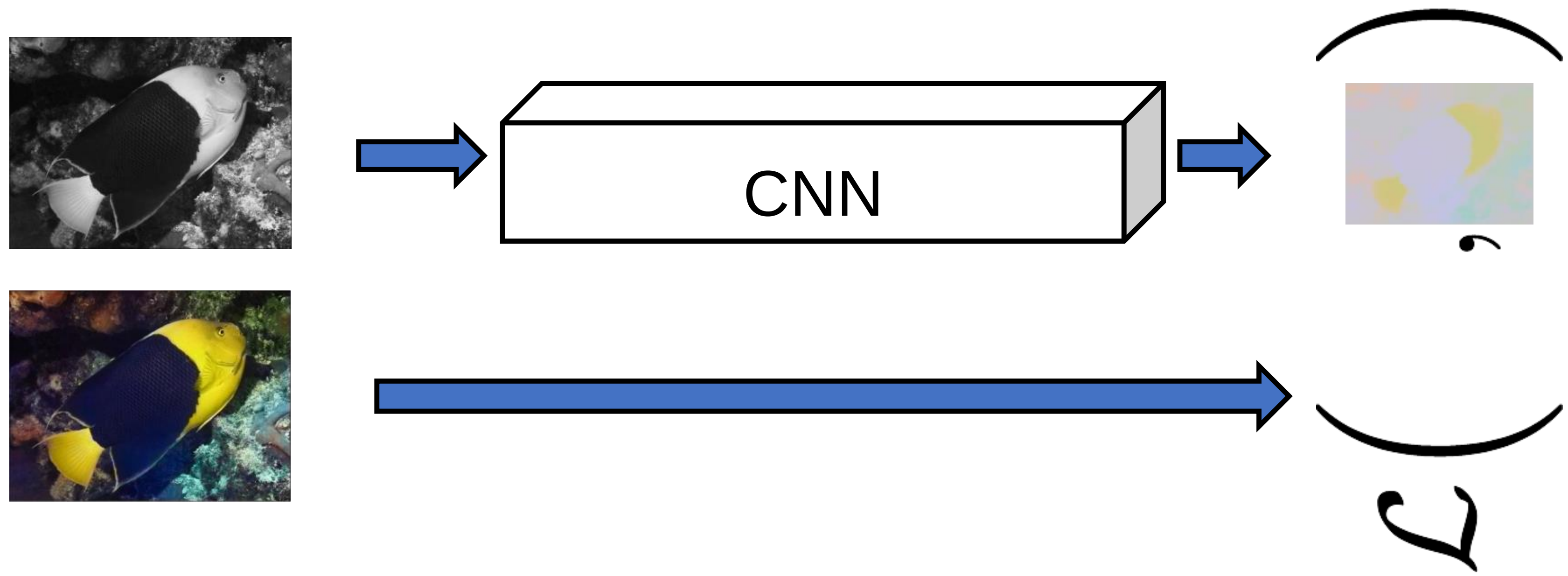


Supervision is free and infinite!

More applications

Image colorization (Zhang et al 2016)

Goal: Generate realistic colours for a given a grayscale image



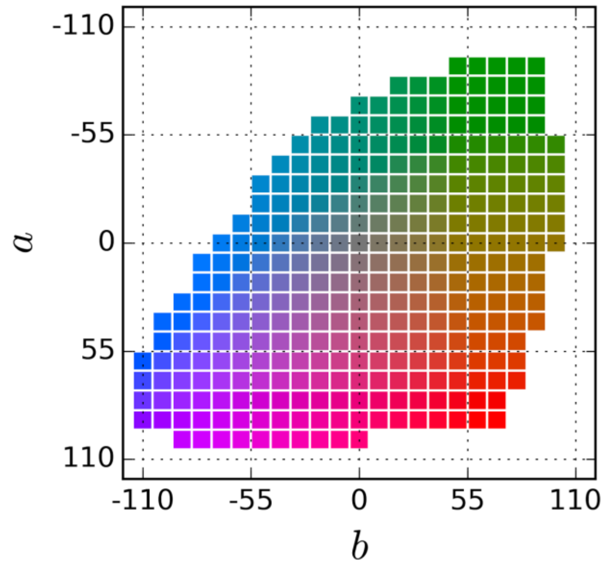
Grayscale | Output | Ground-truth

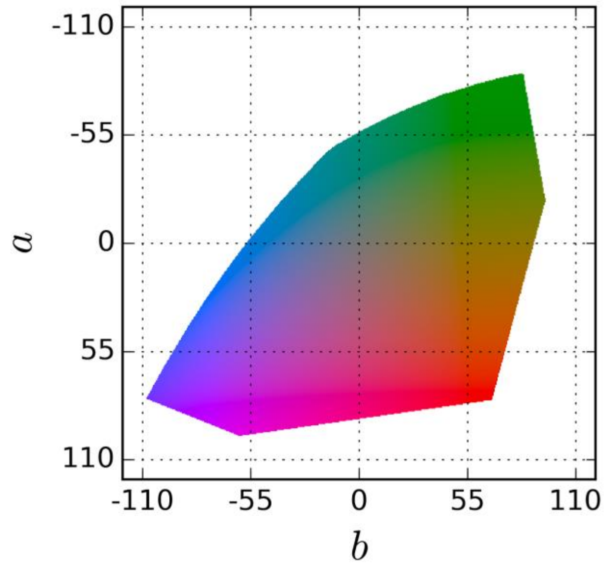
Applications

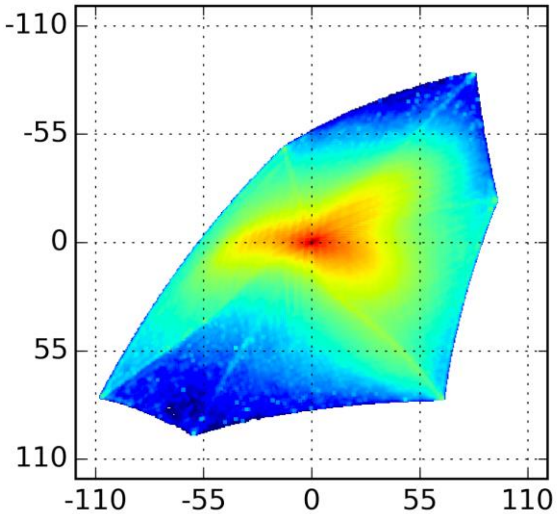
Image colorization (Zhang et al 2016)

Selection of loss function

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







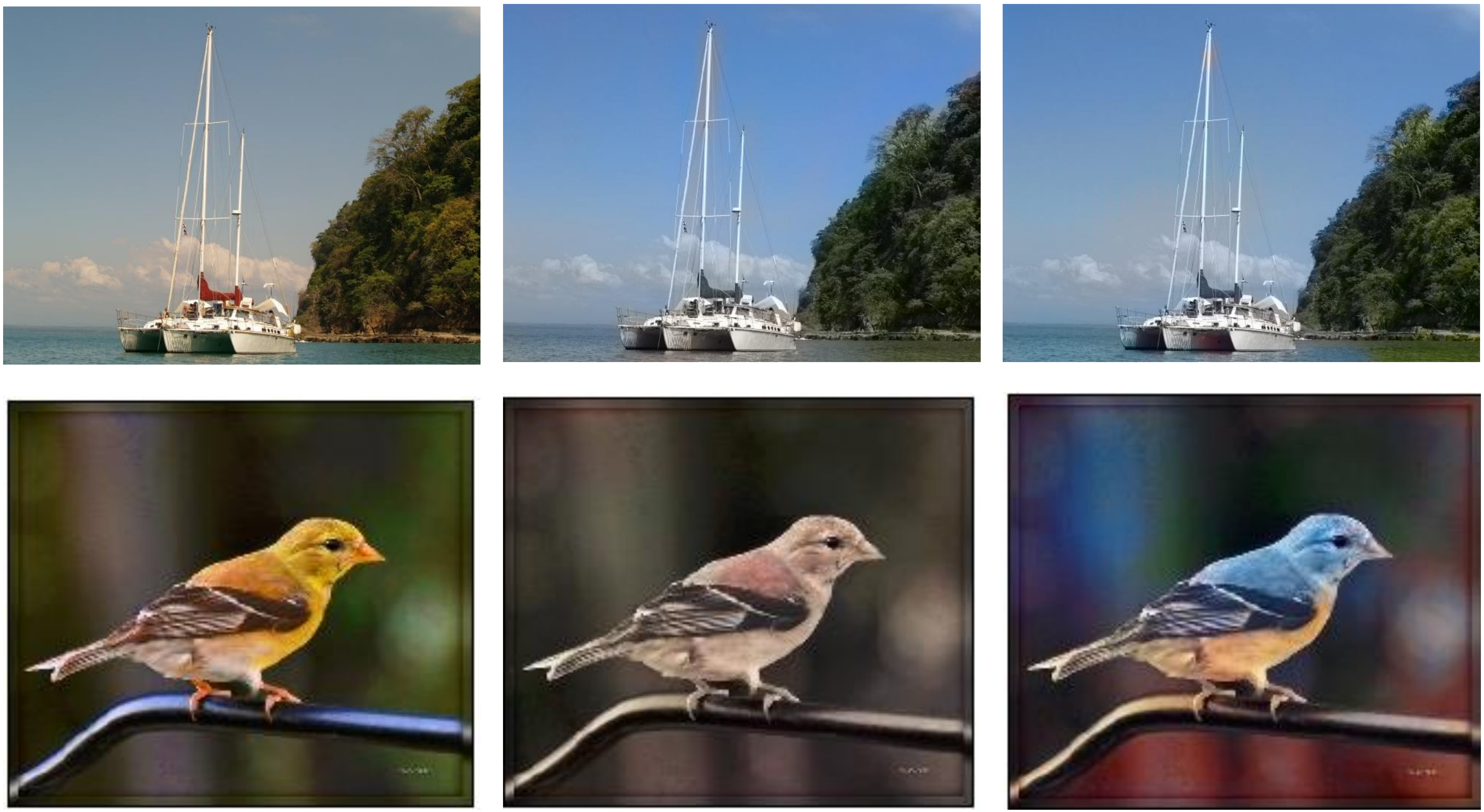
$$L_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

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

Applications

Image colorization (Zhang et al 2016)

Ground Truth | L2 Regression | Class w/ Rebalancing



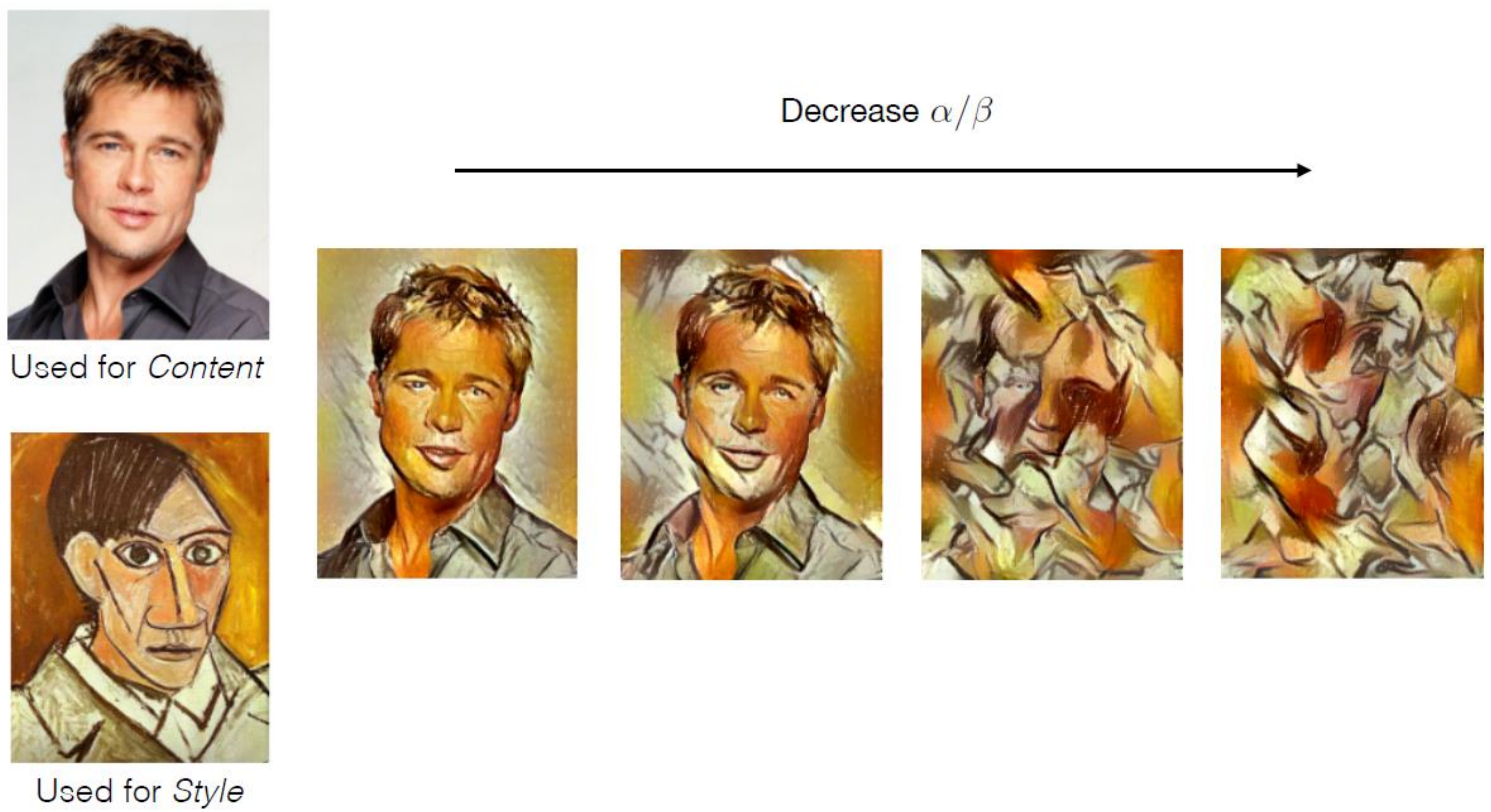
Applications

Style Transfer (Gatys et al 2015)

Goal: Create artistic images of high perceptual quality

Method: Match **style** and **content** of source and target images

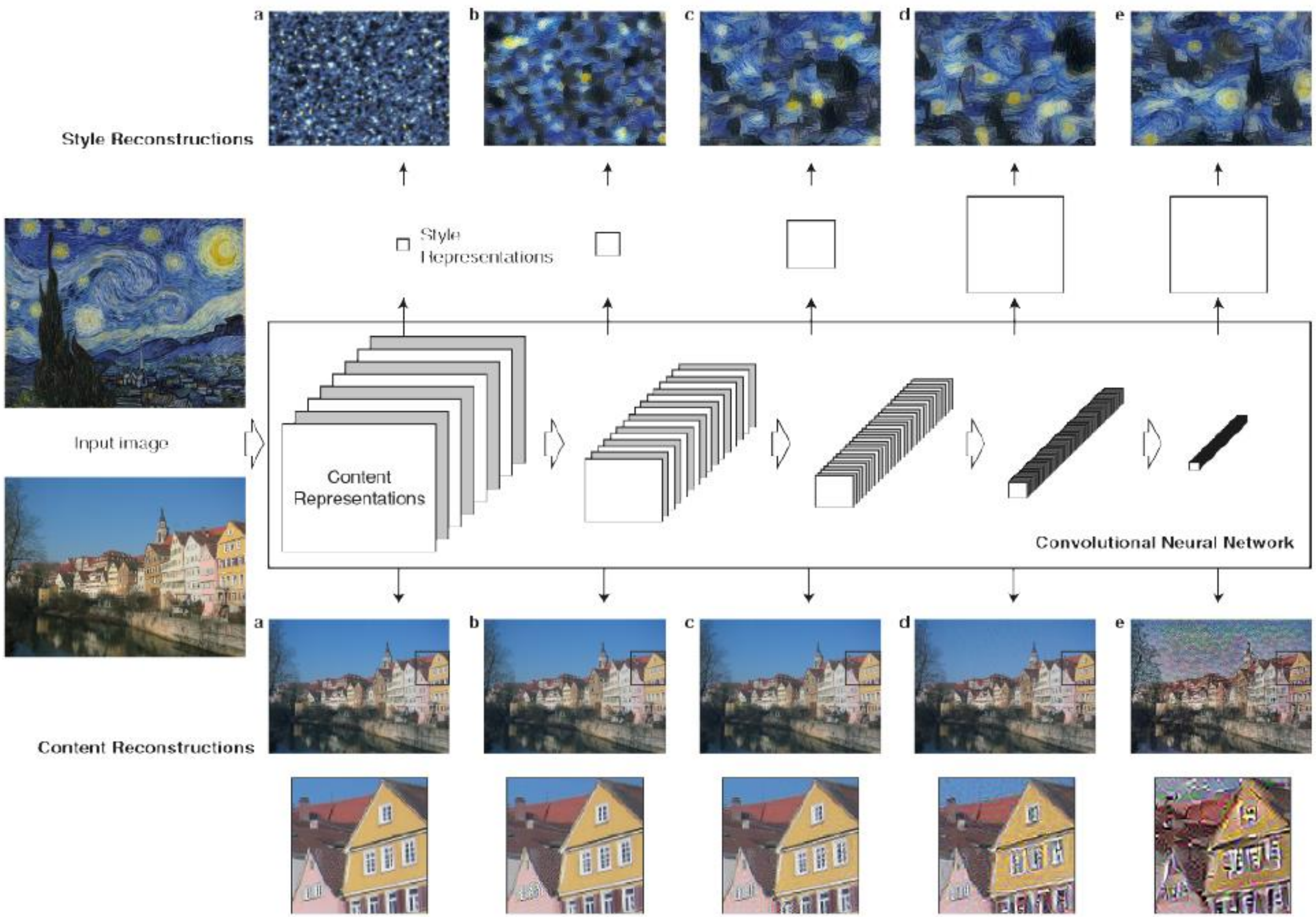
- **Style representation**: correlations between the different filter responses over the spatial extent of feature maps
- **Content representation**: different filter responses over the spatial extent of feature maps
- $Loss = \alpha Loss_{content} + \beta Loss_{style}$



Applications

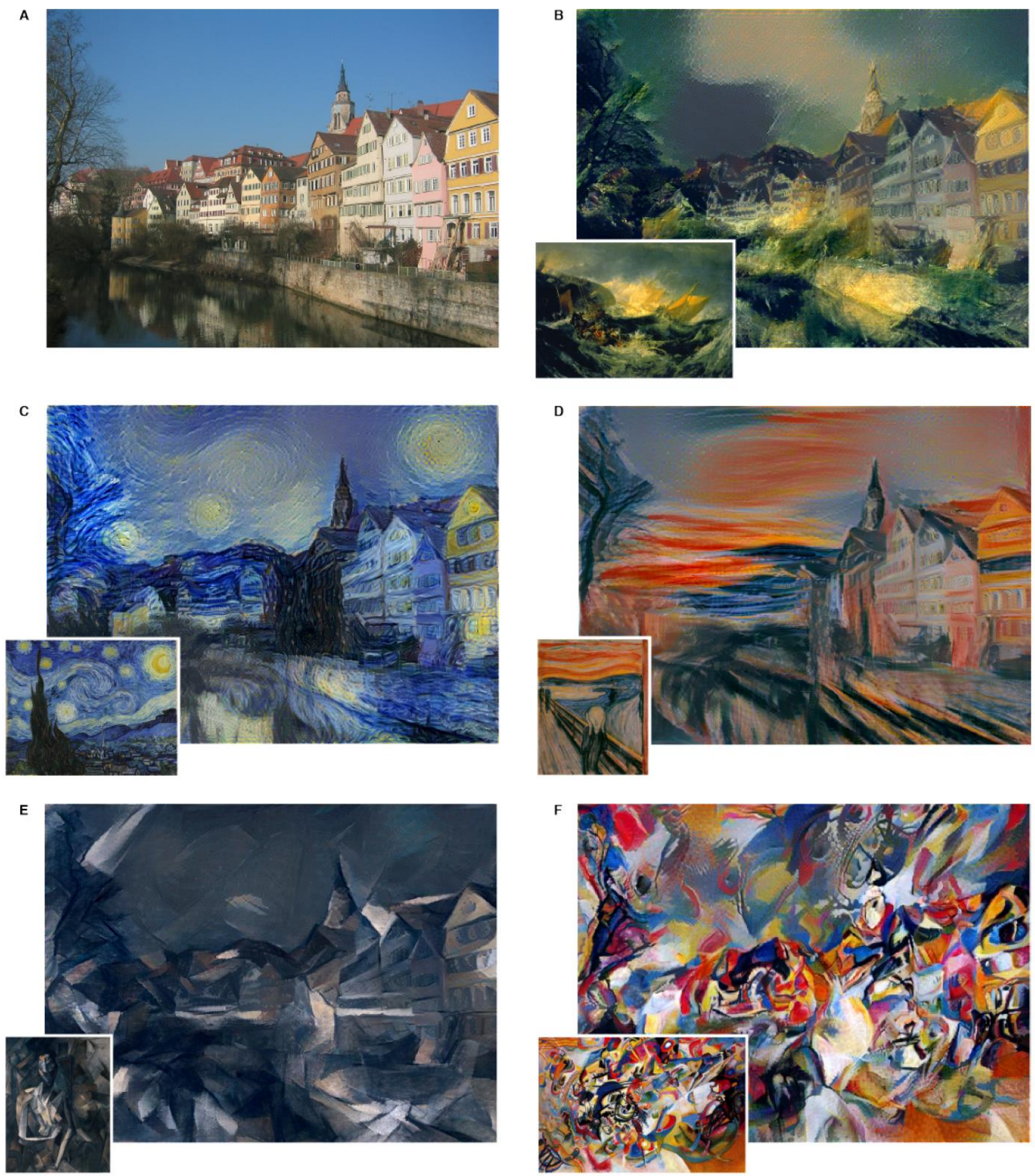
Style Transfer (Gatys et al 2015)

Goal: Create artistic images of high perceptual quality



Applications

Style Transfer (Gatys et al 2015)

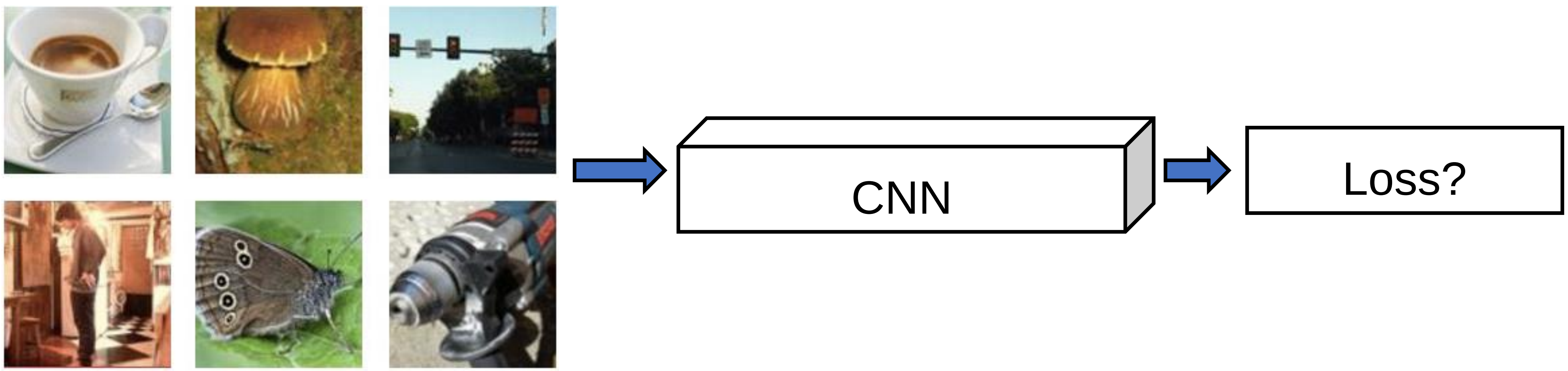


Applications

Generative Adversarial Networks (GANs) (Goodfellow et al 2014)

We can colorize, inpaint and stylize images by using neural networks.

How about generating new images?



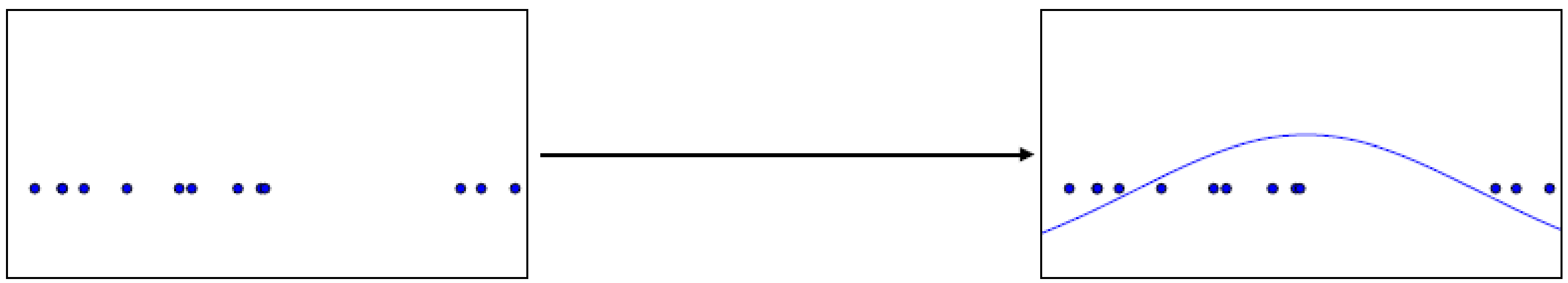
Can we use L2 loss?

What kind of loss can we use?

Applications

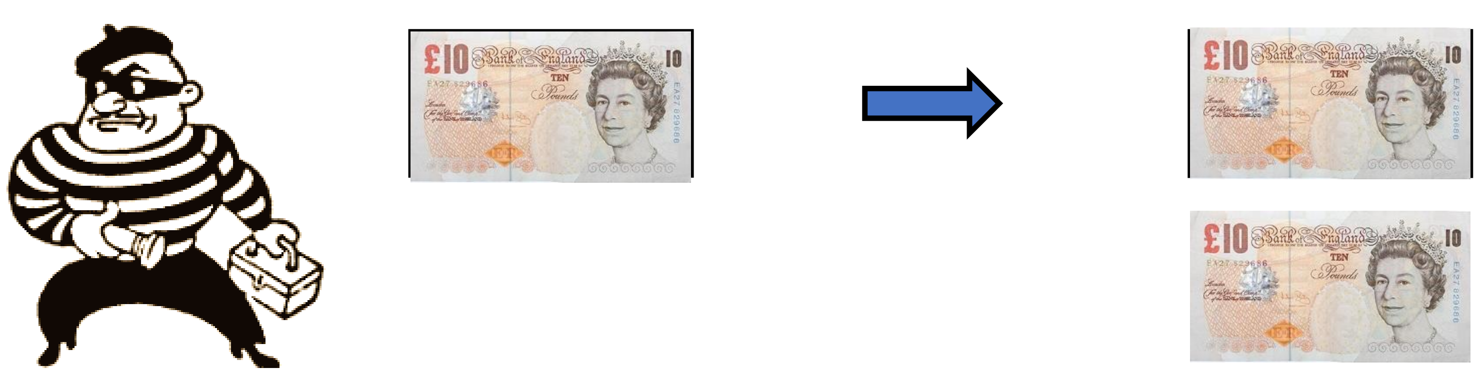
Generative Adversarial Networks (Goodfellow et al 2014)

Density estimation: We want outputs of our model (parametrized by $\boldsymbol{\theta}$) to be similar to distribution of input data



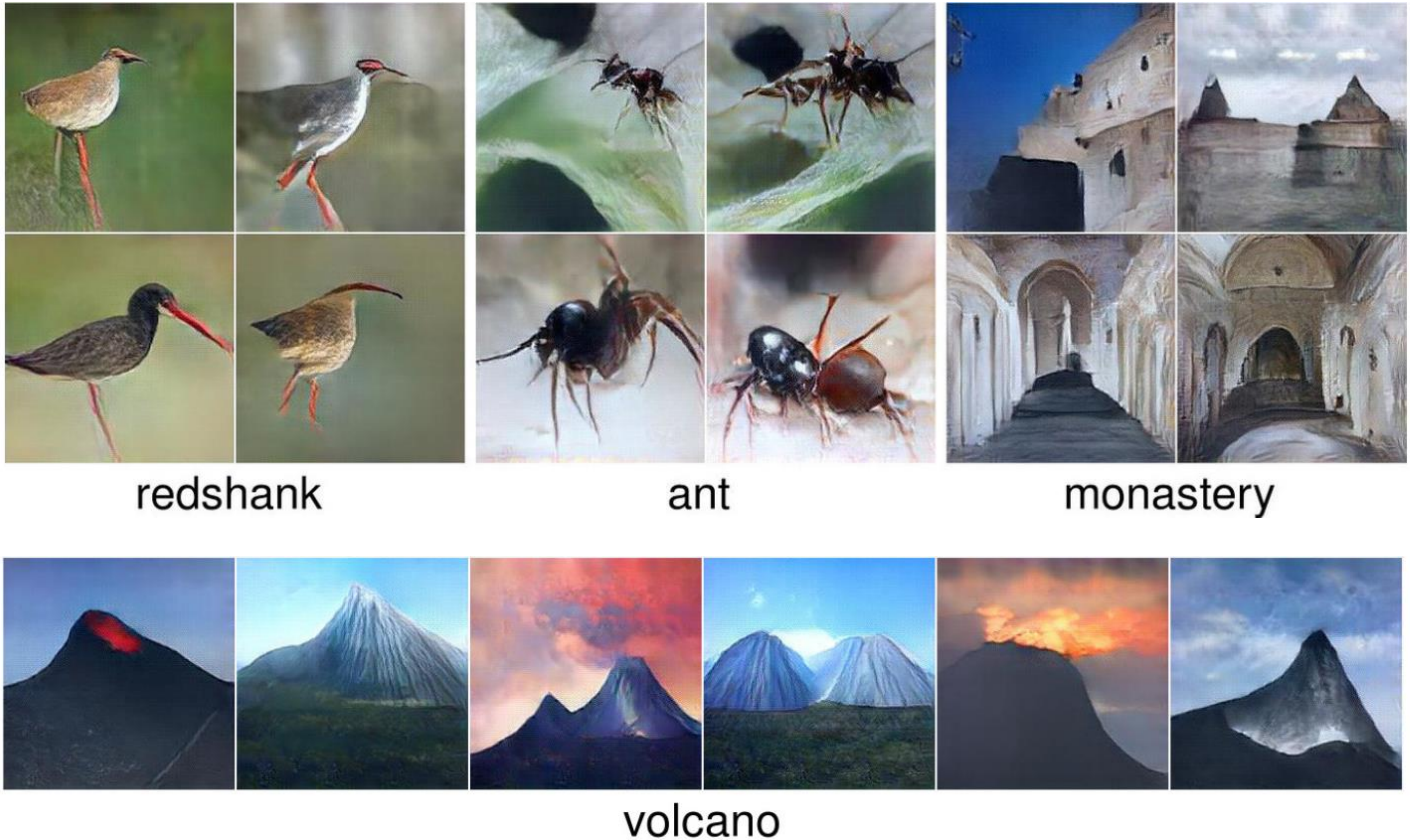
We have a pair of networks, Generator (G) and Discriminator (D)

- They fight against each other
- G goal: make its probability distribution similar to training set and make prediction that G cannot distinguish
- D goal: distinguish between G's prediction and real data



Applications

PPGAN (Nguyen et al 2016)



Applications

GAN

Bad examples:

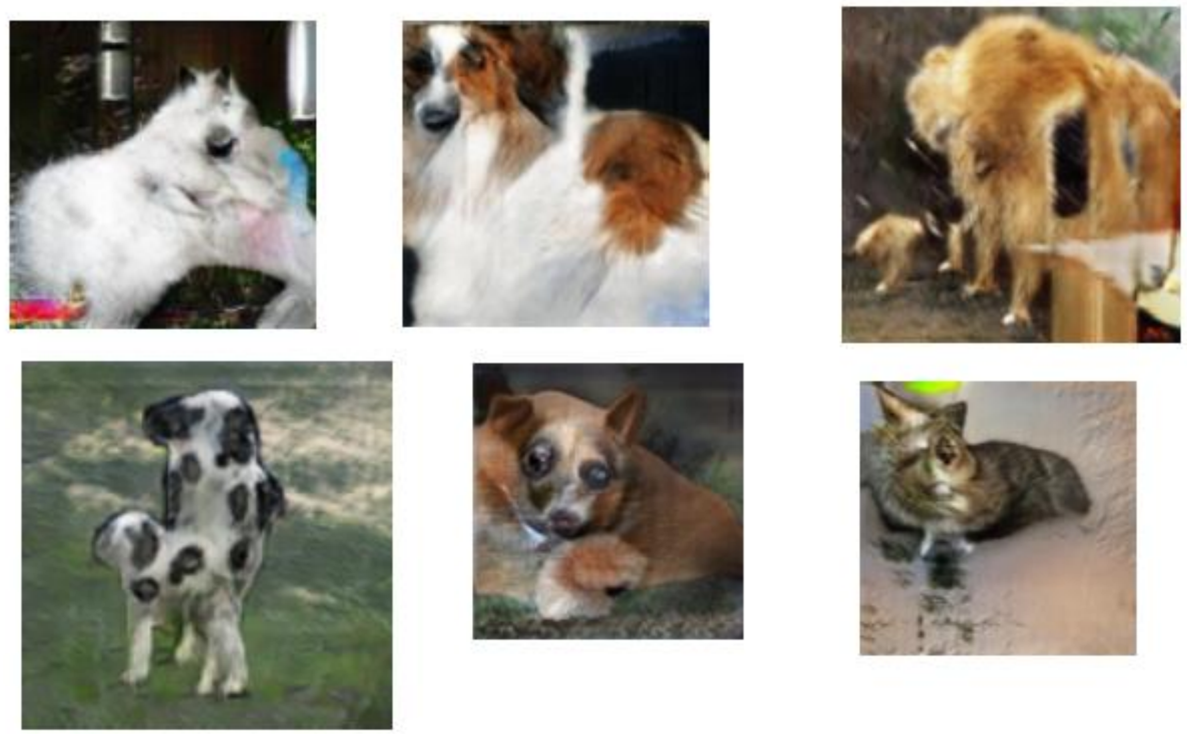


Figure 31: GANs on 128 × 128 ImageNet seem to have trouble coordinating global structure, for example, drawing "Fallout Cow," an animal that has both quadrupedal and bipedal structure.

Neural Photo Editing with Introspective Adversarial Networks

https://www.youtube.com/watch?v=FDELBFSeqQs

Notes

- B1: Chapter 9.4 Fourier Transform
- Additional reading: http://www.deeplearningbook.org/ Part II, Chapter 6