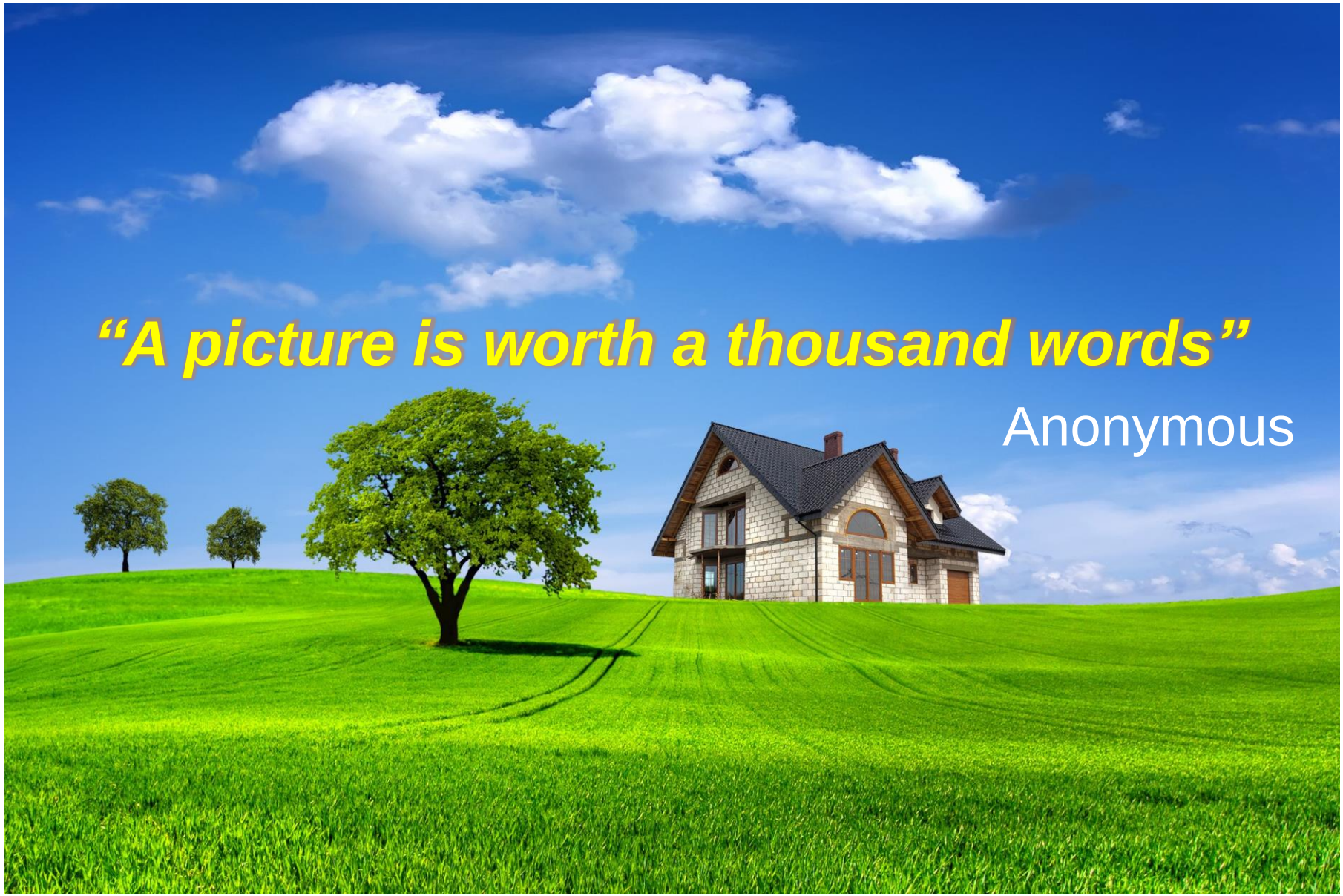
Feature Engineering

Nguyen Ngoc Thao
nnthao@fit.hcmus.edu.vn

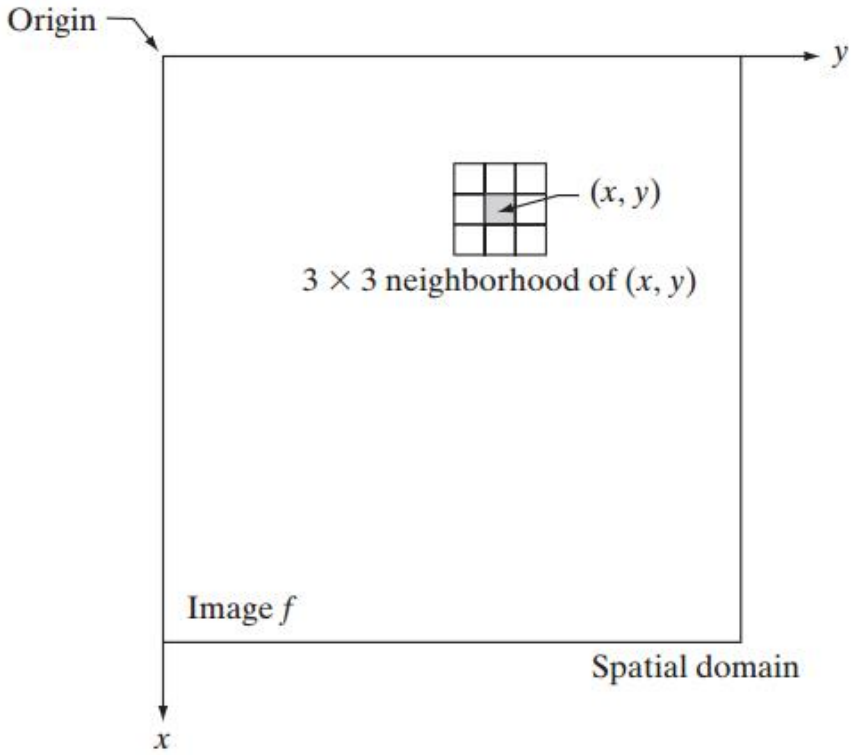
Content outline

- Digital images and Digital image processing
- Handcrafted features
- Automatic features
- Common metrics in vision tasks


“A picture is worth a thousand words”
Anonymous

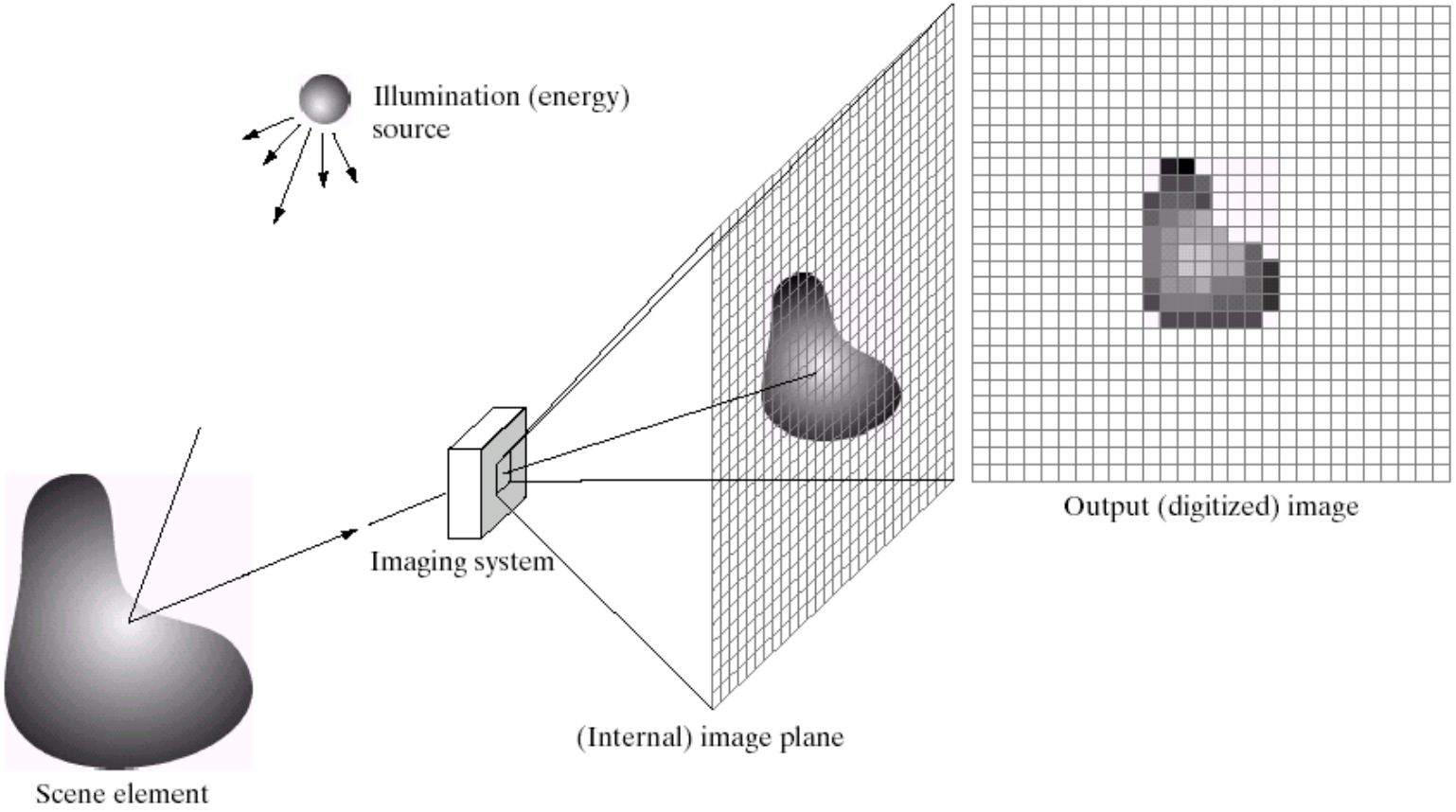
What is an image?

- An image is defined as a two-dimensional function, $\boldsymbol{f(x, y)}$.
  - where x and y are spatial (plane) coordinates
- The amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point.



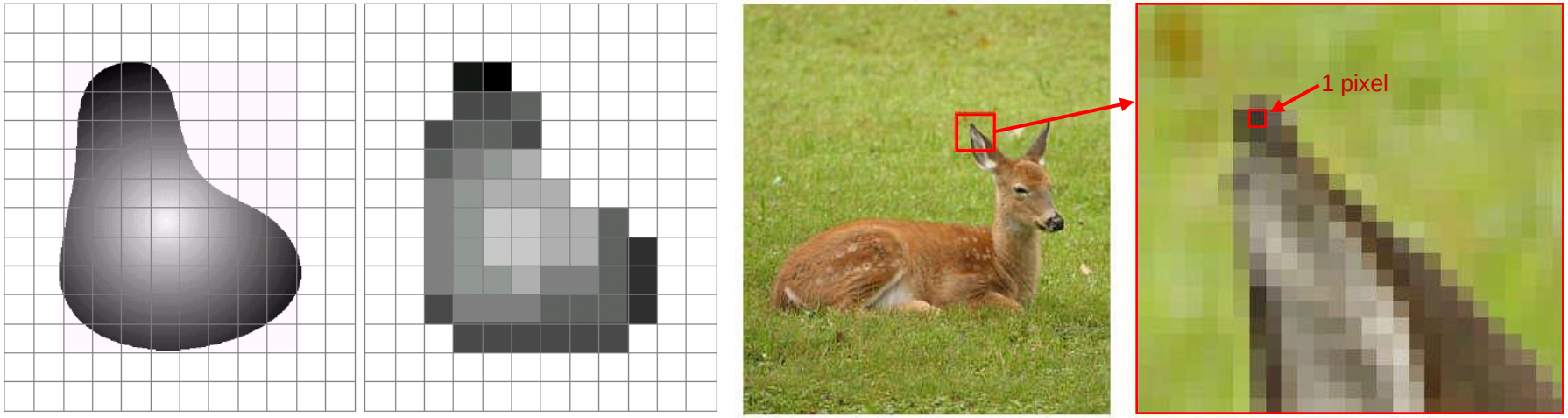
What is a digital image?

- A digital image represents a two-dimensional image as a finite set of digital values, called picture elements or pixels.



What are pixels?

- Pixel values typically represent gray levels, colours, heights, opacities, etc.
- Digitization implies that a digital image is an approximation of a real scene.



Common image formats

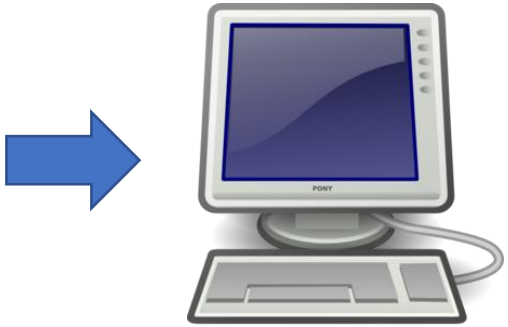
- 1 sample per point (B&W or Grayscale)
- 3 samples per point (Red, Green, and Blue)
- 4 samples per point (Red, Green, Blue, and Opacity)



Digital image processing (DIP)

- Digital image processing refers to the processing of digital images by means of a digital computer.

Original image



Contrast-enhanced image

Digital image processing: Common tasks

- Improve the pictorial information for human interpretation

Image credits: 1

Main tasks in DIP

- Process the image data for storage, transmission and representation for autonomous machine perception

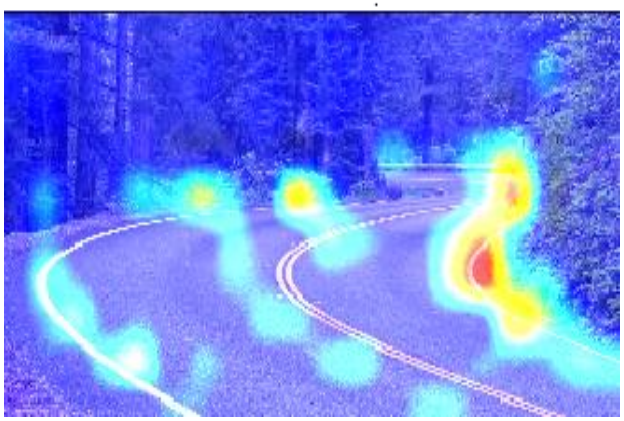
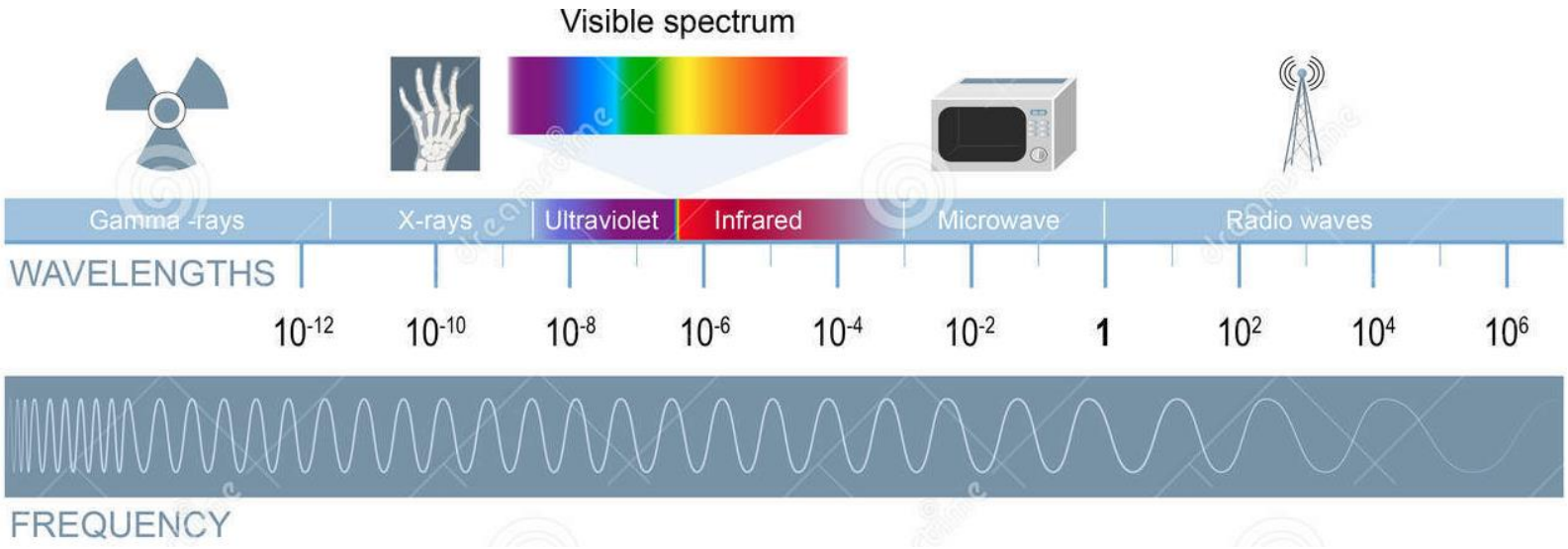




Image credits: 2 and 3

The electromagnetic spectrum

- DIP covers almost the entire electromagnetic spectrum.



- It encompasses a wide and varied field of applications.

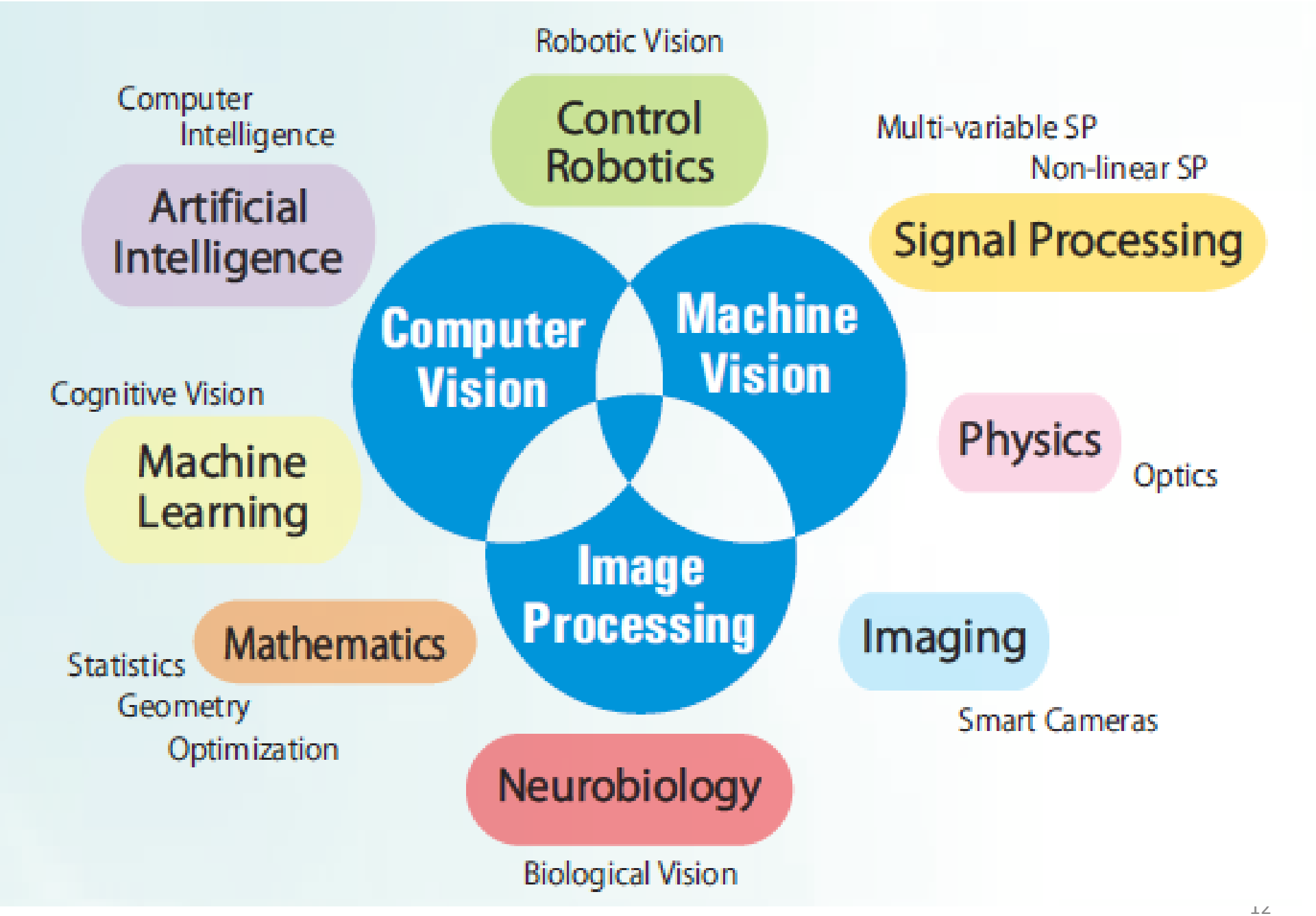
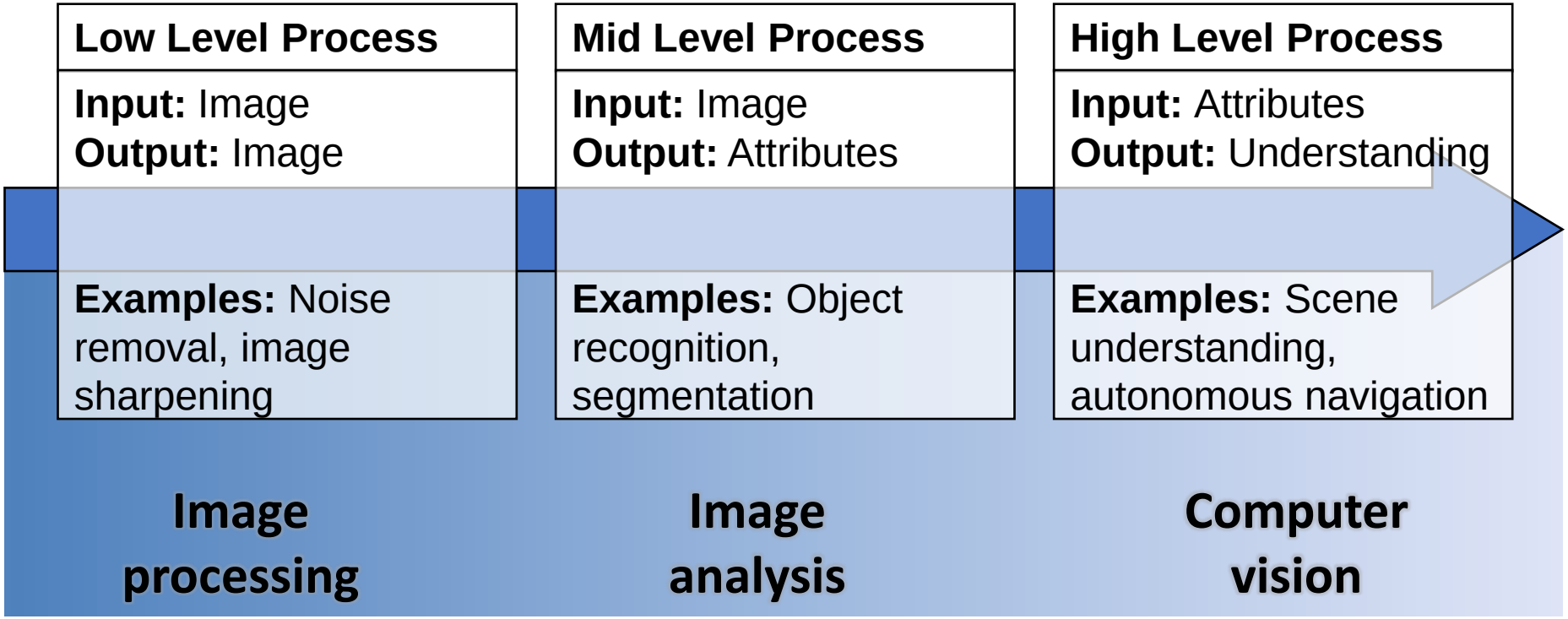

Robotic Vision
Control
Robotics
Computer
Intelligence
Artificial
Intelligence
Multi-variable SP
Non-linear SP
Signal Processing
Computer
Vision
Machine
Vision
Cognitive Vision
Machine
Learning
Physics
Optics
Image
Processing
Mathematics
Statistics
Geometry
Optimization
Imaging
Smart Cameras
Neurobiology
Biological Vision

Image processing to computer vision

- The continuum from image processing to computer vision can be broken up into low-, mid- and high-level processes.



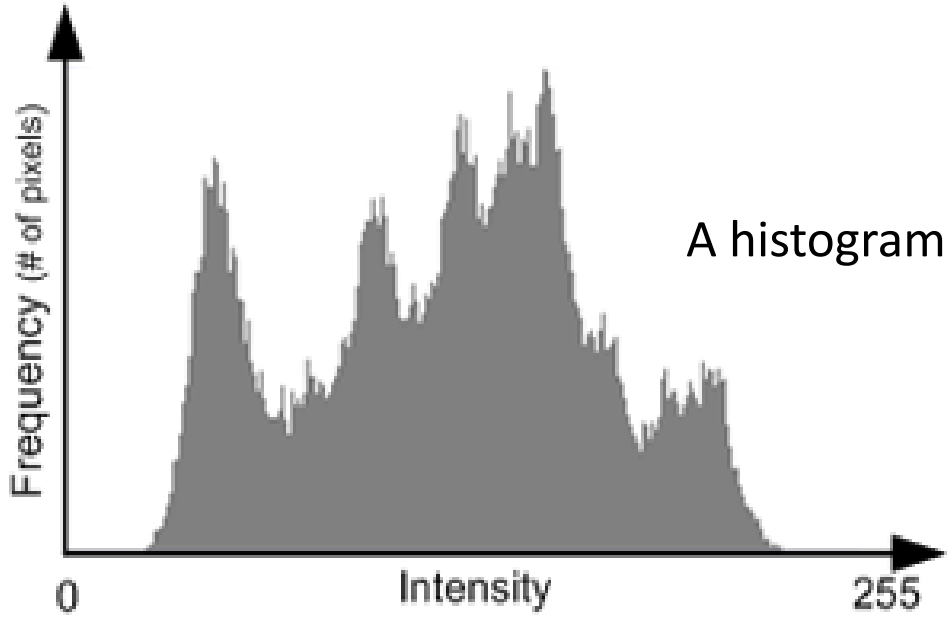
Handcrafted features

Histogram

- The histogram of a digital image with intensity levels in the range $[0, L-1]$ is a discrete function of the form

$$\boldsymbol{h(r_k) = n_k}$$

  - n_k is the number of pixels in the image with intensity value r_k.



A histogram of an 8-bit image

Normalized histogram $\boldsymbol{p(r_k)}$

- It is common practice to normalize a histogram using

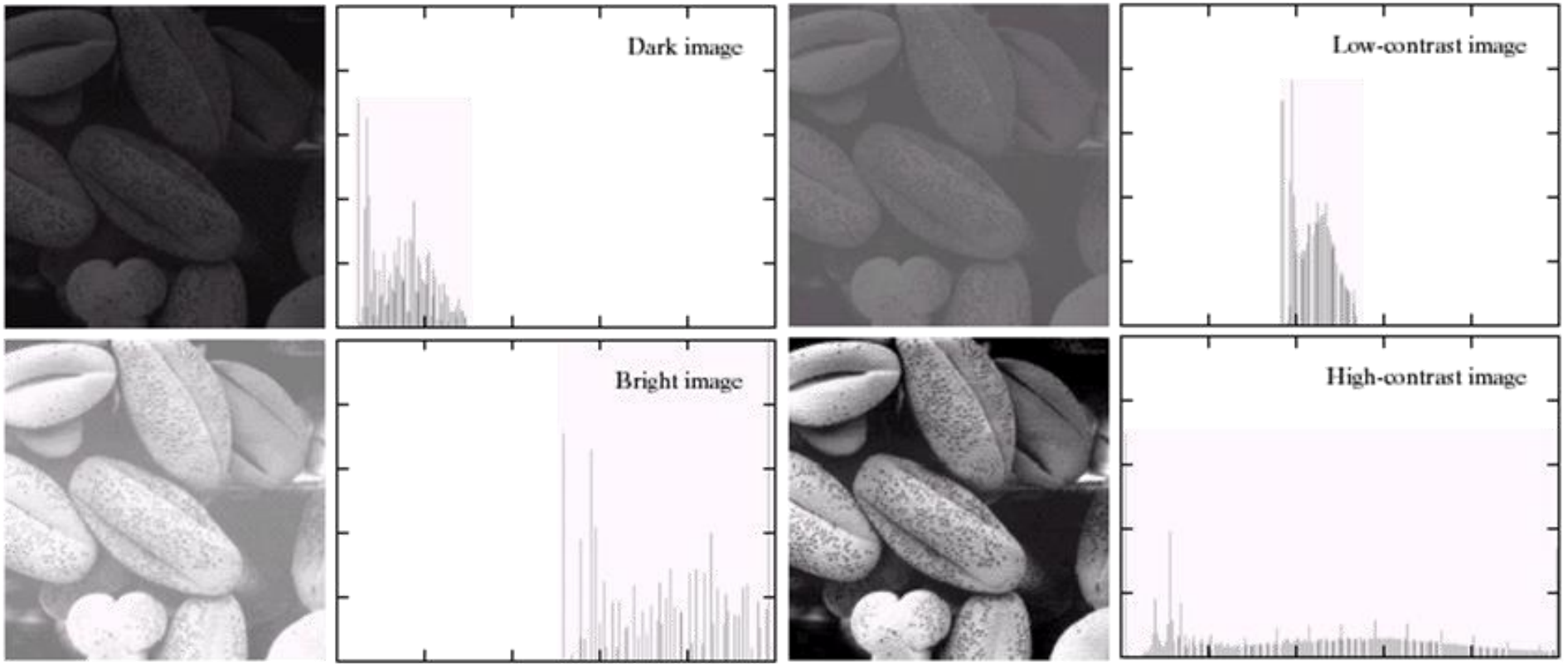
$$\boldsymbol{p(r_k) = \frac{n_k}{MN}}$$

  - M and N are the row and column dimensions of the image
  - $k = 0, 1, 2, \ldots, L-1$

- $p(r_k)$ estimates the probability of occurrence of the intensity level r_k in an image.
  - All the components of a normalized histogram sum to 1.

a b
c d

Four image types: dark, light, low contrast, high contrast, and their corresponding histograms



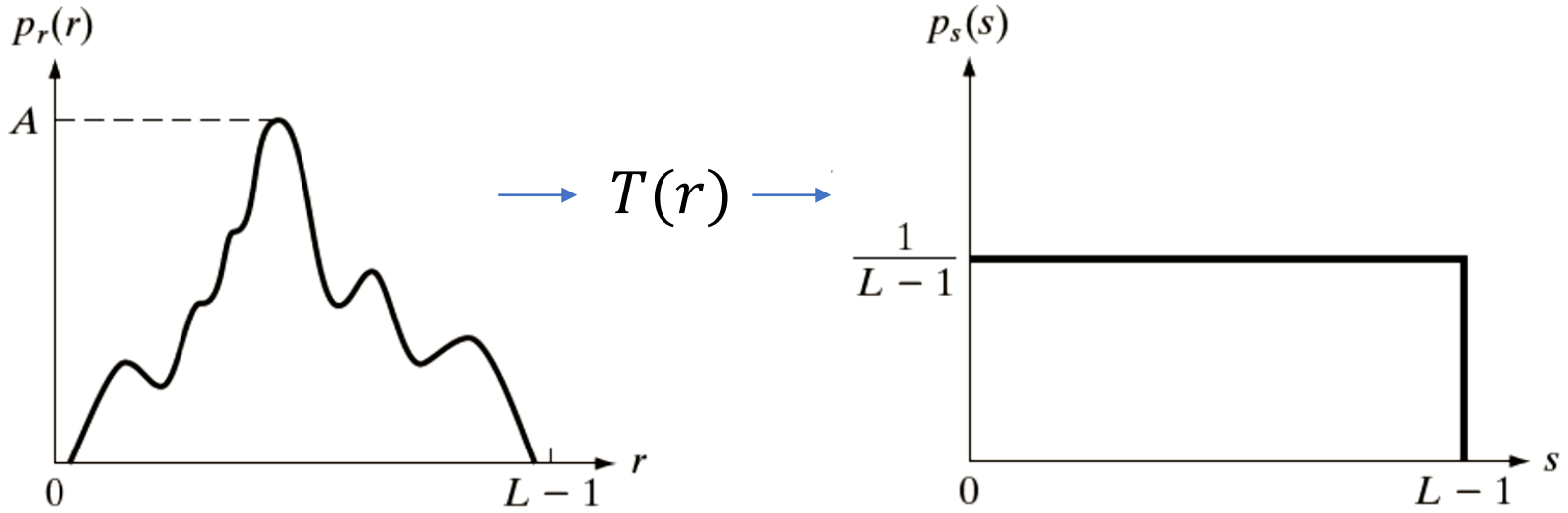
Histogram equalization

- Each pixel in the input image with intensity r_k is mapped into a corresponding pixel of level s_k in the output image.

$$\boldsymbol{s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)}$$

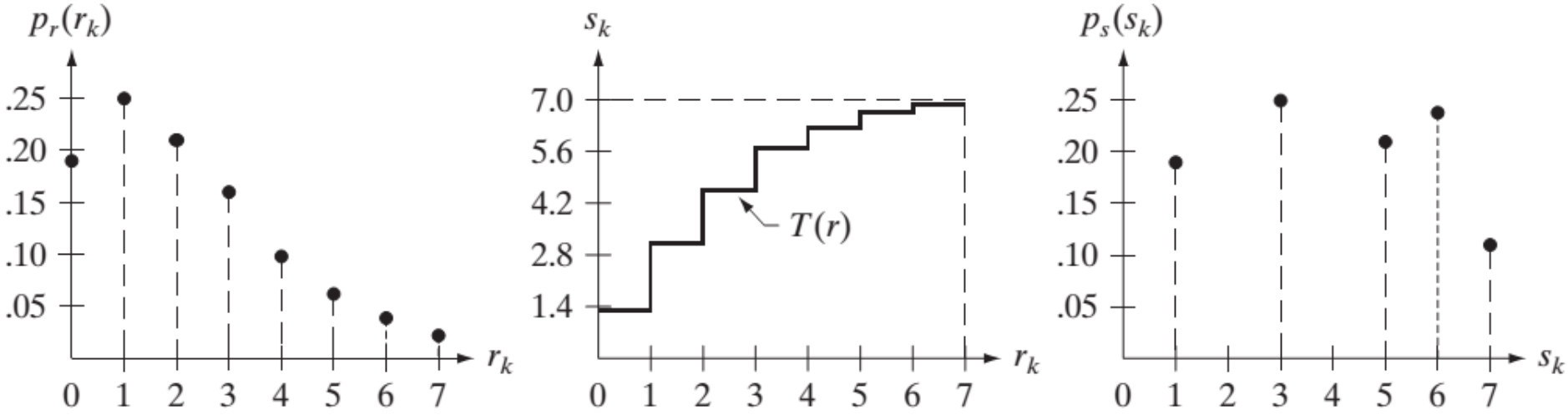
  - where $k = 0, 1, 2, \ldots, L-1$

- The transformation function T is determined **automatically** to produce an output image that has a uniform histogram.



a b

(a) An arbitrary PDF. (b) Result of applying histogram equalization to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

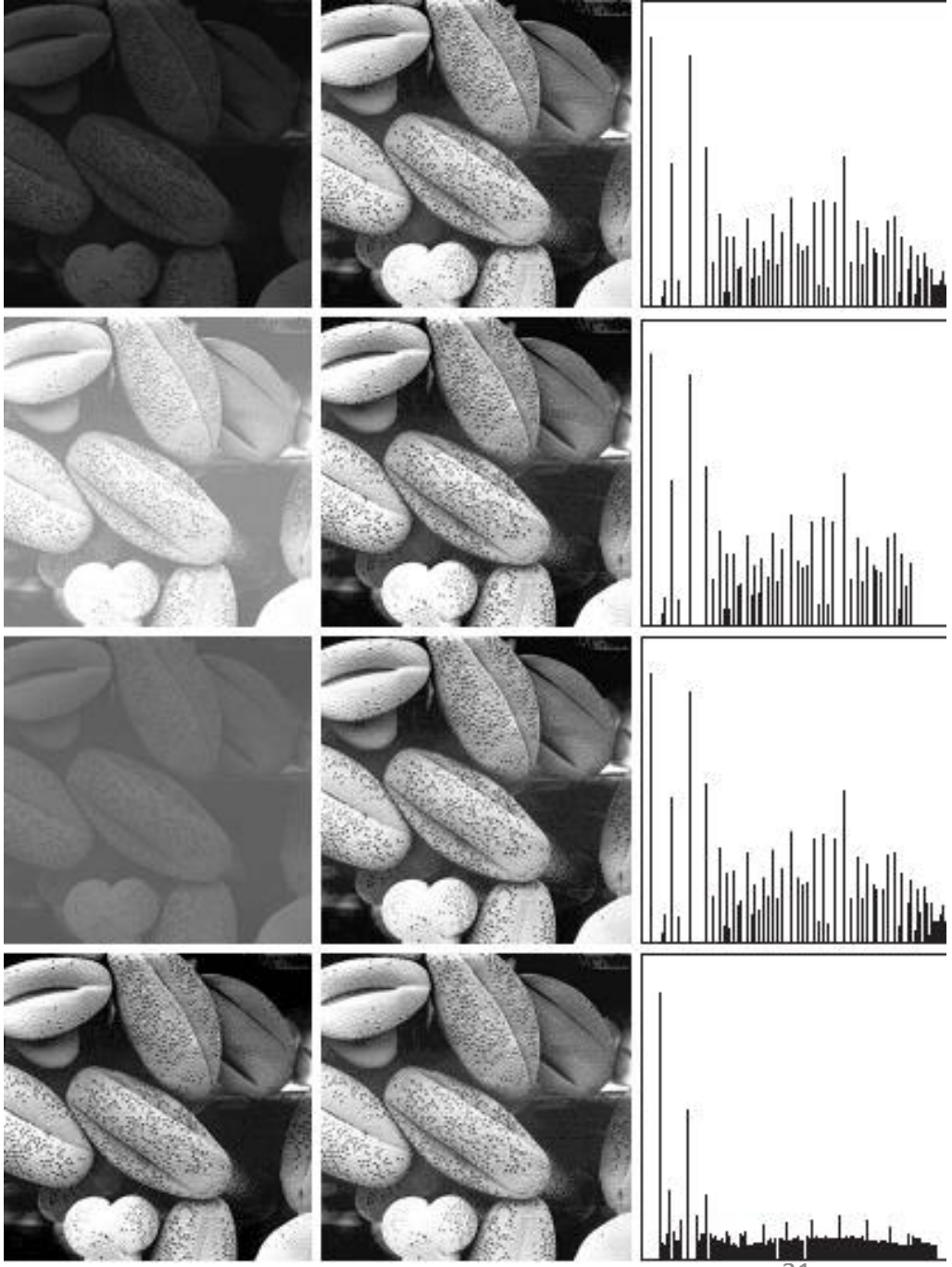


a b c Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

Note: *It cannot be proved (in general) that discrete histogram equalization results in a uniform histogram, yet it has a general tendency to spread the histogram of the input image*

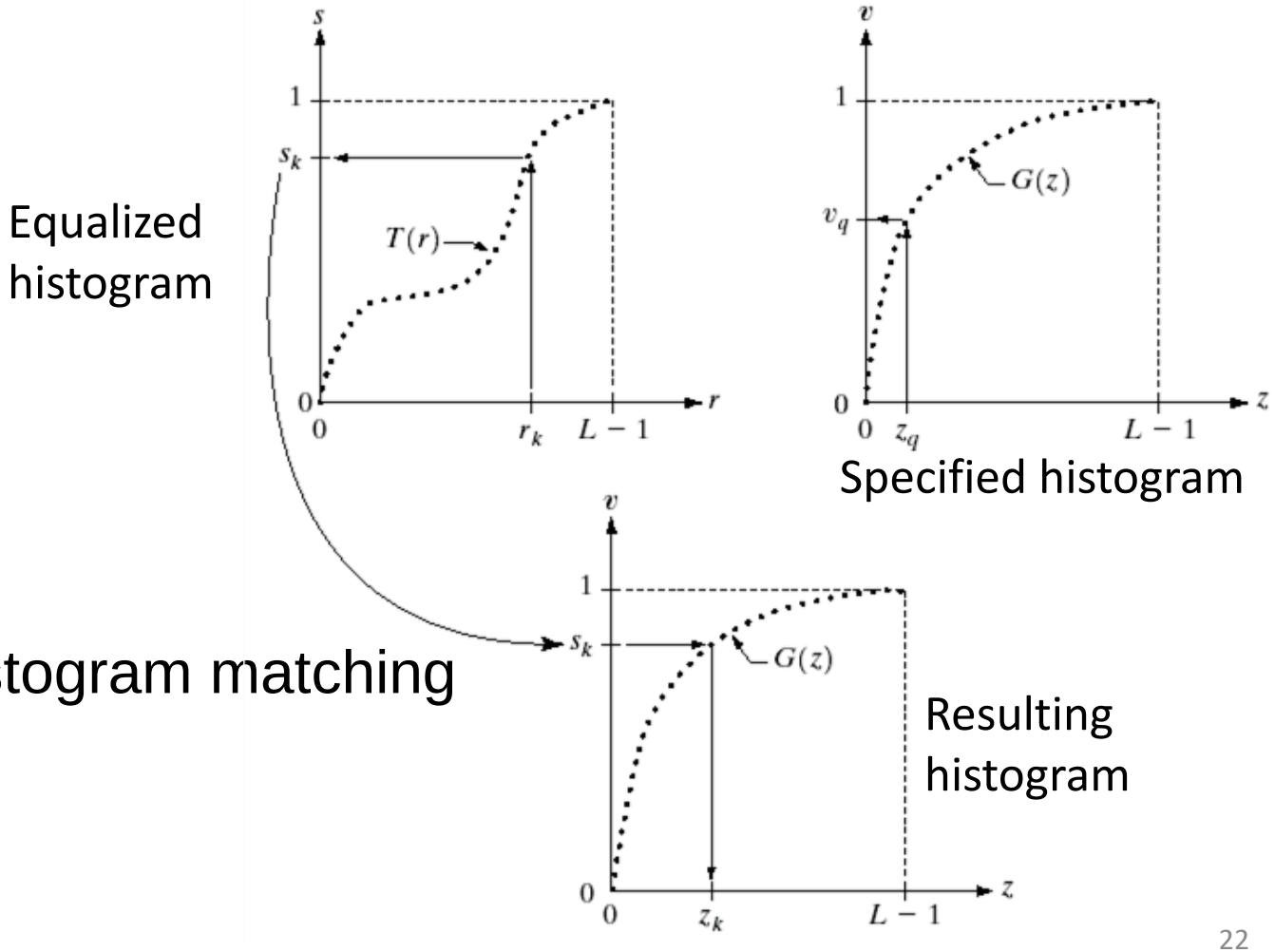
a b c

Left column: original images.
Center column: corresponding histogram-equalized images.
Right column: histograms of the images in the center column.



Histogram specification

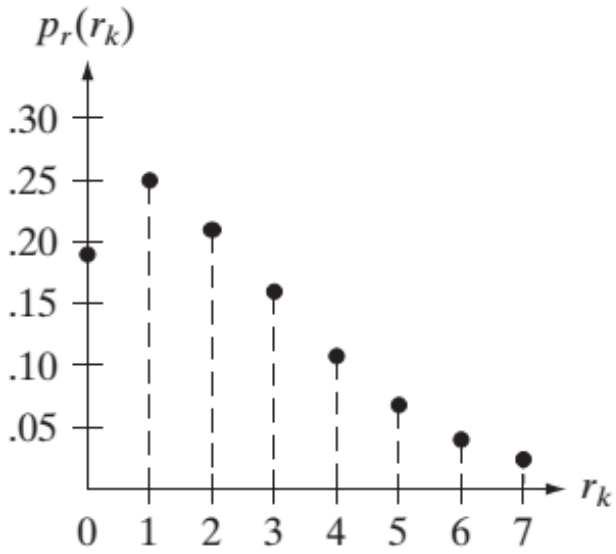
- Create an image whose histogram shape is specified

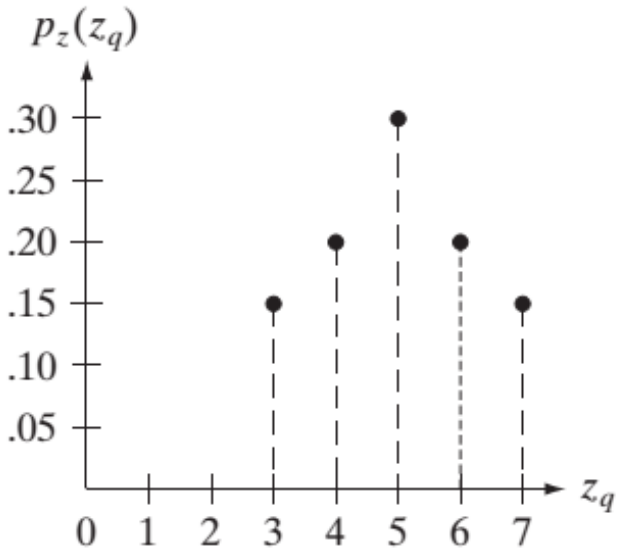


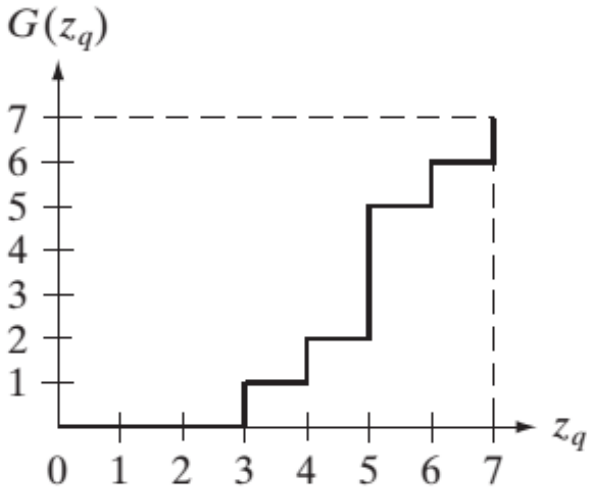
- Also called histogram matching

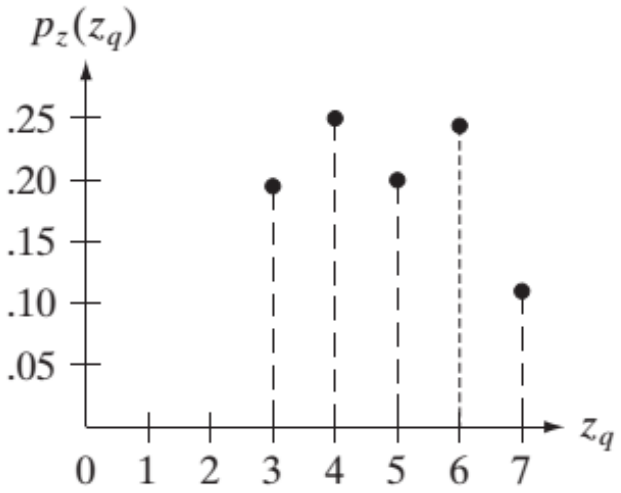
a b
c d

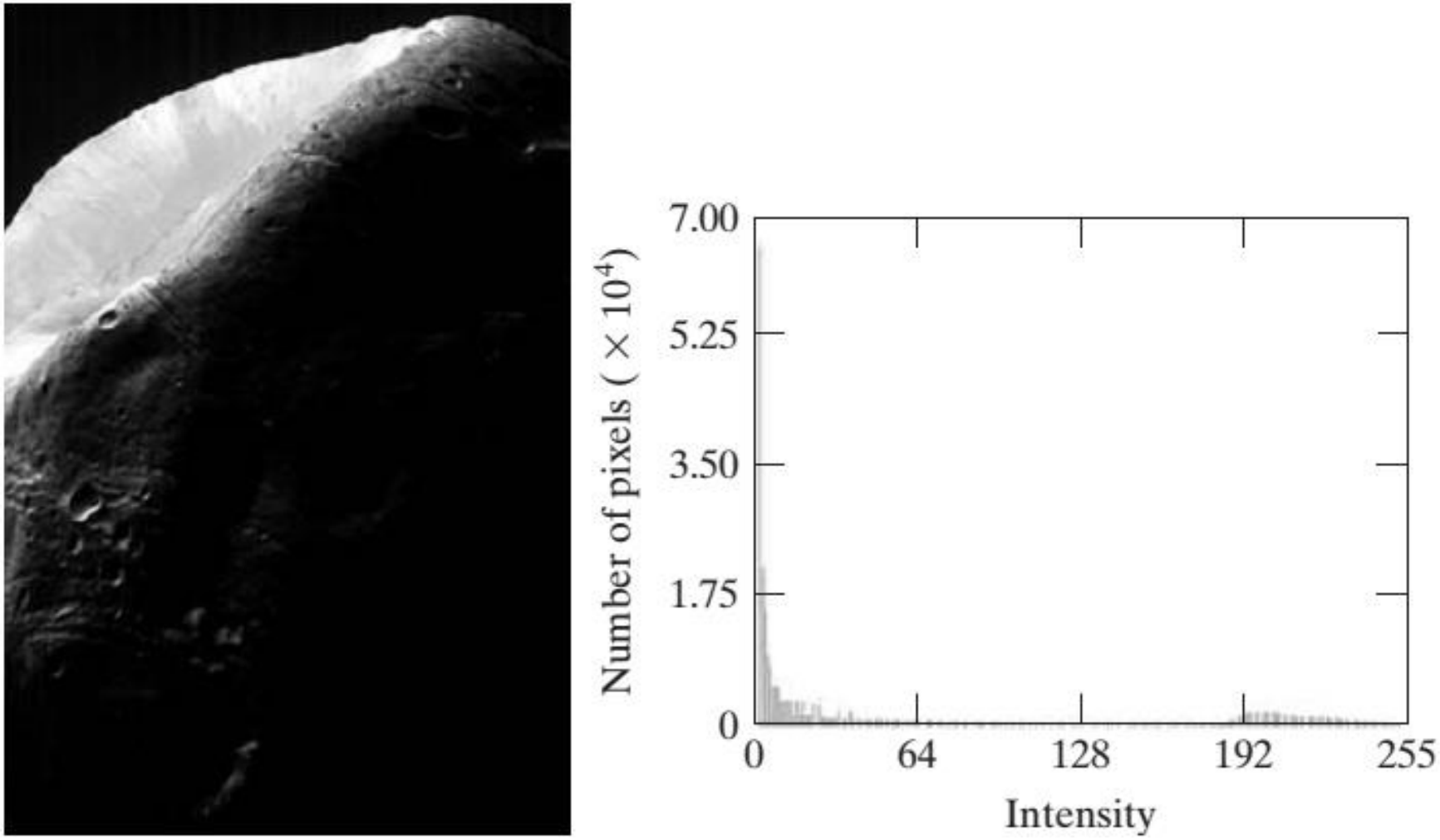
(a) Histogram of a 3-bit image.
(b) Specified histogram.
(c) Transformation function obtained from the specified histogram.
(d) Result of performing histogram specification.
Compare (b) and (d).





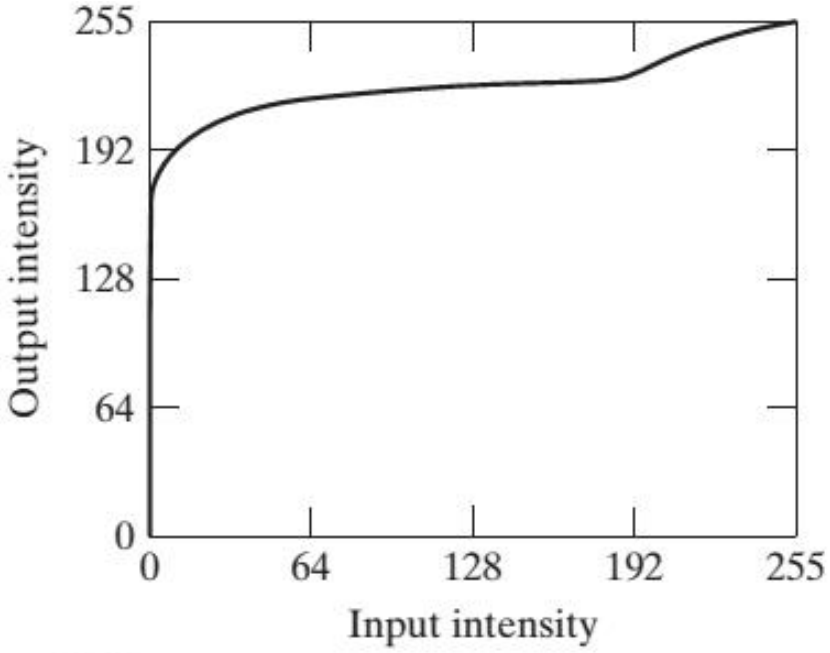


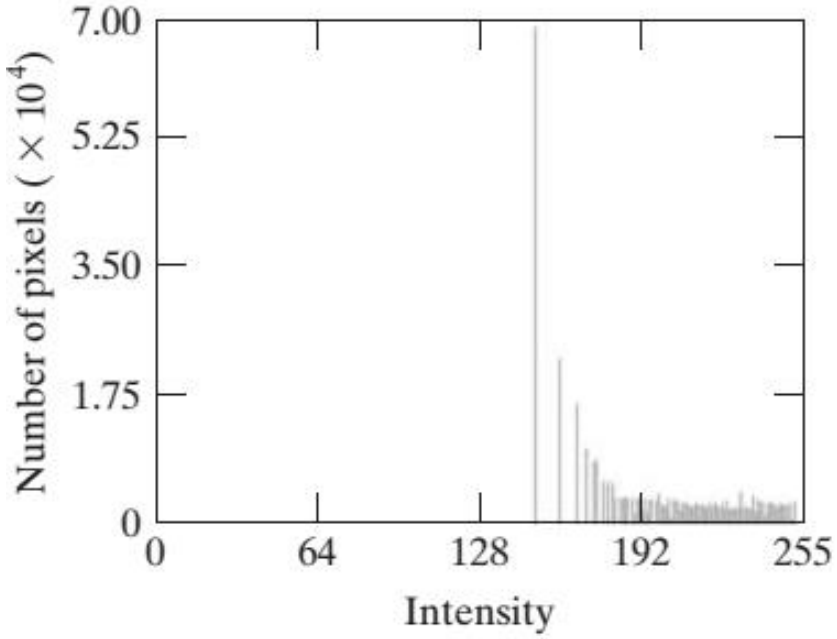




a b

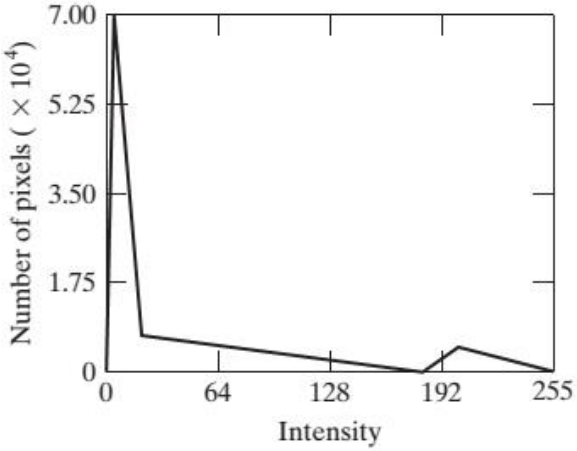
(a) Image of the Mars moon Phobos taken by NASA's Mars Global Surveyor.
(b) Histogram. (Original image courtesy of NASA.)

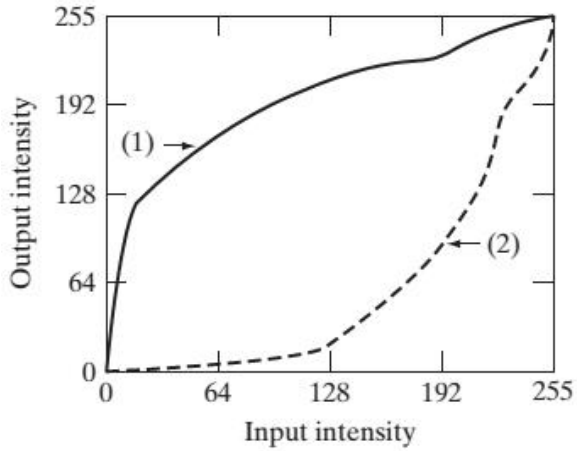


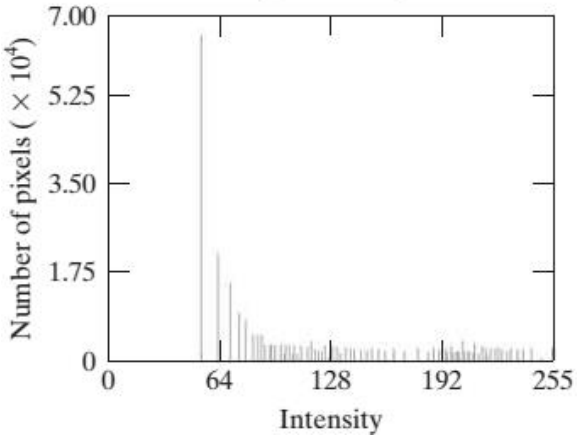


a b
c

(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).





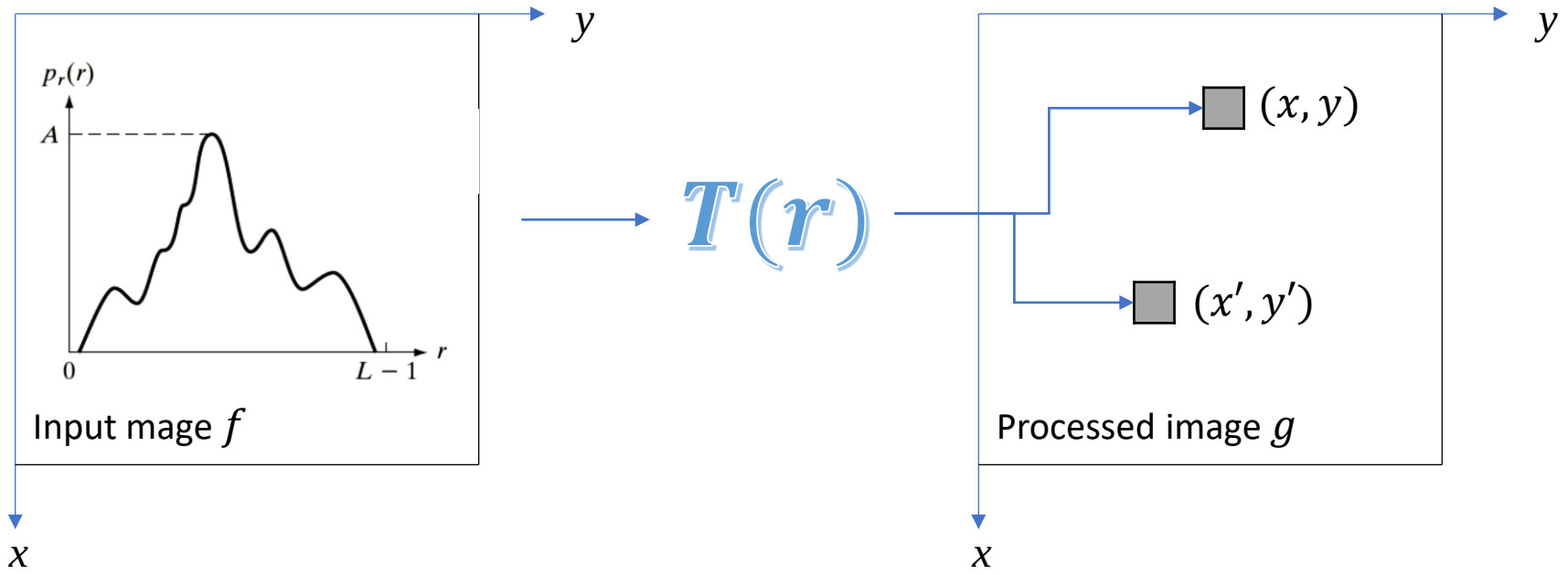


a c
b
d

(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

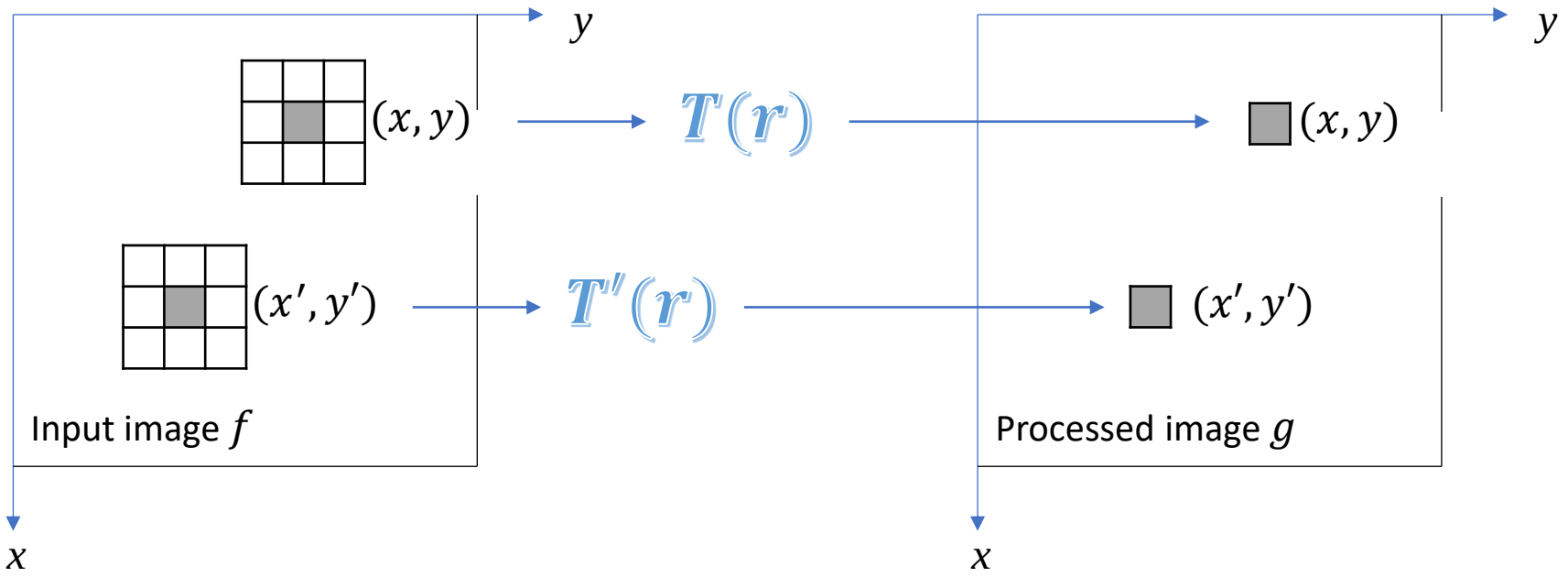
Histogram processing: Global vs. Local

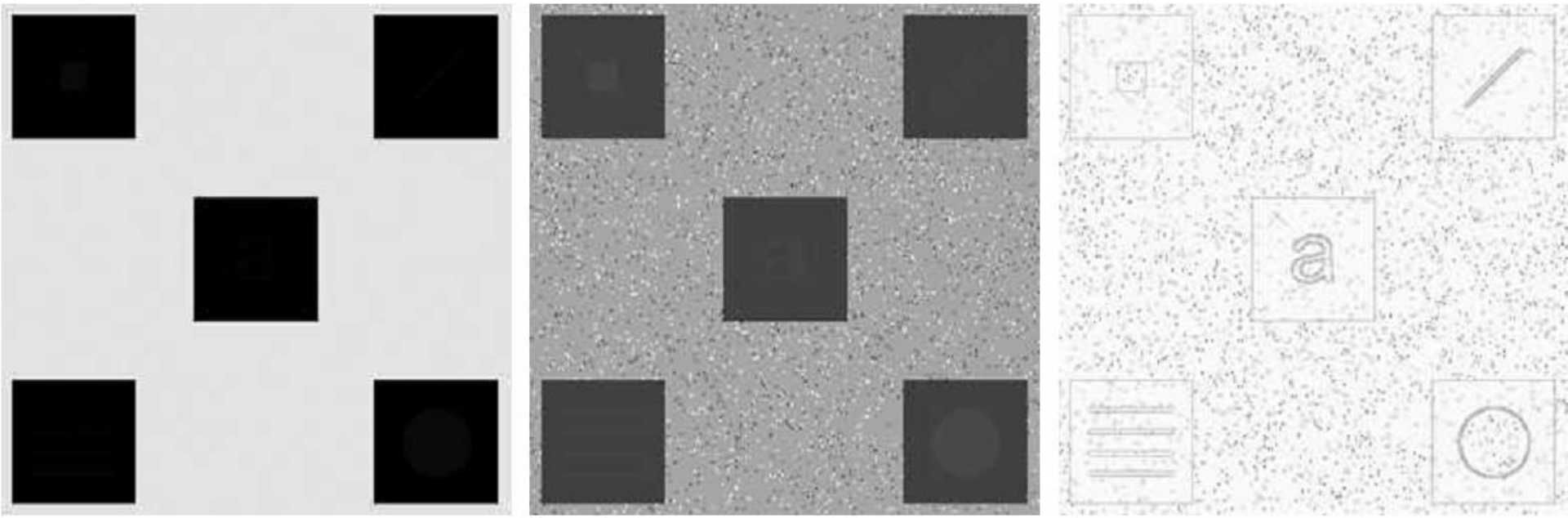
- Global histogram processing modifies the pixels based on the intensity distribution of an entire image.
- Suitable for overall enhancement
- Less effective to details over small areas



Histogram processing: Global vs. Local

- Local histogram processing modifies a pixel at location (x, y) based on the intensity distribution in the neighborhood centered on (x, y).





a b c

(a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3×3.

Global histogram statistics

- Let r denote a discrete random variable of intensity values in the range $[0, L-1]$ and $p(r_i)$ be the normalized histogram component corresponding to r_i
- The mean (a measure of average intensity) is given by

$$\boldsymbol{m = \sum_{i=0}^{L-1} r_i\, p(r_i)}$$

- The variance (a measure of contrast) is given by

$$\boldsymbol{\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2\, p(r_i)}$$

2-bit image ($L = 4$). Intensities are in the range $[0, L-1]$

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

The values of normalized histogram component $p(r_k)$

$$p(r_0) = \frac{6}{25} = 0.24 \qquad p(r_2) = \frac{7}{25} = 0.28$$

$$p(r_1) = \frac{7}{25} = 0.28 \qquad p(r_3) = \frac{5}{25} = 0.20$$

The (global) mean and variance

$$m = \sum_{i=0}^{3} r_i \, p(r_i) = (0)(0.24) + (1)(0.28) + (2)(0.28) + (3)(0.20) = 1.44$$

$$\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 \, p(r_i)$$

$$= (0 - 1.44)^2(0.24) + (1 - 1.44)^2(0.28) + (2 - 1.44)^2(0.28) + (3 - 1.44)^2(0.20)$$

$$= 1.1264$$

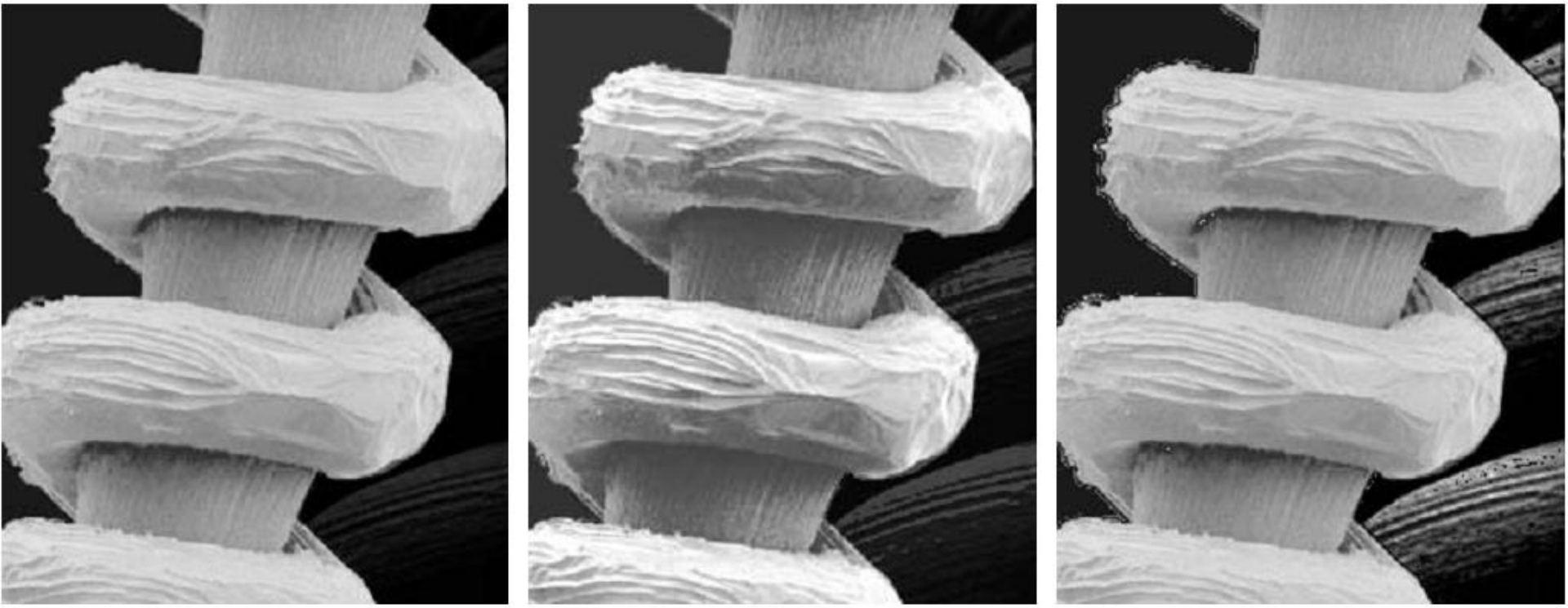
Local histogram statistics

- Let (x, y) denote the coordinates of any pixel in an image and S_{xy} be a neighborhood centered on (x, y)
- Let p_{xy} the histogram of the pixels in region S_{xy}
- The mean value of the pixels in this neighborhood is

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i \, p_{S_{xy}}(r_i)$$

- The variance of the pixels in the neighborhood similarly is

$$\sigma^2_{S_{xy}} = \sum_{i=0}^{L-1} \left(r_i - m_{S_{xy}}\right)^2 p_{S_{xy}}(r_i)$$

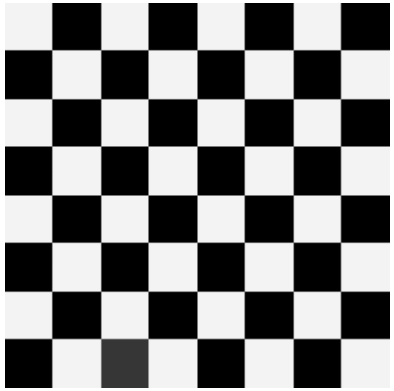


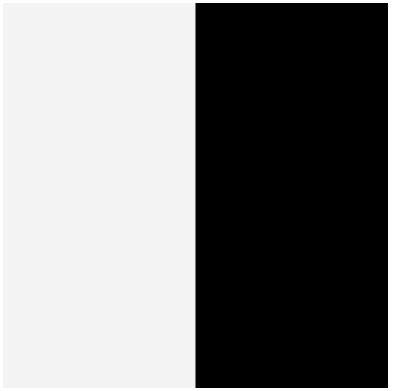
a b c

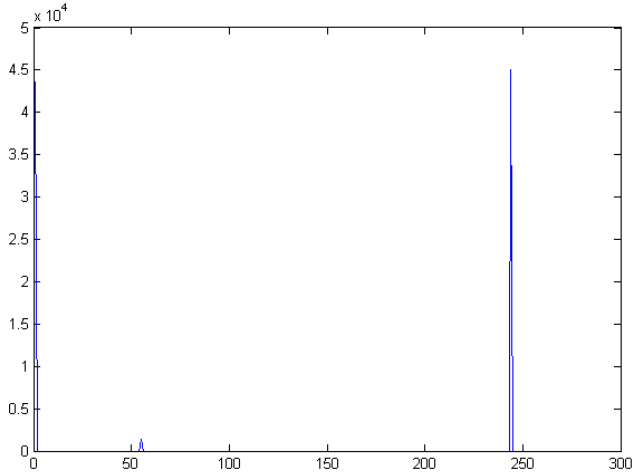
(a) SEM image of a tungsten filament magnified approximately. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Can histogram be a good feature?

- Histogram, as well as some other statistical values, is a weak feature since it is too general

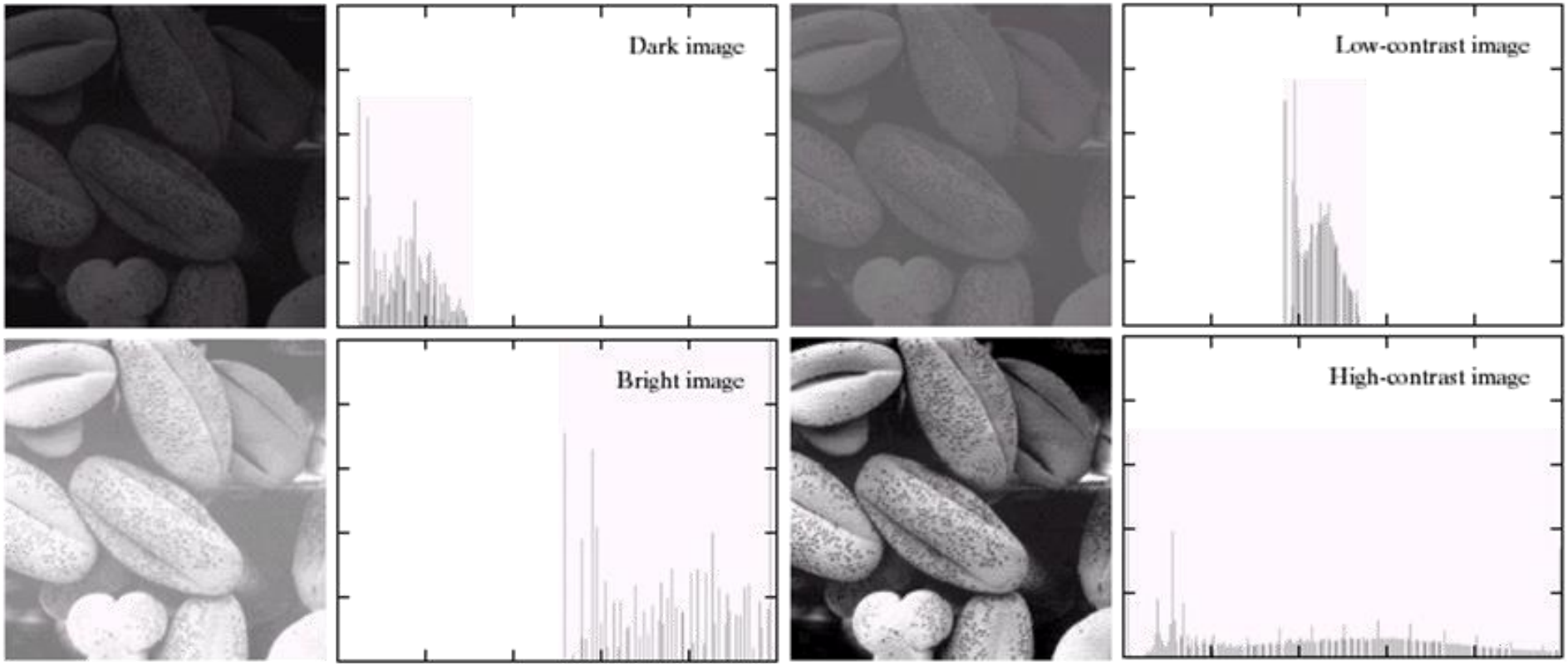






Can intensity be a good feature?

- Intensity is sensitive to noise and illumination changes.
- The relation between intensities may reduce the effect of illumination changes yet the effect of noise remains.



Can color be a good feature?

- No. Color is sensitive to noise and illumination changes

- Some color spaces (e.g., CIE-Lab, CIE-Luv) are empirically shown to be more discriminative than the others (e.g., RGB)

Edge features

Edge detection

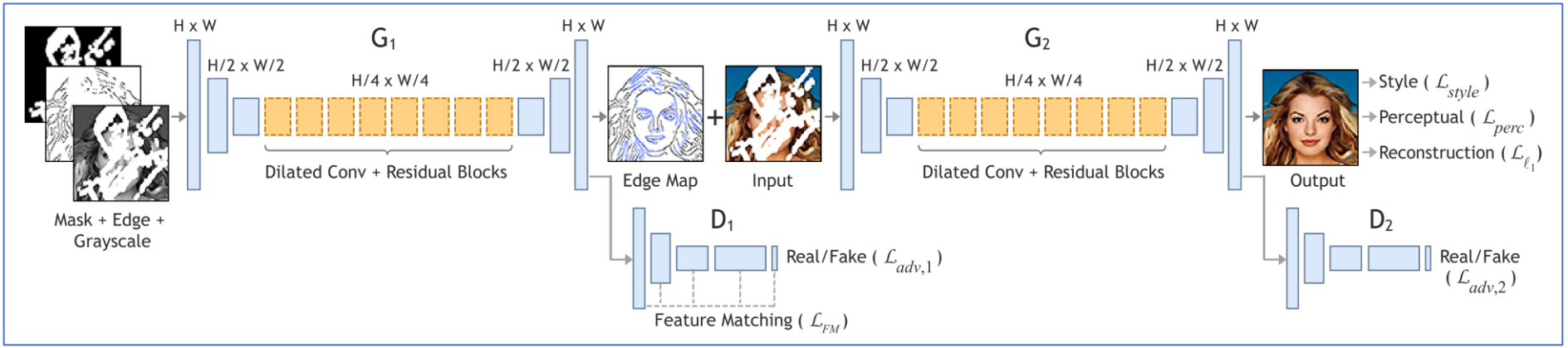
- Edges are defined as curves at which the pixel brightness changes sharply or, more formally, has discontinuities.
- Edge detection includes a variety of mathematical methods that aim at identifying edges.

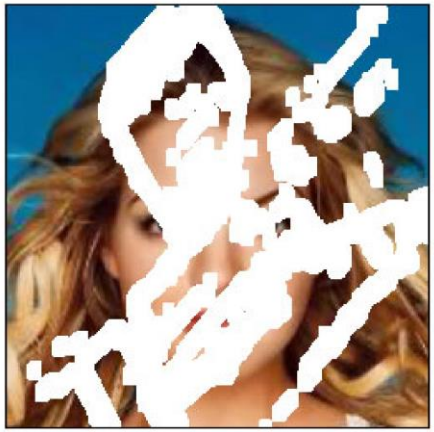
Input image

Edges detected by Prewritt detector

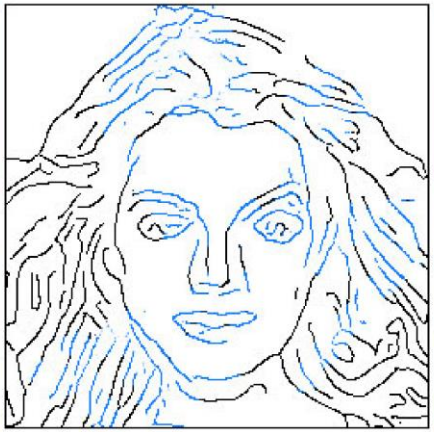
Image credit: RoboFlow

Edges as augmented features





Input image



Edge map

Resulting image

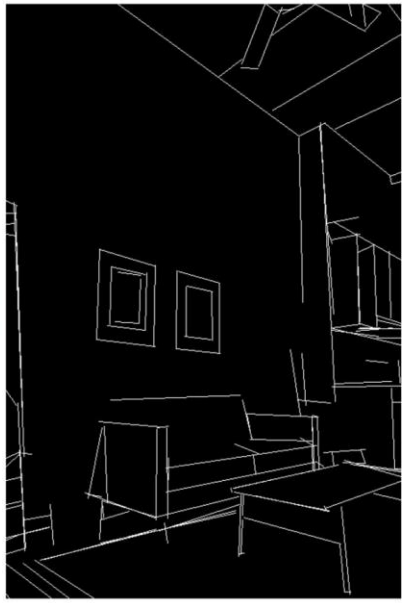
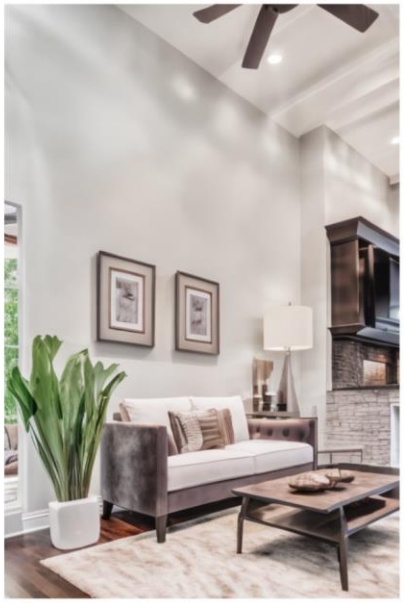
Edges drawn in black are computed (for the available regions) using Canny edge detector; whereas edges shown in blue are hallucinated (for the missing regions) by the edge generator network.

Nazeri, K. "EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning." arXiv preprint arXiv:1901.00212 (2019).

Edges as augmented features





Input image

Edge map

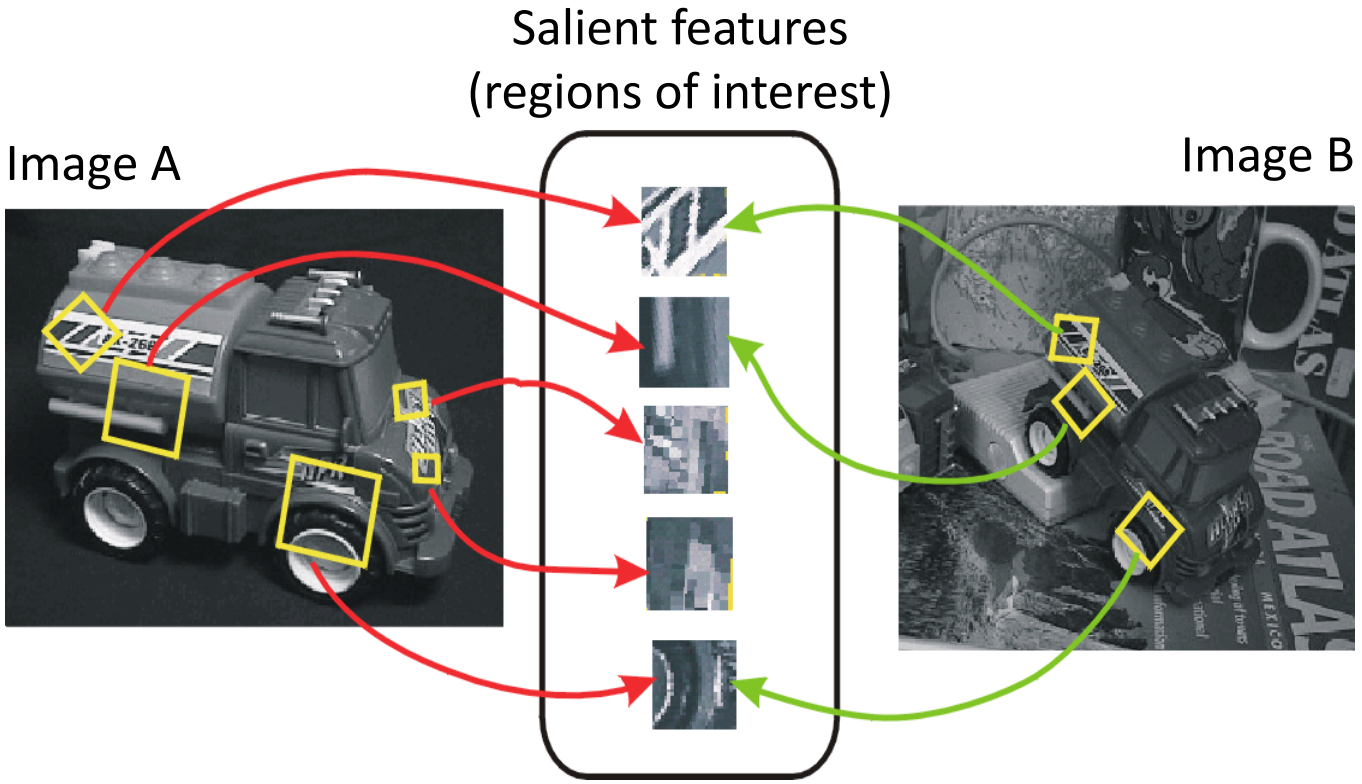
Resulting image

Ye, Hu, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. "Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models." arXiv preprint arXiv:2308.06721 (2023).

Local feature descriptors

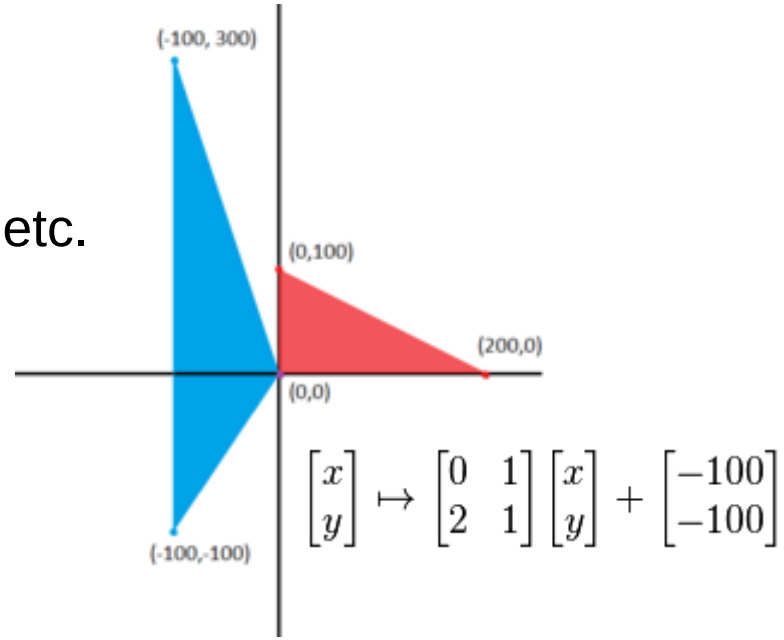
Local features: A definition

- Local features encode image structures in the spatial neighborhoods at a set of feature points chosen at selected scales or orientations



Requiments of a local feature

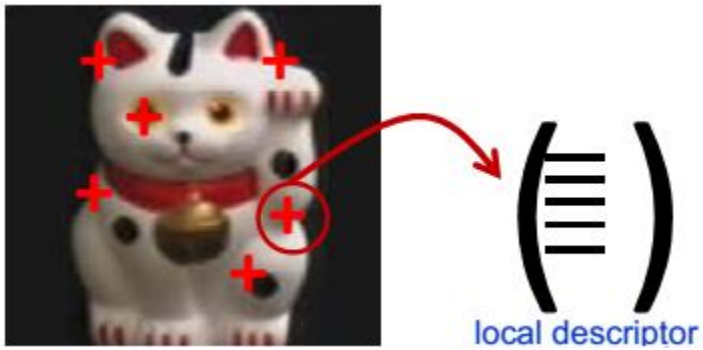
- Invariant to geometric (i.e., affine) transformations
  - Translation
  - Scaling
  - Rotation
  - Sheer mapping, squeeze mapping, etc.

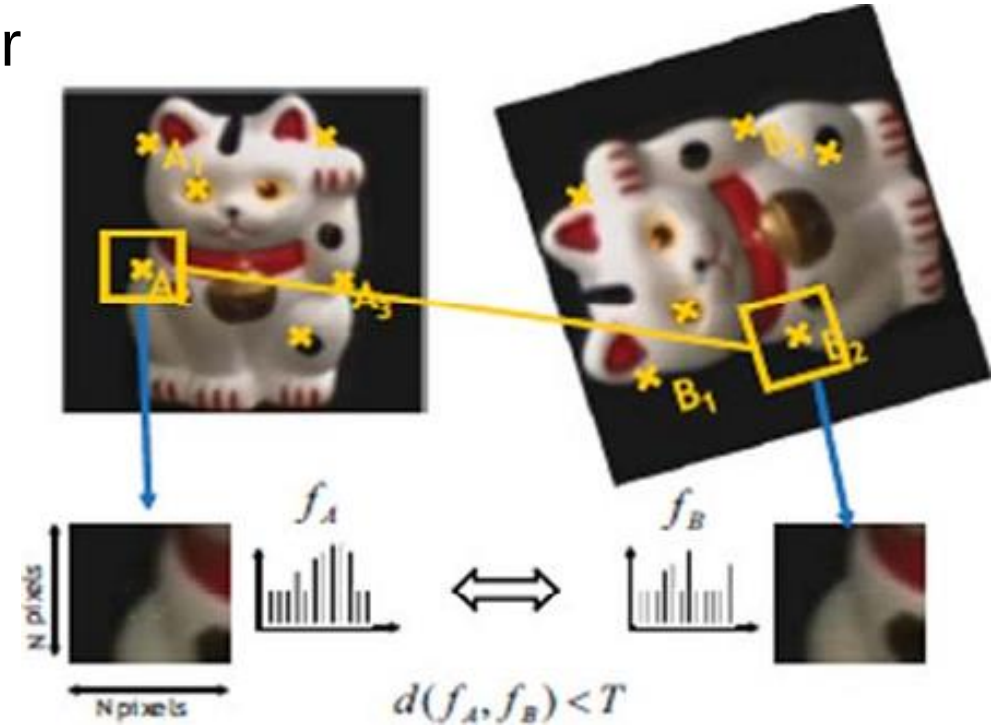


- Invariant to photometric (illumination, exposure) changes
- Less affected by noise or blur

Matching with local features

1. Find the interest points
2. Consider the region around each keypoint
3. Compute a local descriptor from the region and normalize the feature
4. Match local descriptor





Types of local features

- Gradient-based local features
  - Scale-Invariant Feature Transform (SIFT) [Lowe, 2004]

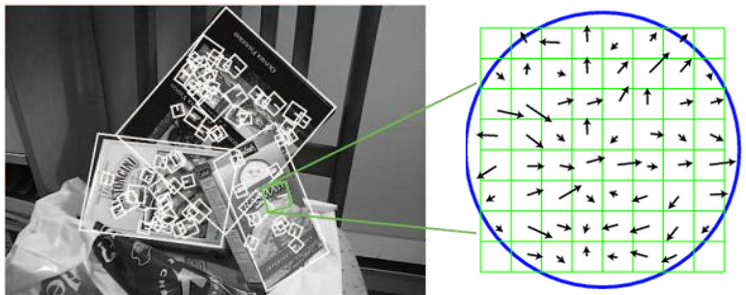
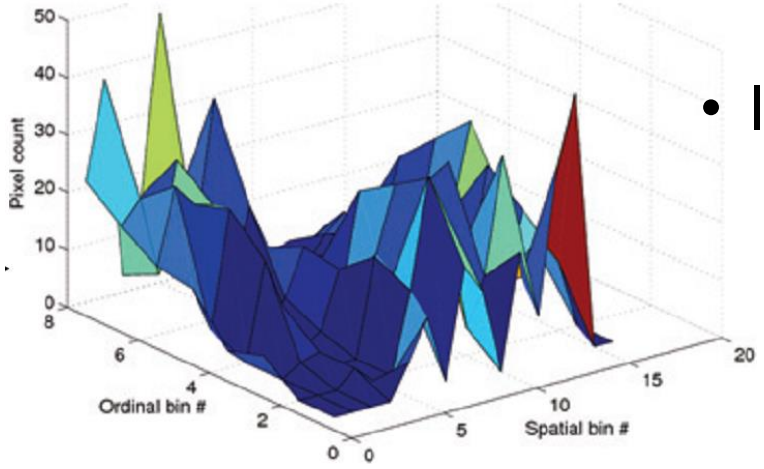
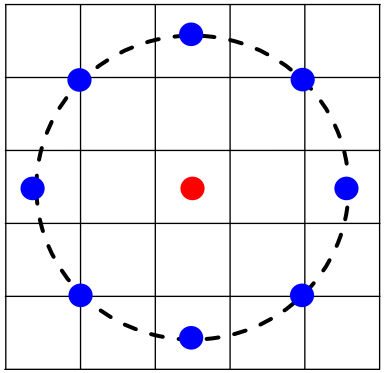


Image gradients



- Intensity-based local features
  - Local intensity order pattern (LIOP) [Wang et al., 2011]



- LBP-based local features
  - Local Binary Pattern (LBP) [Ojala et al., 2002]

Types of local features

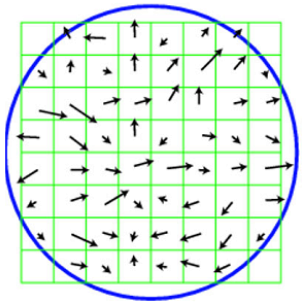
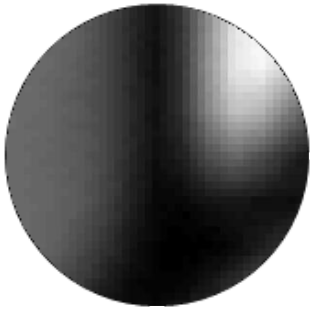


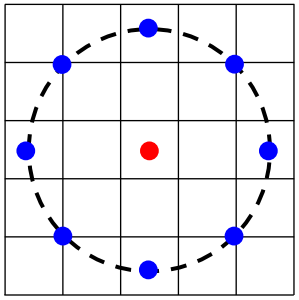
Image gradients

- ✓ discriminative to directional changes
- ✗ computationally heavy



Grayscale intensity

- ✓ invariant to illumination changes
- ✓ computationally light
- ✗ sensitive to noise

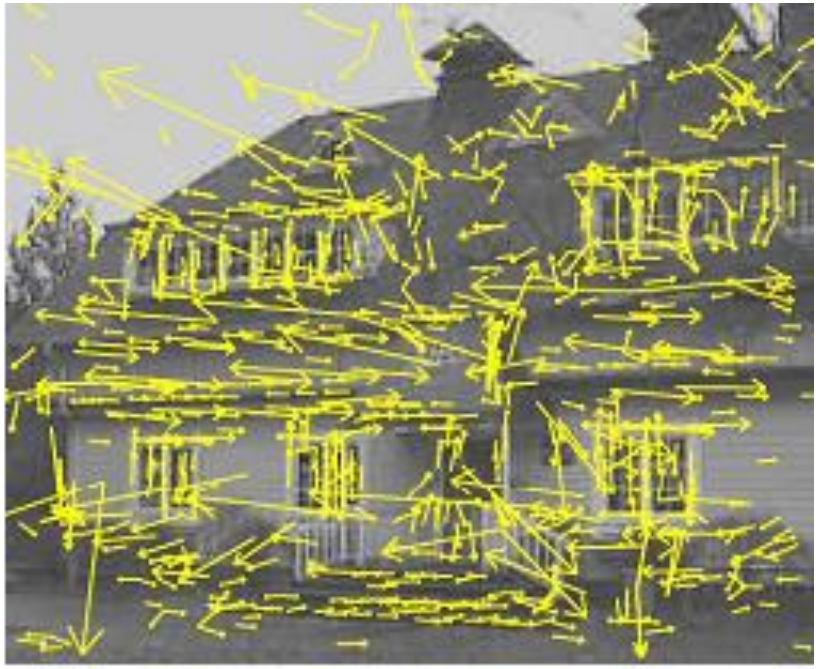


Local Binary Pattern

- ✓ invariant to illumination changes
- ✓ computationally light
- ✓ robust to noise
- ✗ high dimensionality

Scale Invariant Feature Transform (SIFT)

- Step 1: Detect the interesting points using Difference of Gaussians (DOG)



832 DOG extrema

Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.

Scale Invariant Feature Transform (SIFT)

- Step 2: Accurate keypoint localization
  - Aim: reject low contrast points and points that lie on the edge
  - Reject low contrast points
    - Fit keypoint at $\underline{x}$ to nearby data using quadratic approximation

$$D(\underline{x}) = D + \frac{\partial D^T}{\partial \underline{x}}\underline{x} + \frac{1}{2}\underline{x}^T\frac{\partial^2 D^T}{\partial \underline{x}^2}\underline{x}$$

where $D(x,\sigma) = [G(x,k\sigma) - G(x,\sigma)] * I(x)$

    - Calculate the local maxima of the fitted function $\{\underline{X} = (x, y, \sigma)\}$
    - Discard local minima (for contrast) $D(\hat{\underline{x}}) < 0.03$

$$\frac{\partial D}{\partial \underline{x}} = \frac{\partial\left[D + \frac{\partial D^T}{\partial \underline{x}}\underline{x} + \frac{1}{2}\underline{x}^T\frac{\partial^2 D^T}{\partial \underline{x}^2}\underline{x}\right]}{\partial \underline{x}} = 0 \Rightarrow \hat{\underline{x}} = -\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D}{\partial \underline{x}}$$

Scale Invariant Feature Transform (SIFT)

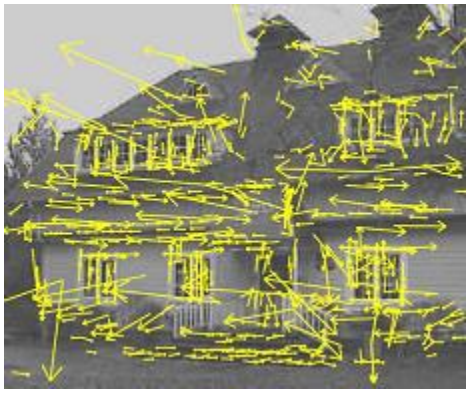
- Step 2: Accurate keypoint localization
  - Eliminating edge response: DOG gives strong response along edges
    $\Rightarrow$ Eliminate those responses
    - Solution: check “cornerness” of each keypoint
      - On the edge, one of principle curvatures is much bigger than another
      - High cornerness $\Leftrightarrow$ No dominant principal curvature component
    - Consider the concept of Hessian and Harris corner

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \qquad \frac{\text{trace}\,(H)^2}{\det H} < \frac{(r+1)^2}{r}$$

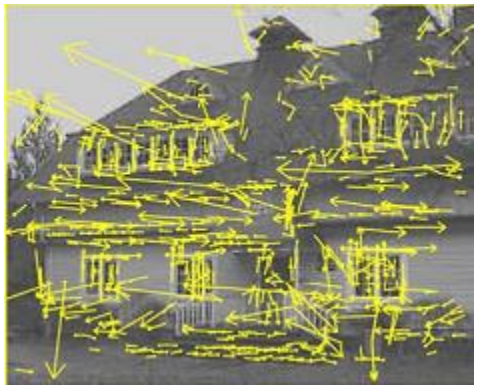
Discard points with response below threshold

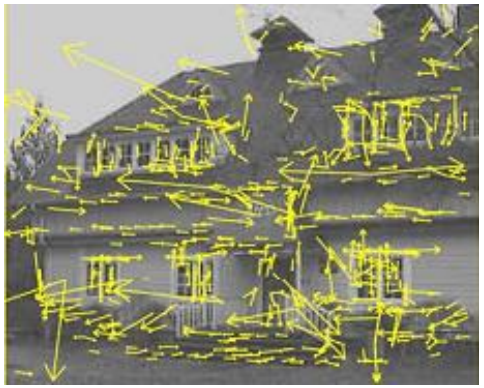
Scale Invariant Feature Transform (SIFT)

- Step 2: Accurate keypoint localization



  - 729 out of 832 are left after contrast thresholding





  - 536 out of 729 are left after cornerness thresholding

Scale Invariant Feature Transform (SIFT)

- Step 3: Orientation assignment
  - Aim: Assign constant orientation to each keypoint based on local image property to obtain rotational invariance

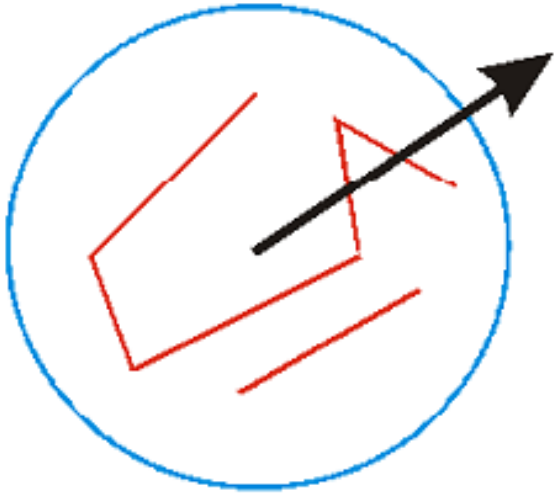
  - The magnitude and orientation of gradient of an image patch $I(x, y)$ at a particular scale is

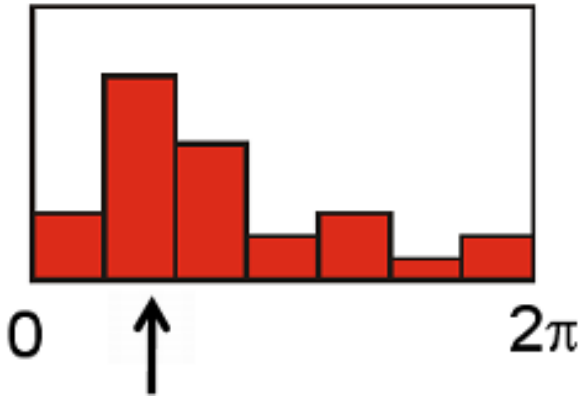
$$m(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1} \frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}$$

Scale Invariant Feature Transform (SIFT)

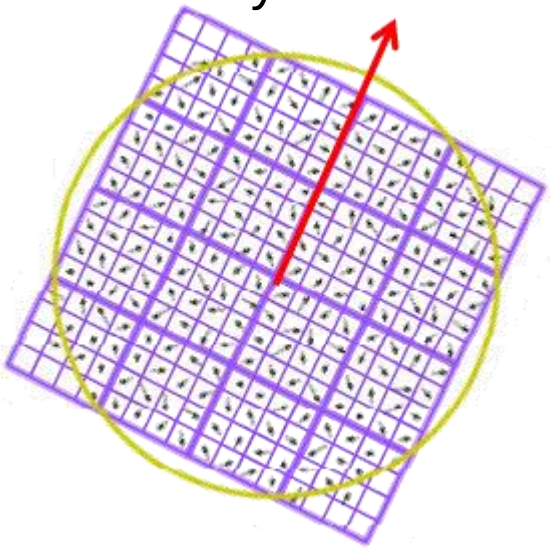
- Step 3: Orientation assignment
  - Create weighted (magnitude + Gaussian) histogram of local gradient directions computed at selected scale
  - Assign dominant orientation of the region as that of the peak of smoothed histogram
  - For multiple peaks create multiple keypoints





Scale Invariant Feature Transform (SIFT)

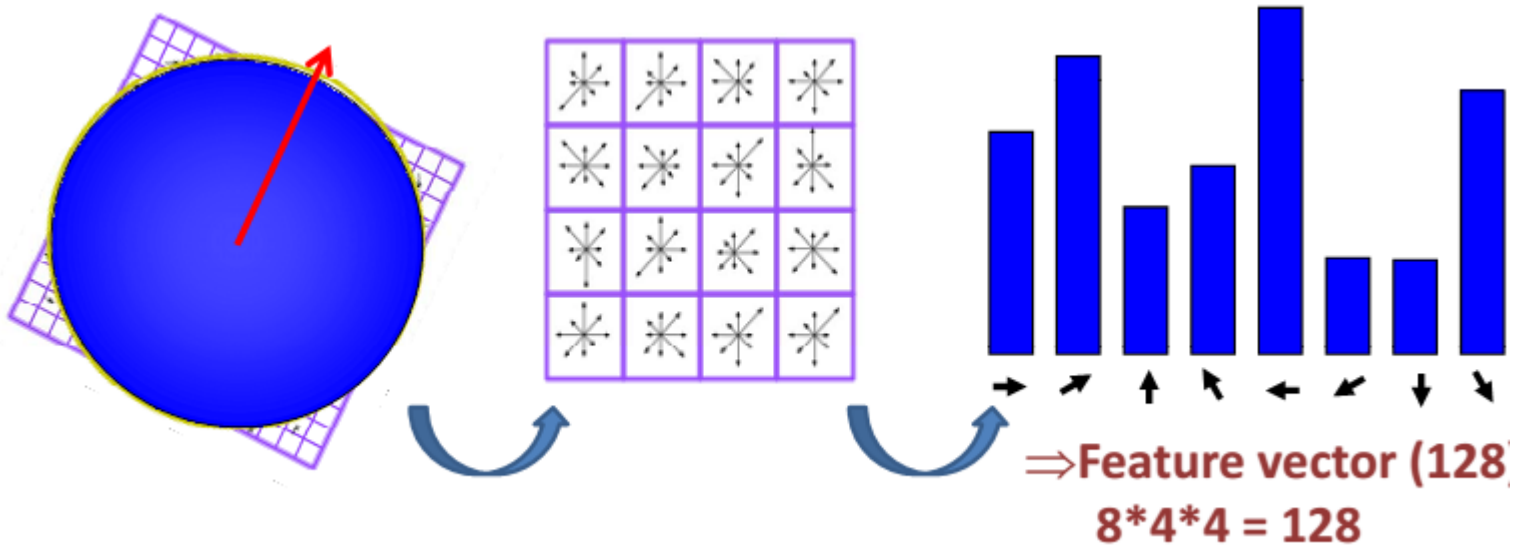
- Prior condition: Precise location, scale and orientation to each keypoint are already obtained
- Step 4: Local image descriptor
  - Aim: Obtain local descriptor that is highly distinctive yet invariant to variation like illumination and affine change
  - Consider a rectangular grid 16×16 in the direction of the dominant orientation of the region.



  - Divide the region into 4×4 sub-regions.
  - Consider a Gaussian filter above the region which gives higher weights to pixel closer to the center of the descriptor

Scale Invariant Feature Transform (SIFT)

- Step 4: Local image descriptor
  - Create an 8-bin gradient histogram for each sub-region



  - Finally normalize 128-dim vector to make it illumination invariant

SIFT: Applications

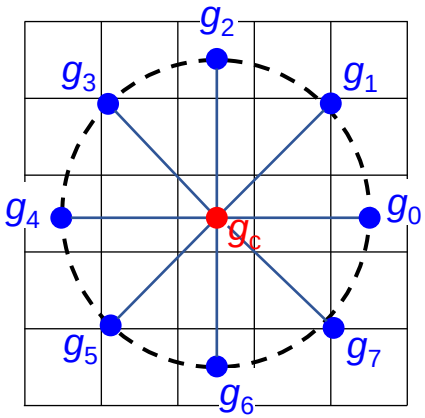
- Panorama image synthesis

Local Binary Patterns (LBP)

- A texture operator that describes the local information around each pixel.

$$\mathrm{LBP}_{P,R}(x,y) = \sum_{p=0}^{P-1} s(g_p - g_c)\, 2^p,$$

$$s(z) = \begin{cases} 1 & z \geq 0 \\ 0 & otherwise \end{cases}$$



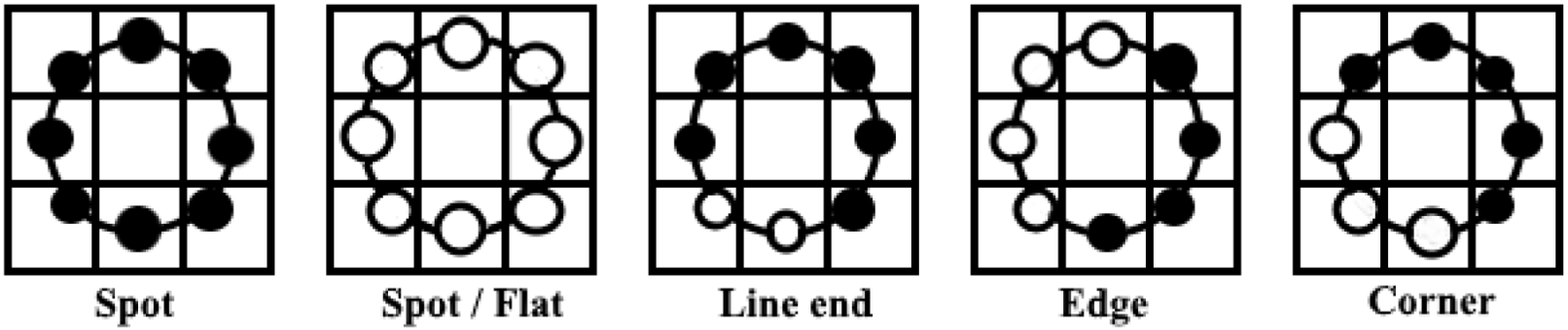
R: radius of the neighborhood, P: number of neighbors

g_c, g_p : the gray value of the center pixel and of p^{th} neighboring pixels

Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." IEEE Transactions on pattern analysis and machine intelligence 24.7 (2002): 971-987.

Local Binary Patterns (LBP)

- LBP can detect several simple yet essential patterns.



Input image

gray values in the $n \times n$ neighborhood

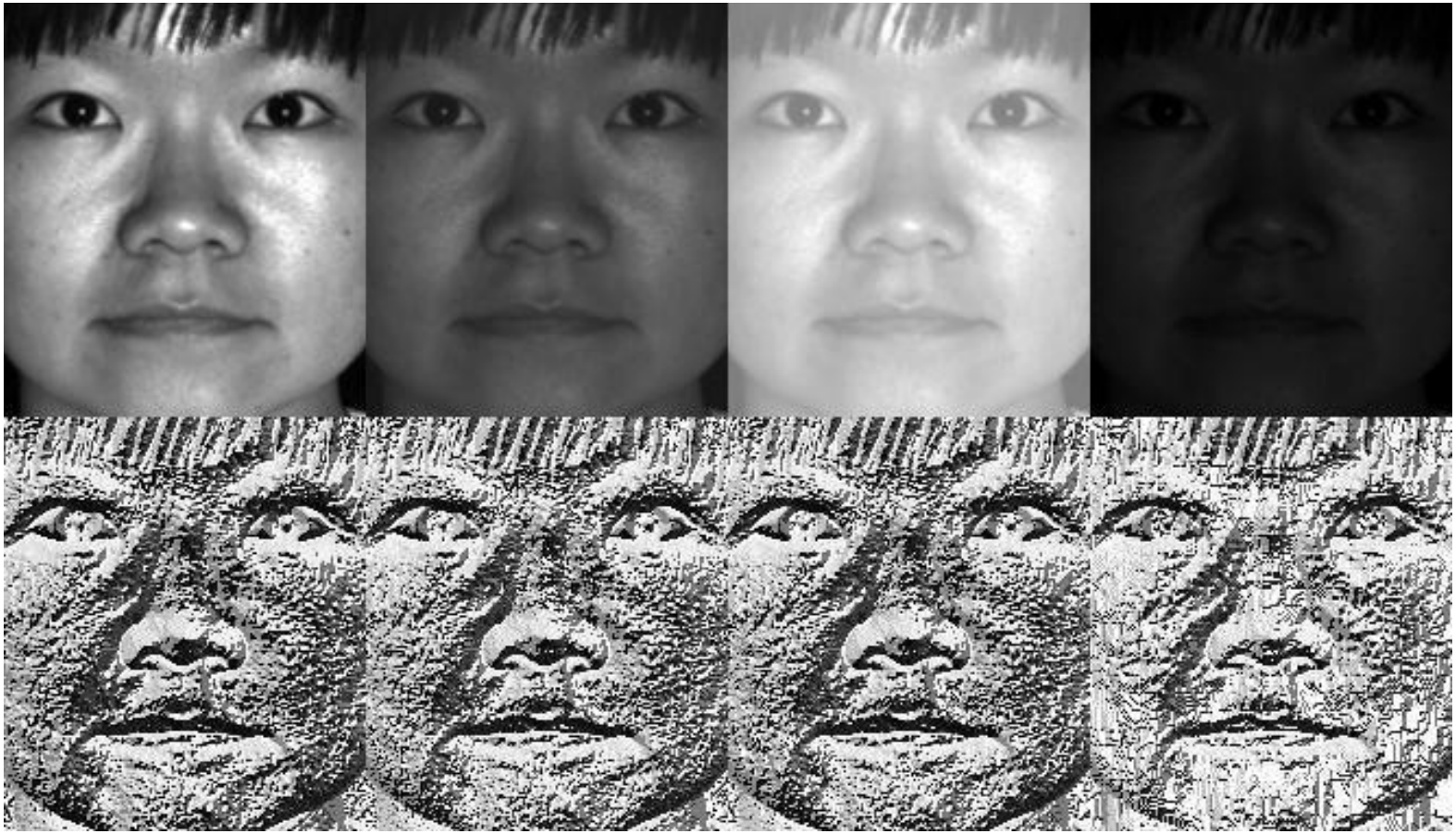
137	140	143
144	140	139
132	135	136

Output image

Properties of LBP

1. Invariant to any monotonic gray-level transformation
2. Nonparametric method
    - Require no assumptions about the underlying distribution
3. Highly discriminative against illumination changes
4. The operator is intuitive and computationally simple
5. The LBP code is quantized by its nature

Properties of LBP



Drawbacks of LBP

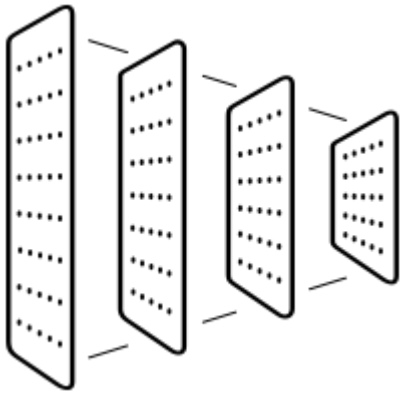
- Thresholding function $s(g_p - g_c)$
  - Unstable on noisy or near-uniform regions
  - Fail to deal with image details whose $g_p - g_c$ are of the same sign yet different magnitudes.

$$s(g_p - g_c) = \begin{cases} 1 & g_p - g_c \geq 0 \\ 0 & otherwise \end{cases}$$

$$g_c = 29, g_p = 30 \Rightarrow s(g_c - g_p) = 0$$
$$g_c = 30, g_p = 30 \Rightarrow s(g_c - g_p) = 1$$

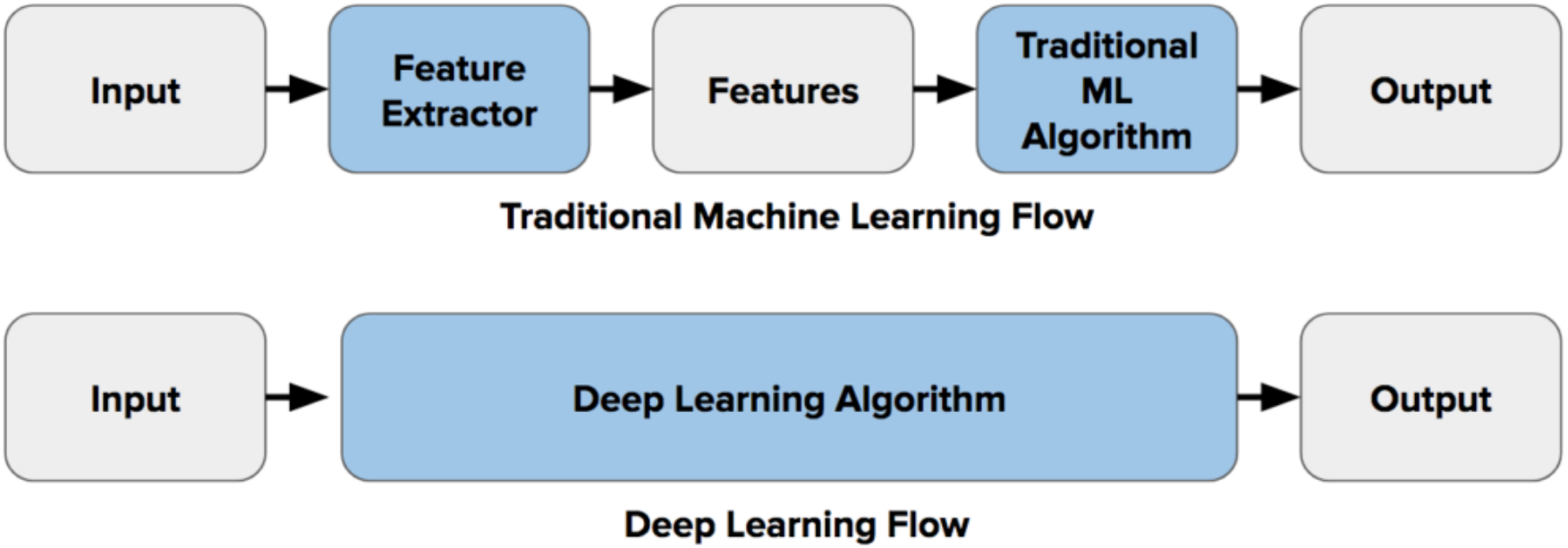
- The feature vectors are usually high dimensional.
  - $LBP_{8,R}$ has 2^8 (256) dimensions

Auttomatic features



Feature extraction

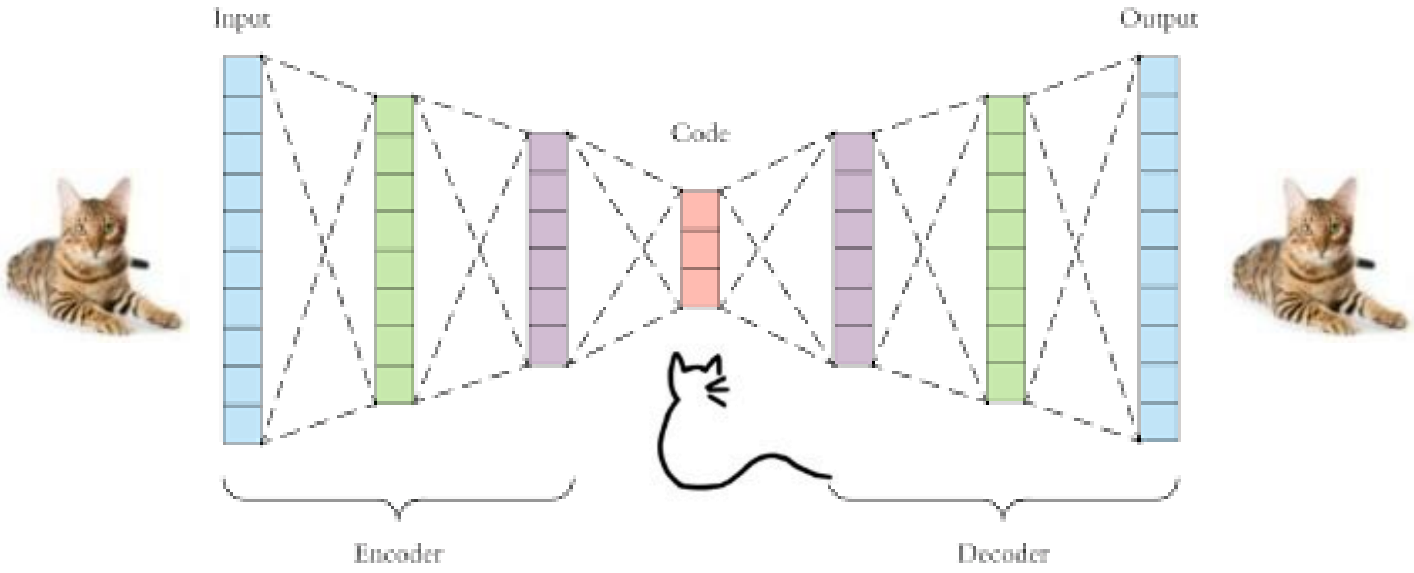
- Handcrafted features are manually engineered by the scientist.



- Automatic features are inherently learned from a ML algorithm.

Features from an Autoencoder

- An autoencoder is a neural network that is trained to attempt to copy its input to its output.

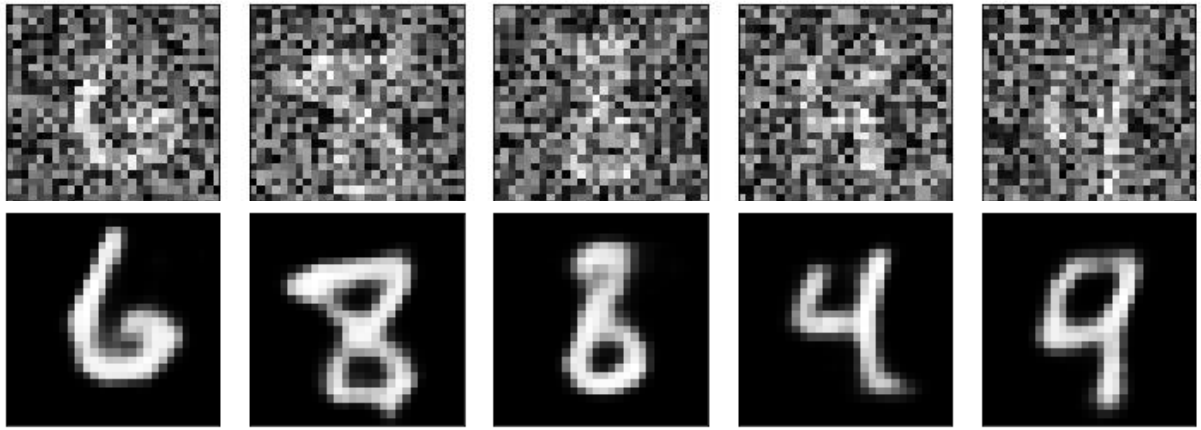


- Once trained, the decoder is discarded, and the encoder is used as needed to create compact representations of input.

Autoencoder: Some applications

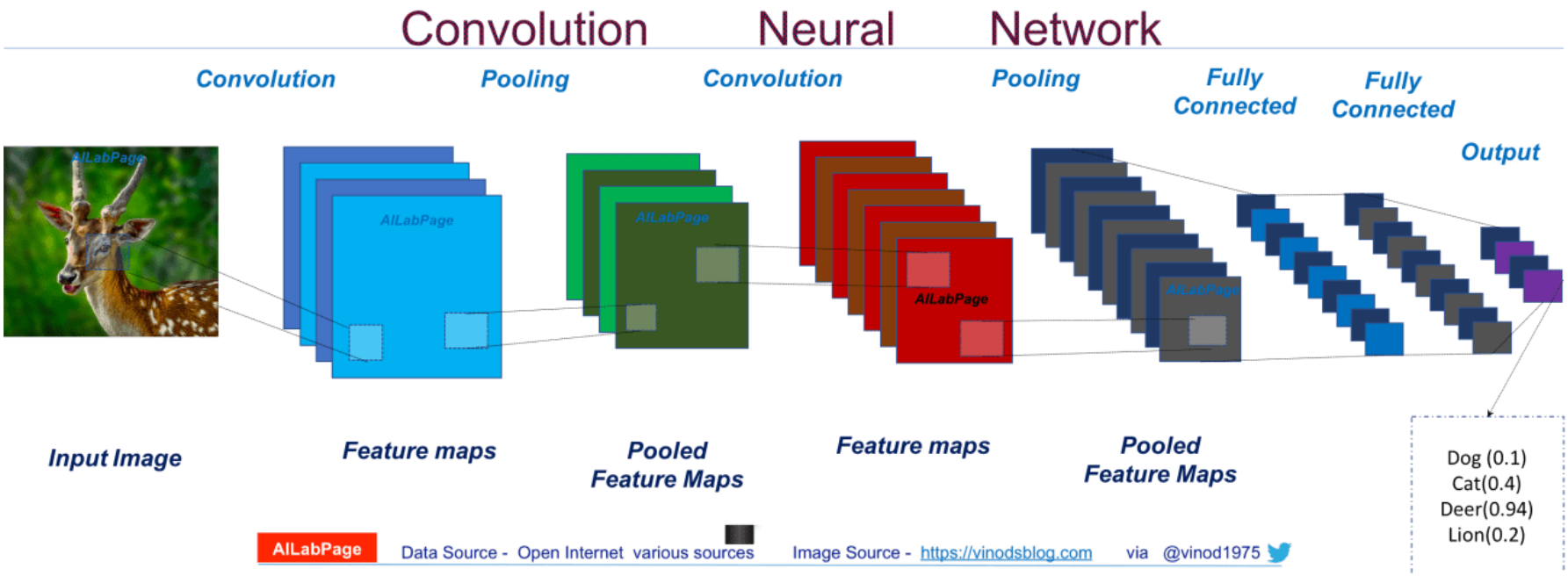
- Image colorization

- Image denoising



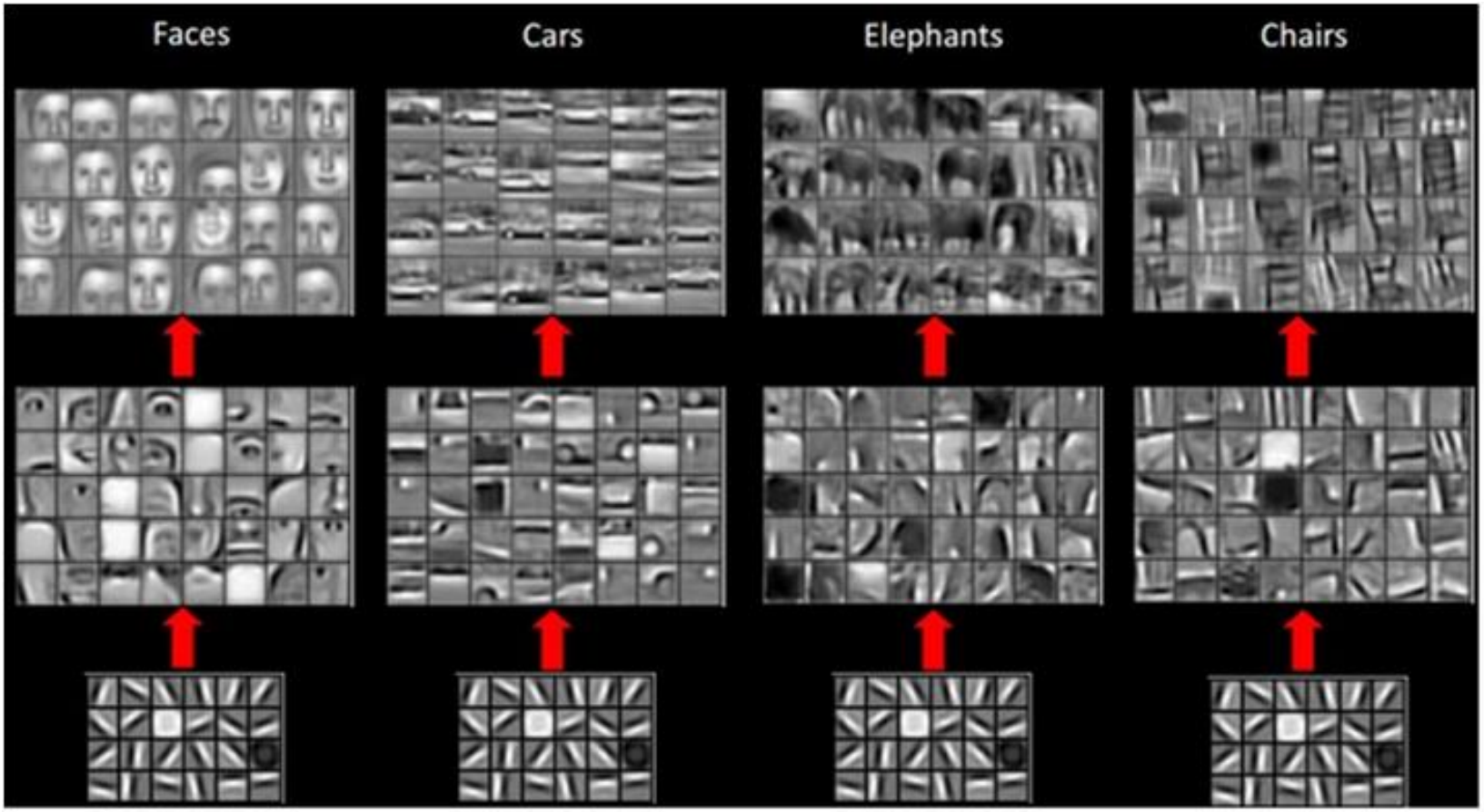
Features from a CNN

- Features in the last layer of a convolutional neural network is usually extracted for further learning in another model.



Features from a CNN

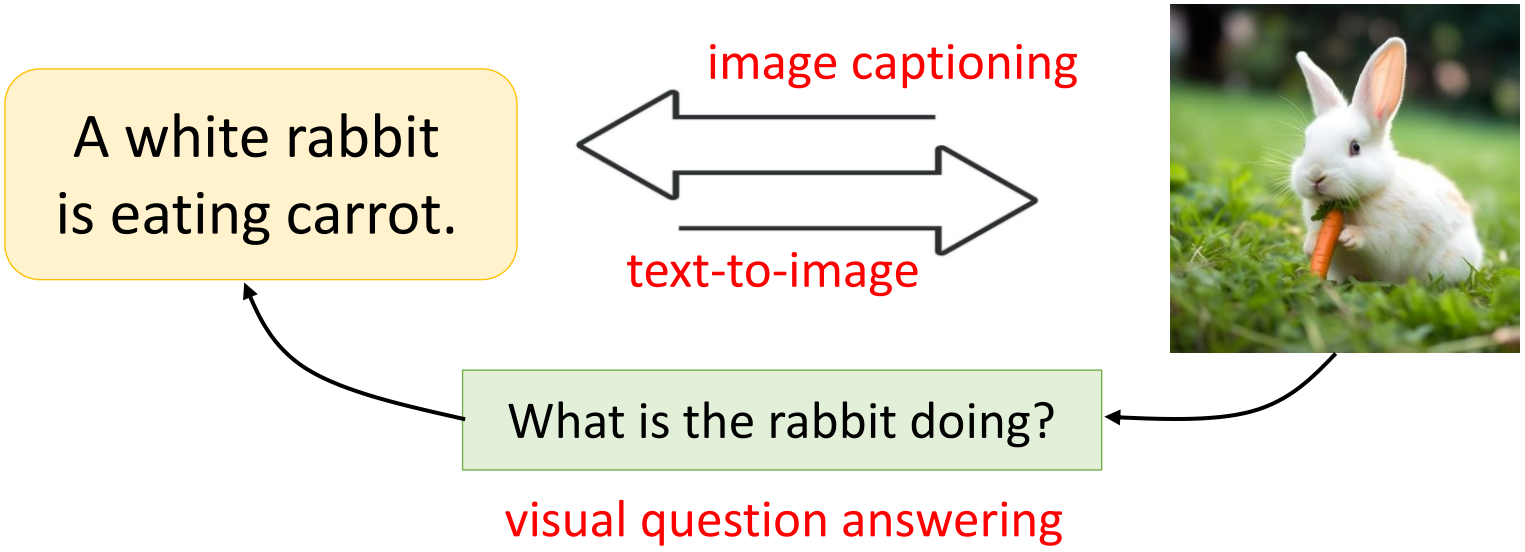
- CNN extracts features from the input image in a hierarchical way, from low-level cues to more abstract shapes.



Vision-Language Pre-training

Vision-Language Pre-training (VLP)

- VLP jointly trains models on both visual and textual data, enabling them to understand and generate contextually relevant outputs across both modalities.
- It serves as a base for tasks requiring a deep understanding of the relationship between images and text.

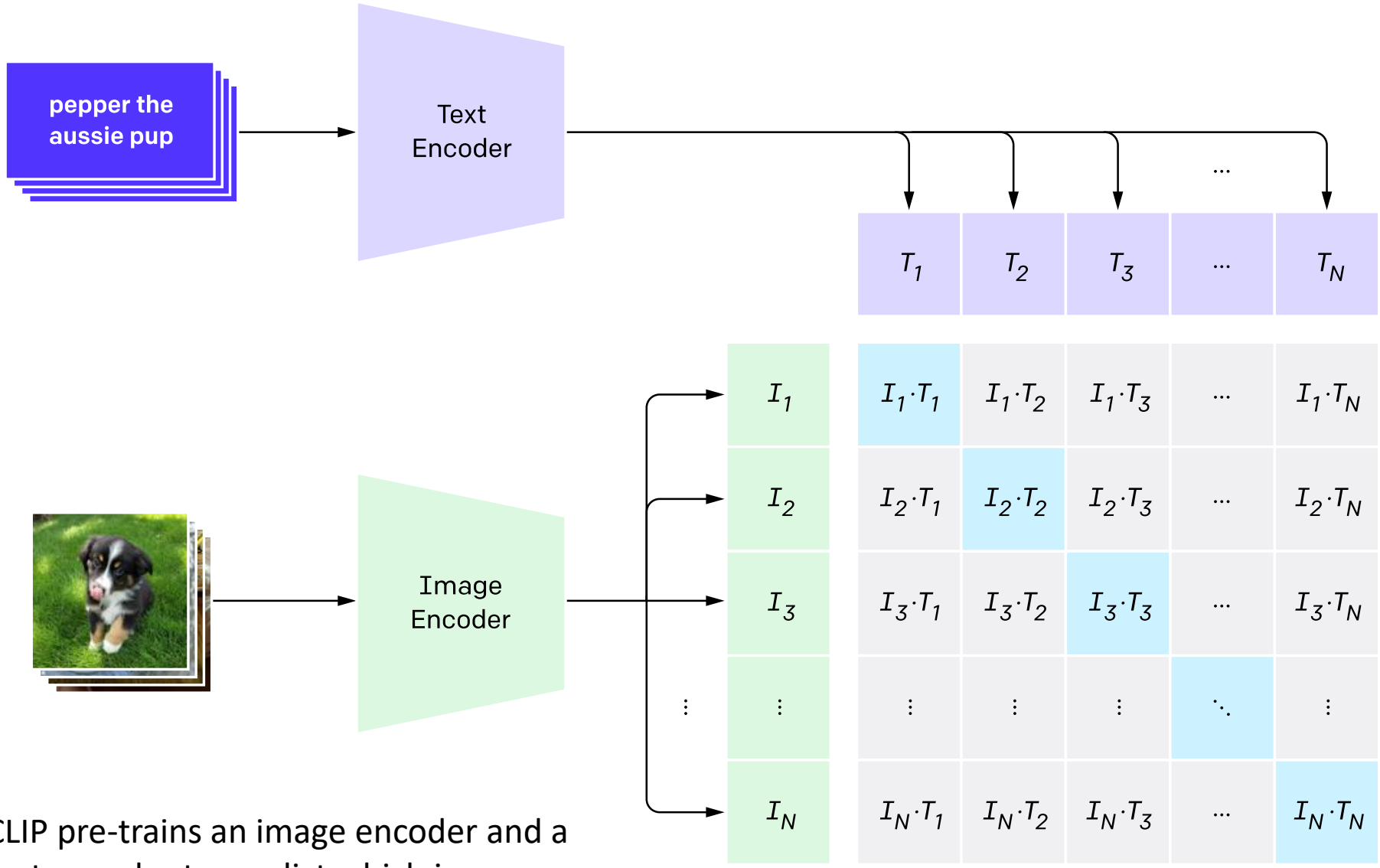


OpenAI CLIP model

- CLIP (Contrastive Language-Image Pre-Training) is a neural network trained on diverse (image, text) pairs.
- It predicts the most relevant text snippet for a given image using natural language instructions.
- Thus, it showcases zero-shot capabilities similar to GPT-2 and GPT-3.

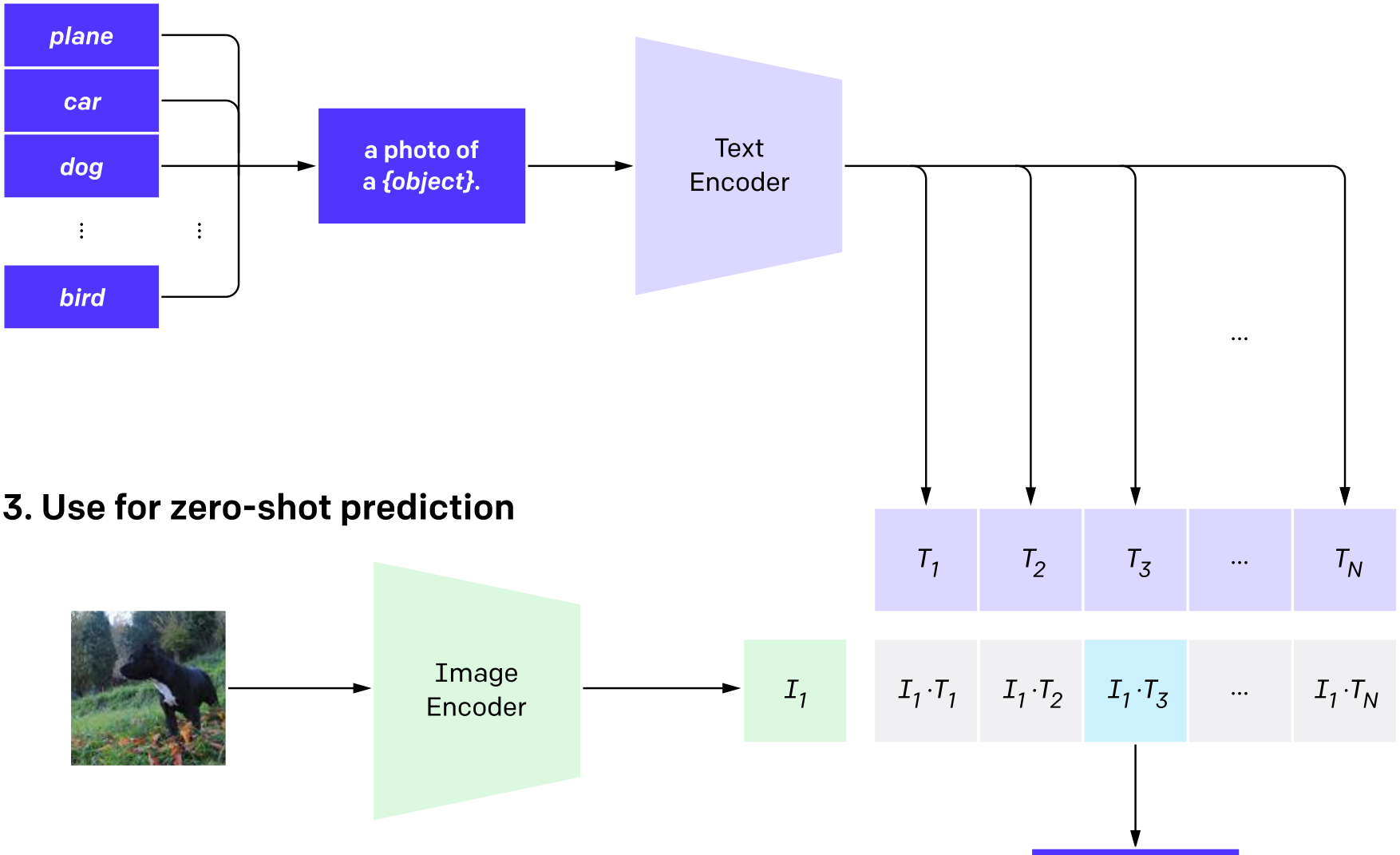
- CLIP remarkably matches the performance of the original ResNet50 on ImageNet in a "zero-shot" setting.
  - That is, CLIP does not rely on the 1.28 million labeled examples traditionally used.

1. Contrastive pre-training



CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in the dataset.

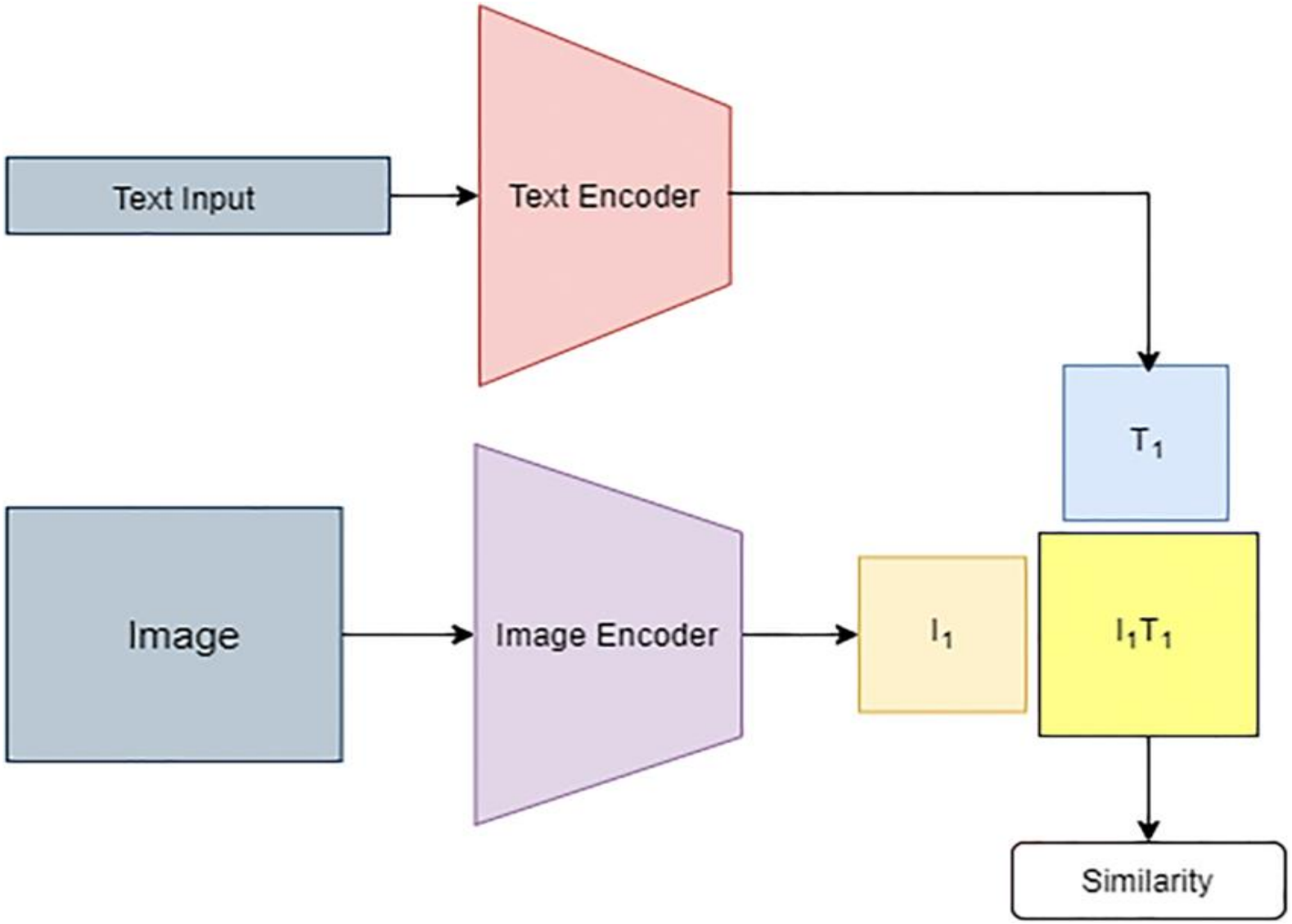
2. Create dataset classifier from label text



3. Use for zero-shot prediction

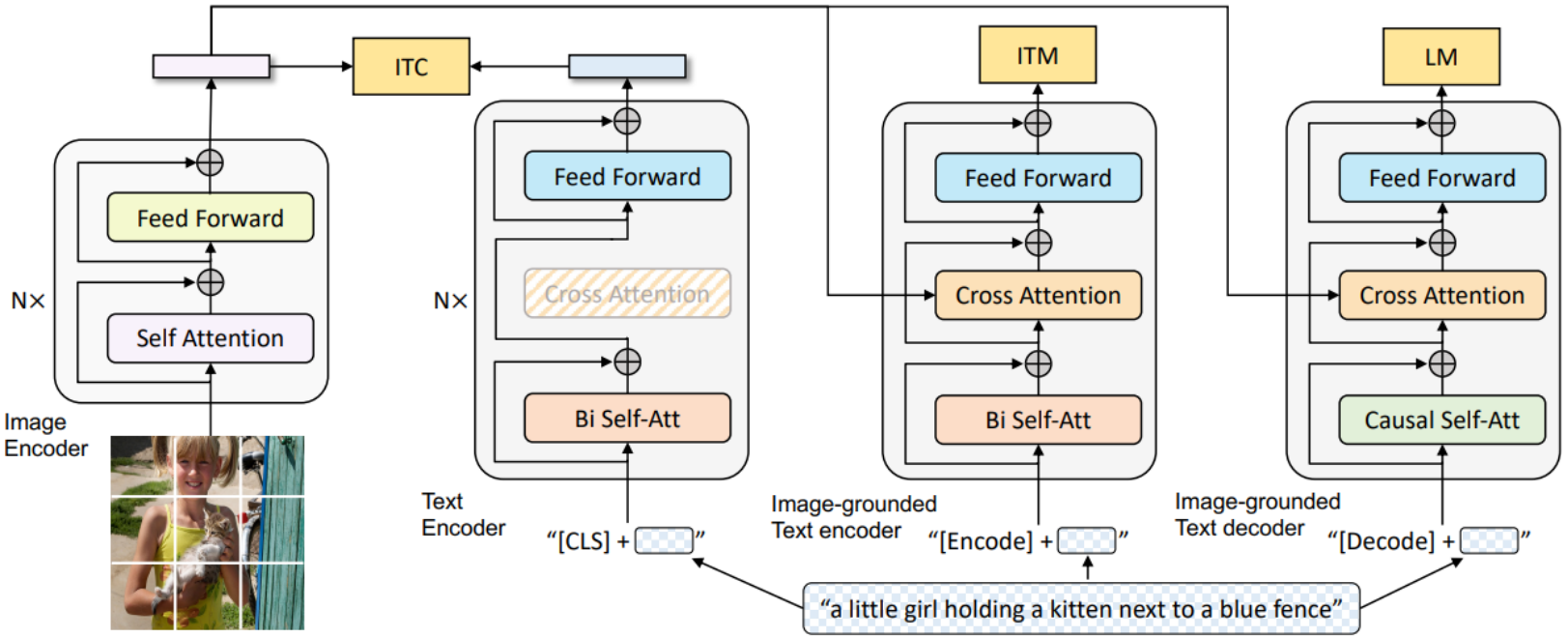
CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as "a photo of a dog" and predict the class of the caption CLIP estimates best pairs with a given image.

OpenAI CLIP: Compute similarity



BLIP model (ICML, 2022)

- BLIP (Bootstrapping Language-Image Pre-training) utilizes the noisy web data by bootstrapping the captions.
- There are a captioner generates synthetic captions and a filter removes the noisy ones.

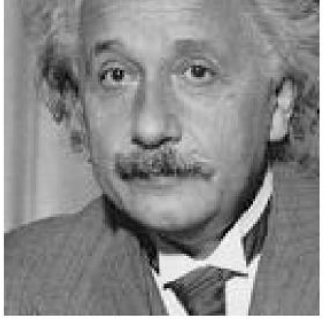


Similarity metrics for vision tasks

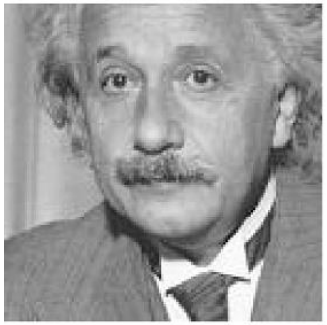
Pixel-based metrics: MAE and MSE

- Let $\mathbf{x}$ be the original image with n pixels x_i. Similarly, let $\mathbf{y}$ be the image after inpainting (with the same size) with pixels y_i.
- Mean absolute error (MAE): $$\boldsymbol{MAE(x,y) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - y_i\|_1}$$
  - The changes in MAE are linear and therefore intuitive.
- Mean squared error (MSE): $$\boldsymbol{MSE(x,y) = \frac{1}{n}\sum_{i=1}^{n} (x_i - y_i)^2}$$
  - MSE punishes larger errors more than smaller errors.

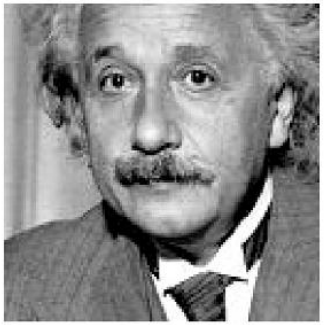
Einstein image altered with different types of distortions



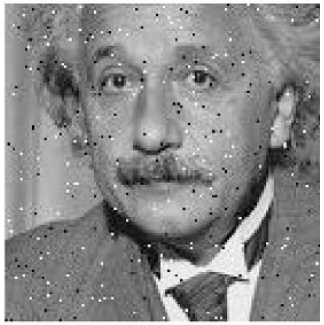
(a)



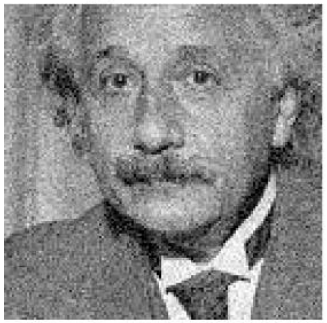
(b) MSE = 309



(c) MSE = 306

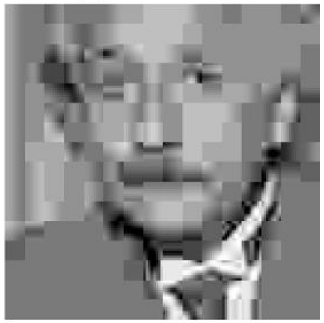


(d) MSE = 313

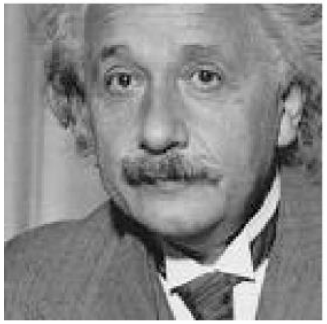


(e) MSE = 309

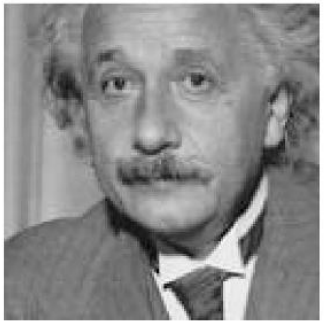
(f) MSE = 308



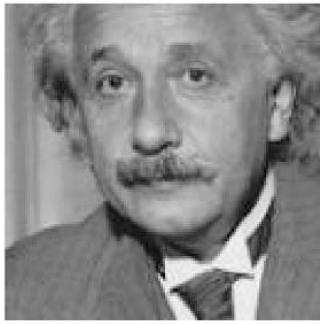
(g) MSE = 309



(h) MSE = 871



(i) MSE = 694



(j) MSE = 590

(a) "original image"
(b) mean luminance shift
(c) a contrast stretch
(d) impulsive noise contamination
(e) white Gaussian noise contamination
(f) Blurring
(g) JPEG compression
(h) a spatial shift (to the left)
(i) spatial scaling (zooming out);
(j) a rotation (counterclockwise).

Note that images (b)–(g) have almost the same MSE values but drastically different visual quality. Also, note that the MSE is highly sensitive to spatial translation, scaling, and rotation [Images (h)–(j)].

Image credit: Modern Image Quality Assessment

Pixel-based metrics: PSNR

- Peak Signal to Noise Ratio (PSNR) is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.
- It is usually derived from MSE, using the decibel (dB) unit.

$$PSNR(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \frac{x_{max}^2}{MSE(\mathbf{x}, \mathbf{y})}$$

  - x_{max} is the maximum signal in the image (e.g., 255 for 8-bit images)
  - Identical images $\rightarrow$ MSE = 0 and PSNR undefined.

- A high-quality image has a high PSNR, but not reverse.

Pixel-based metrics: SSIM

- Structural Similarity Index Measure (SSIM) assesses the structural similarity between two images by comparing their local patches after normalizing luminance and contrast.
- Let $\mathbf{x}$ and $\mathbf{y}$ be the local patches extracted from the same position in the two images, $\mathbf{X}$ and $\mathbf{Y}$, respectively.
- $SSIM(\mathbf{x}, \mathbf{y}) = l(\mathbf{x}, \mathbf{y}) \cdot c(\mathbf{x}, \mathbf{y}) \cdot s(\mathbf{x}, \mathbf{y})$ (luminance, contrast, structure)

$$= \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right) \cdot \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right) \cdot \left(\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}\right)$$

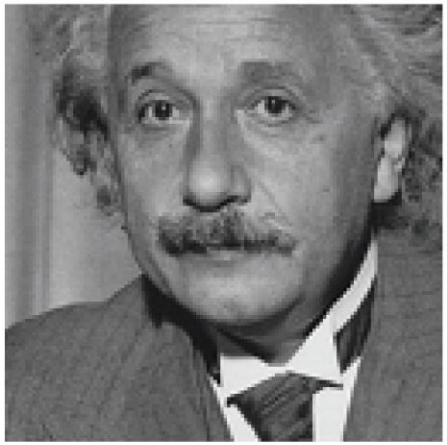
  - μ_x, μ_y, σ_x, and σ_y are means and standard deviations of the local patches $\mathbf{x}$ and $\mathbf{y}$, respectively. C_1, C_2, and C_3 are constants.

Pixel-based metrics: SSIM

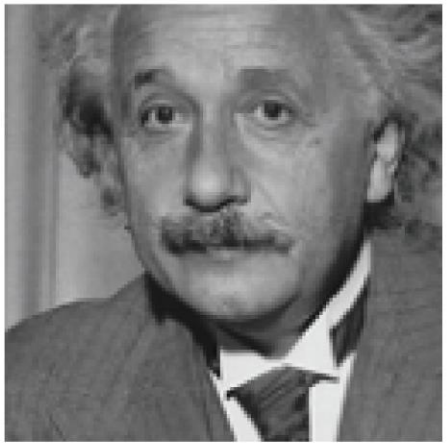
- Patches are extracted by applying a sliding window onto the image, with a stride of one pixel.
- The mean SSIM for the images $\mathbf{X}$ and $\mathbf{Y}$ is defined as

$$SSIM(\mathbf{X}, \mathbf{Y}) = \frac{1}{M} \sum_{j=1}^{M} SSIM(\mathbf{x}_j, \mathbf{y}_j)$$

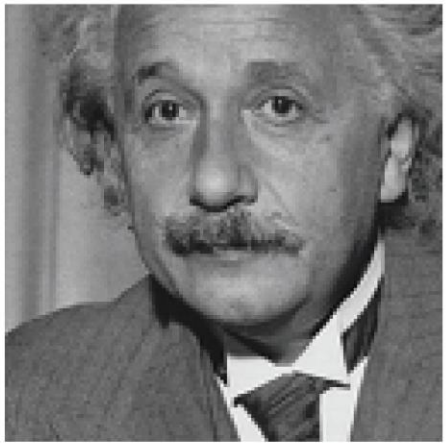
  - M is the total number of patches in the images.



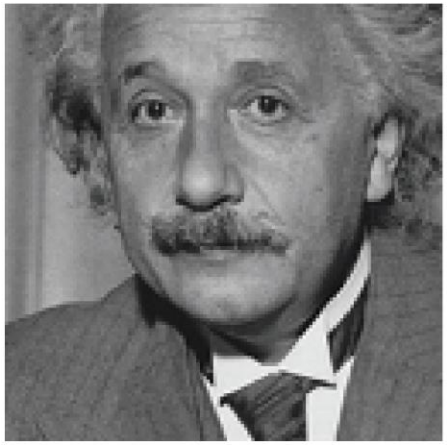
(a) SSIM = 1



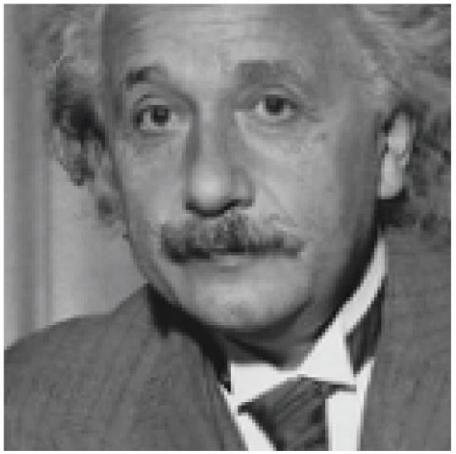
(b) SSIM = 0.505



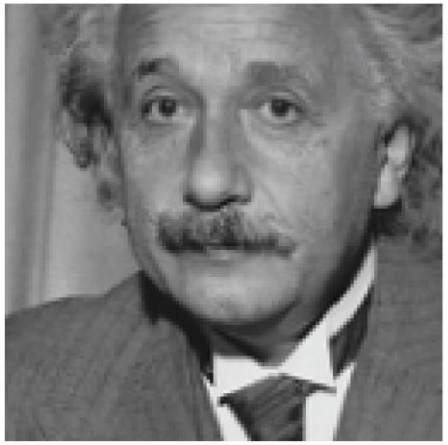
(c) SSIM = 0.404



(d) SSIM = 0.399



(e) SSIM = 0.549



(f) SSIM = 0.551

Some examples to show that SSIM may not estimate the geometric distortions correctly, leading to low SSIM scores. (a) Original image. The other images are processed via: (b) Scaling (shrink), (c) Anti-clockwise rotation, and (d) Clockwise rotation.

Perceptual metrics: FID

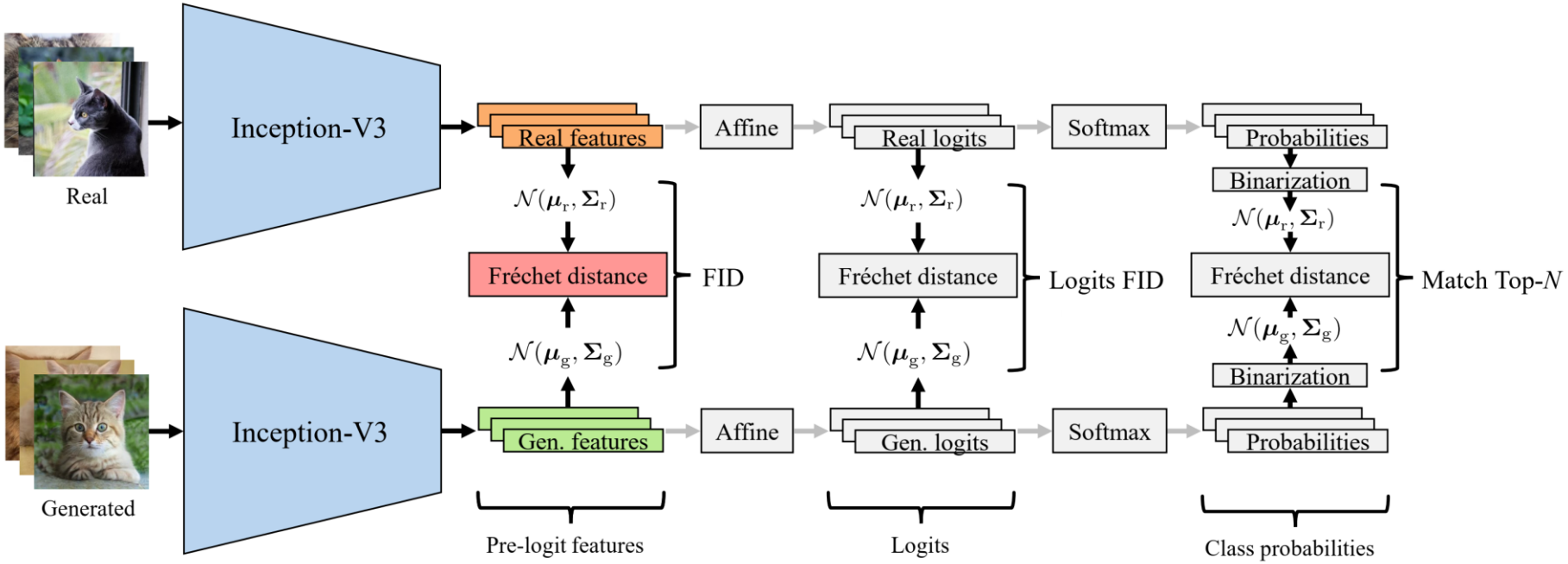
- Fréchet inception distance (FID) assesses the quality of images created by a generative model.
  - introduced in 2017, used in high-resolution StyleGAN 1-2 networks.
- For two multidimensional Gaussian distributions, $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu', \Sigma')$, FID is explicitly solvable as

$$d_F\left(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma')\right)^2 = \|\mu - \mu'\|_2^2 + \mathrm{tr}\left(\Sigma + \Sigma' - 2(\Sigma\Sigma')^{1/2}\right)$$

Perceptual metrics: FID

- Consider a function $\boldsymbol{f}$: $\Omega_X \to \mathbb{R}^n$ and two datasets $S, S' \subset \mathbb{R}^n$.
- Compute $\boldsymbol{f}(S), \boldsymbol{f}(S') \subset \mathbb{R}^n$
- Fit two Gaussian distributions, $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu', \Sigma')$, respectively for $\boldsymbol{f}(S)$ and $\boldsymbol{f}(S')$.
- Return $d_F\left(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma')\right)^2$

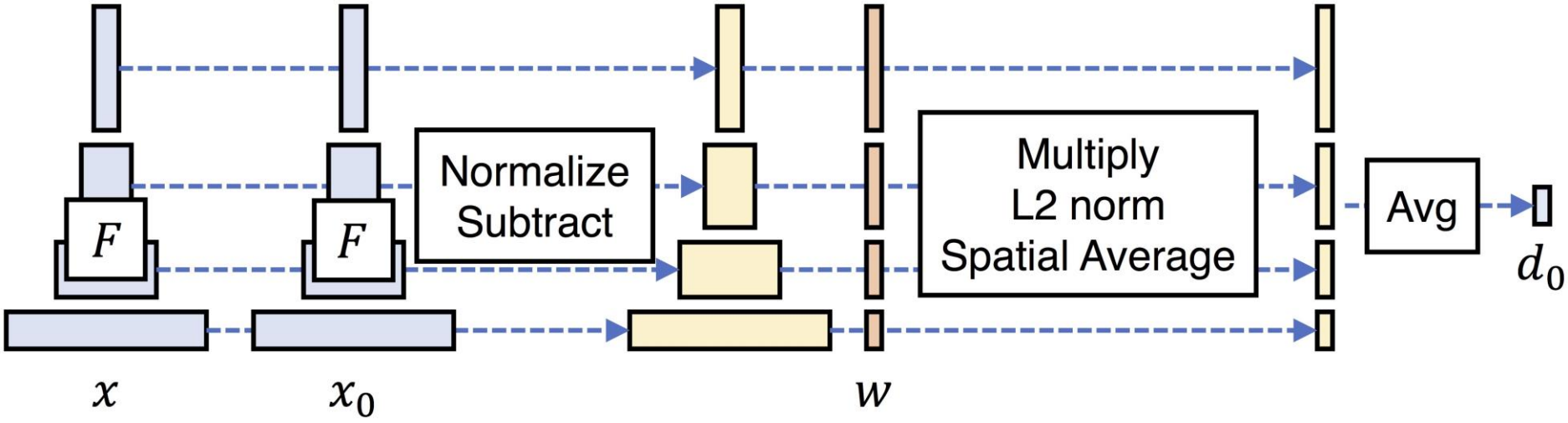
Perceptual metrics: FID



Overview of the Fréchet Inception Distance (FID). First, the real and generated images are separately passed through a pre-trained network, typically the Inception-V3, to produce two sets of feature vectors. Then, both distributions of features are estimated with multivariate Gaussians, and FID is defined as the Fréchet distance between the two Gaussians.

Kynkäänniemi, Tuomas, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. "The Role of ImageNet Classes in Fréchet Inception Distance". ICLR 2023.

Learned Perceptual Image Patch Similarity



$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} ||w_l \odot (\hat{y}^l_{hw} - \hat{y}^l_{0hw})||_2^2$$

Zhang, Richard, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. "The unreasonable effectiveness of deep features as a perceptual metric." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 586-595. 2018.

References

- Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 3rd edition, 2008. Chapter 3
- Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.
- Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." IEEE Transactions on pattern analysis and machine intelligence 24.7 (2002): 971-987.
- Images are obtained from the above materials and Google


...the end.