ĐẠI HỌC
BÁCH KHOA

25 YEARS ANNIVERSARY
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

IT4142E

Introduction to Data Science

Chapter 8: Applications to Computer Vision

Lecturer:

Muriel VISANI: murielv@soict.hust.edu.vn

Acknowledgements:

Khoat Than
Nguyen Thi Oanh

Department of Information Systems
School of Information and Communication Technology - HUST

Contents of the course

Goals of this chapter

Goal	Description of the goal
M1	**Understand and be able to design and manage the systems which are based on Data Science (DS)**
M1.2	Identify, compare, and categorize the data type and systems in practice
M1.3	Be able to design systems based on DS in their future organizations

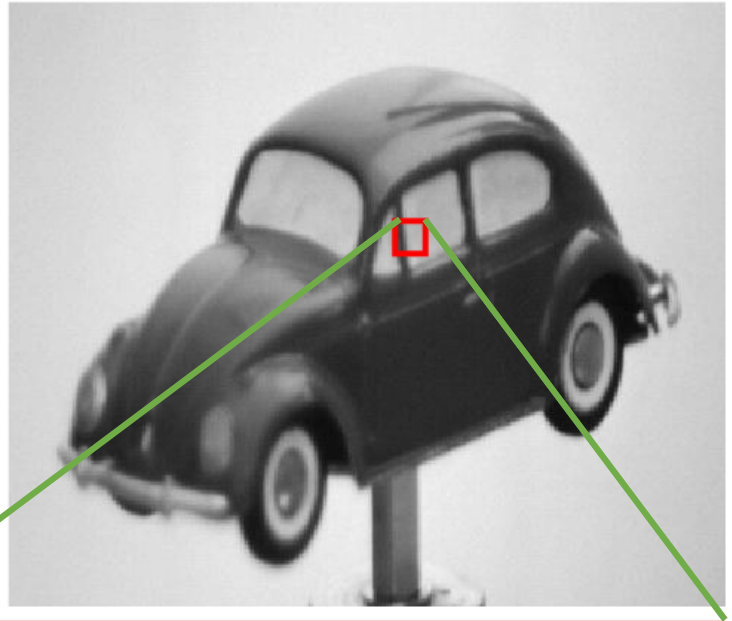
Outline of this chapter

- What is a digital image?
- Basic characteristics of digital images
  – Color, histogram, brightness, contrast, …
  – Basic pre-processing typically applied to digital images
- Computer Vision and Applications
- Some widely used tasks in Computer Vision
- Spatial convolutions
- Supervised classification for computer vision
- Summary
- Some tutorials

What is a digital image?

What is a digital image?

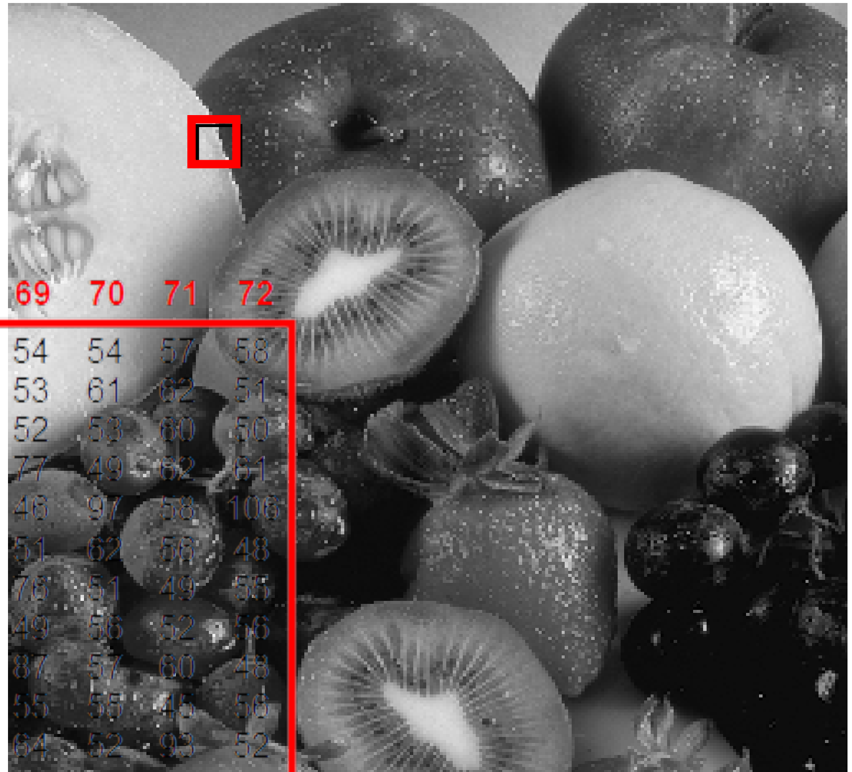
- What can a human see on the picture?
  - A car
- What does the machine see?
  - Image is a matrix of pixels
    - Size of the matrix: W x H
  - 1 pixel (gray levels):
    - An intensity value: 0-255
    - Black: 0
    - White: 255



64	60	69	100	149	151	176	182	179
65	62	68	97	145	148	175	183	181
65	66	70	95	142	146	176	185	184
66	66	68	90	135	140	172	184	184
66	64	64	84	129	134	168	181	182
59	63	62	88	130	128	166	185	180
60	62	60	85	127	125	163	183	178
62	62	58	81	122	120	160	181	176
63	64	58	78	118	117	159	180	176

What is a digital image?

- For an image I
  - Index (0,0): Top left corner
  - I(x,y): intensity of pixel at the position (x,y)



y \ x =	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
y = 41	210	209	204	202	197	247	143	71	64	80	84	54	54	57	58
42	206	196	203	197	195	210	207	56	63	58	53	53	61	62	51
43	201	207	192	201	198	213	156	69	65	57	55	52	53	60	50
44	216	206	211	193	202	207	208	57	69	60	55	77	49	62	61
45	221	206	211	194	196	197	220	56	63	60	55	46	97	58	106
46	209	214	224	199	194	193	204	173	64	60	59	51	62	56	48
47	204	212	213	208	191	190	191	214	60	62	66	76	51	49	55
48	214	215	215	207	208	180	172	188	69	72	55	49	56	52	56
49	209	205	214	205	204	196	187	196	86	62	66	87	57	60	48
50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63

Computer Vision *vs.* Image processing

- Image Processing
  - Work with image as a pixel matrix (bitmap image)
  - Input: image ➔ output: image
  - Help human to process / enhance quality / modify images
- Computer Vision
  - Make computers understand the contents of images / video
  - Input: image ➔ output: information about the image

What kind of scene?

Where are the cars?

How far is the building?

…

A great library for basic Computer Vision and Image processing

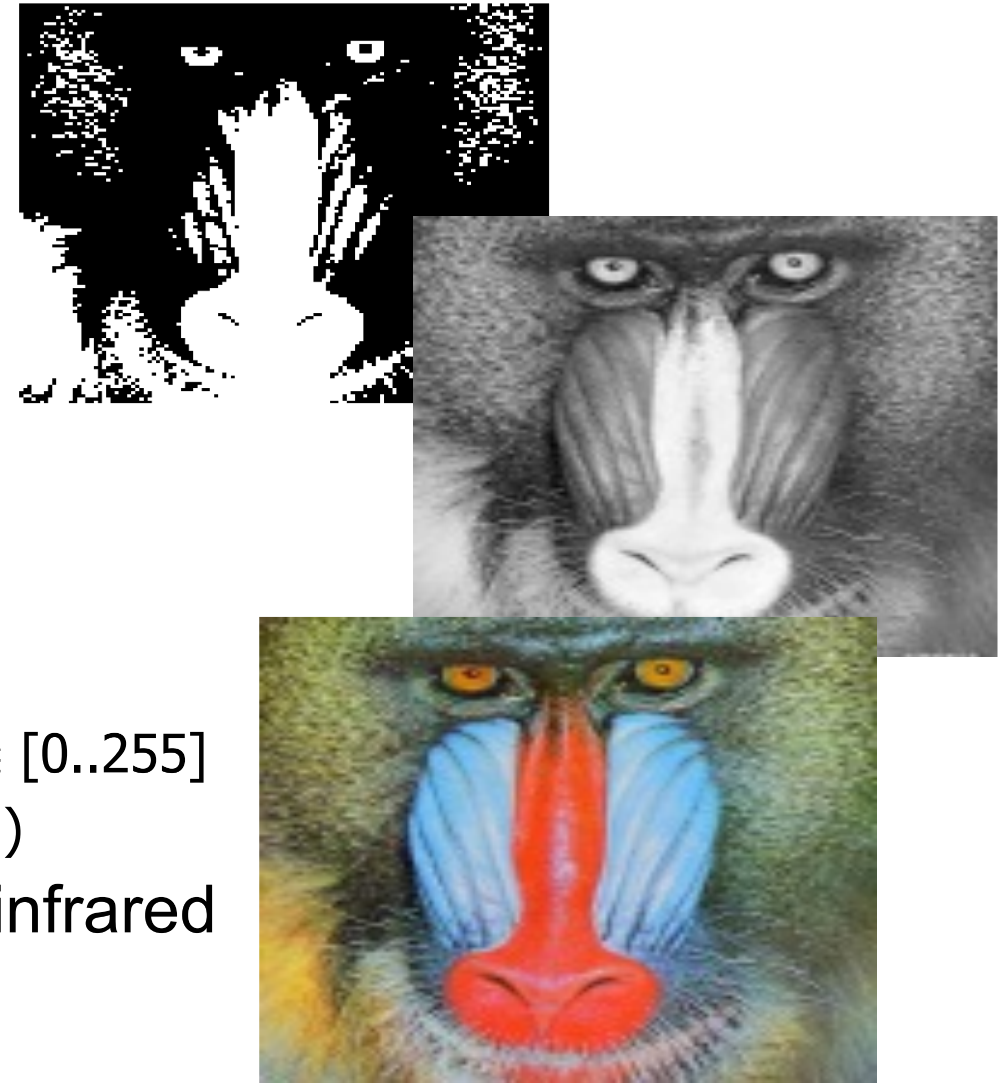
- OpenCV:
  - Can be used with Python / C++…
  - Supports many image processing / computer vision algorithms

Basic characteristics of digital images

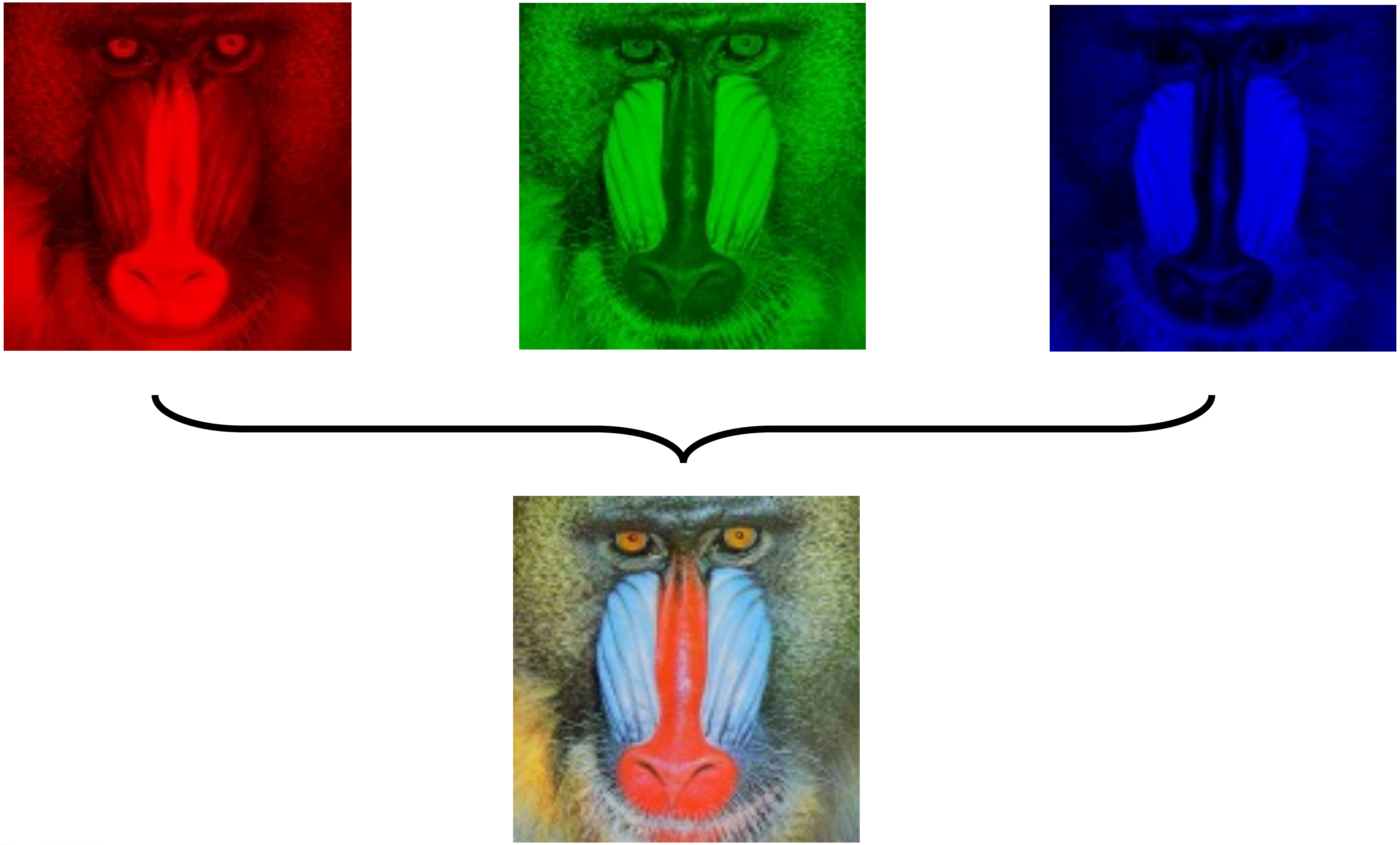
Color, histogram, brightness, contrast, texture, …

Main types of images

- Main type of images
  - Binary image:
    - $I(x,y) \in \{0 , 1\}$
    - 1 pixel: 1 bit
  - Greyscale image:
    - $I(x,y) \in [0..255]$
    - 1 pixel: 8 bits (1 byte)
  - Color image
    - $I_R(x,y), I_G(x,y)\ \ I_B(x,y) \in [0..255]$
    - 1 pixel: 24 bits (3 bytes)
  - Other : multi-spectral, infrared image, depth image,…



Color image in RGB space



Color spaces for color images

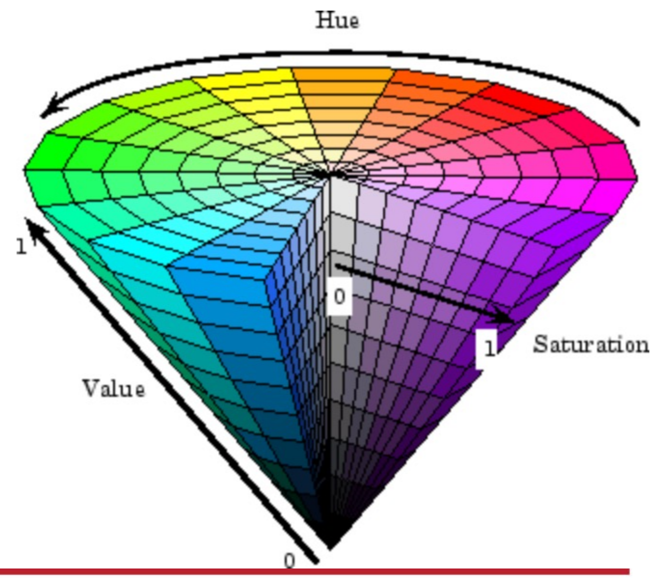
- There exists other color spaces, beyond RGB:
  - Lab, HSV, …
  - Some color spaces are better for some image processing or computer vision applications

- For instance, the Hue-Saturation-Value (HSV) color space is useful for segmentation and recognition

- We identify for a pixel:
  - The pixel *intensity (value)*
  - The pixel *color (hue + saturation)*

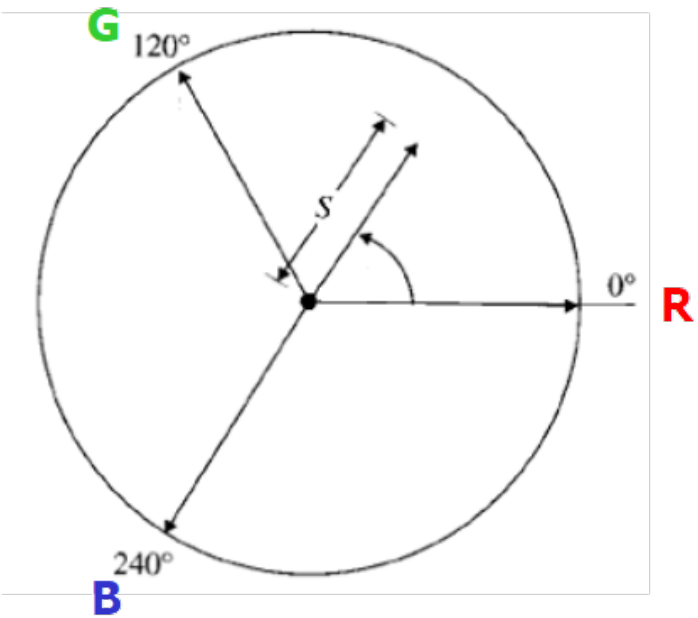
- RGB does not have this seperation

Illustration of the HSV Color Space



Example of HSV (Hue – Saturation- Value)

Attribute	Description
H	Hue, which corresponds to the color's position on a color wheel. H is in the range [0, 1]. As H increases, colors transition from red to orange, yellow, green, cyan, blue, magenta, and finally back to red. Both 0 and 1 indicate red.
S	Saturation, which is the amount of hue or departure from neutral. S is in the range [0, 1]. As S increases, colors vary from unsaturated (shades of gray) to fully saturated (no white component).
V	Value, which is the maximum value among the red, green, and blue components of a specific color. V is in the range [0, 1]. As V increases, the corresponding colors become increasingly brighter.



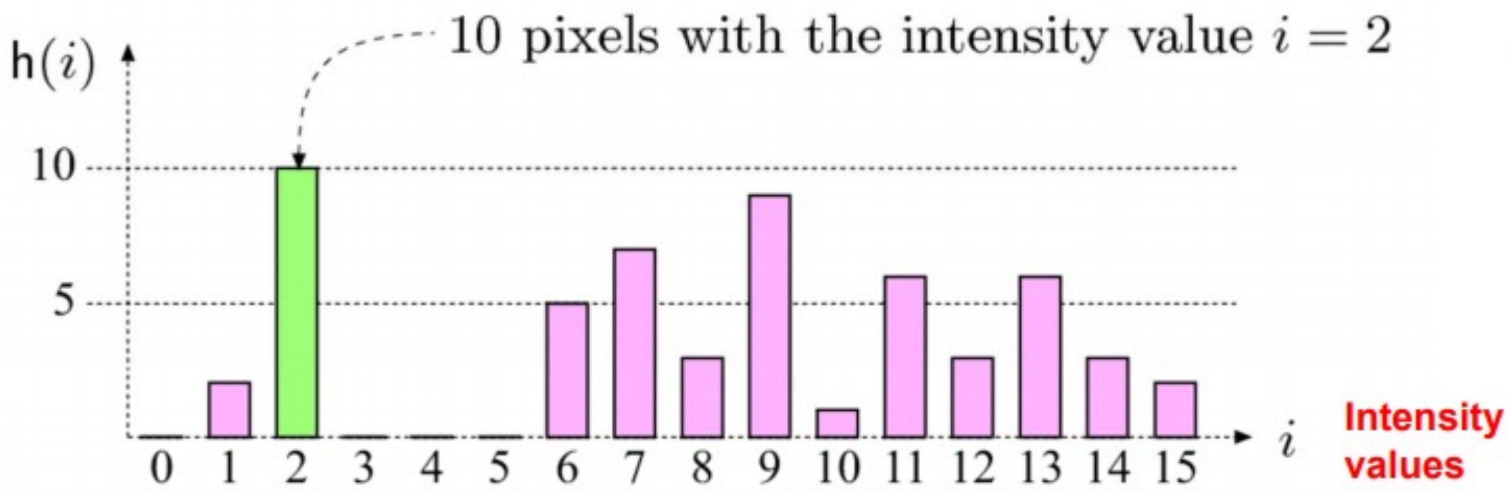
Color spaces for color images

- There exists other color spaces, beyond RGB:
  - Lab, HSV, …
  - Some color spaces are better for some image processing or computer vision applications

- For more information on color spaces:
  - https://www.mathworks.com/help/images/understanding-color-spaces-and-color-space-conversion.html

Image histogram

- Histogram is a graphical representation of the repartition of colours among the pixels of a numeric image.



h(i)	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Image histogram

- Histogram
  - Should be normalized by dividing all elements to total number of pixels in the image

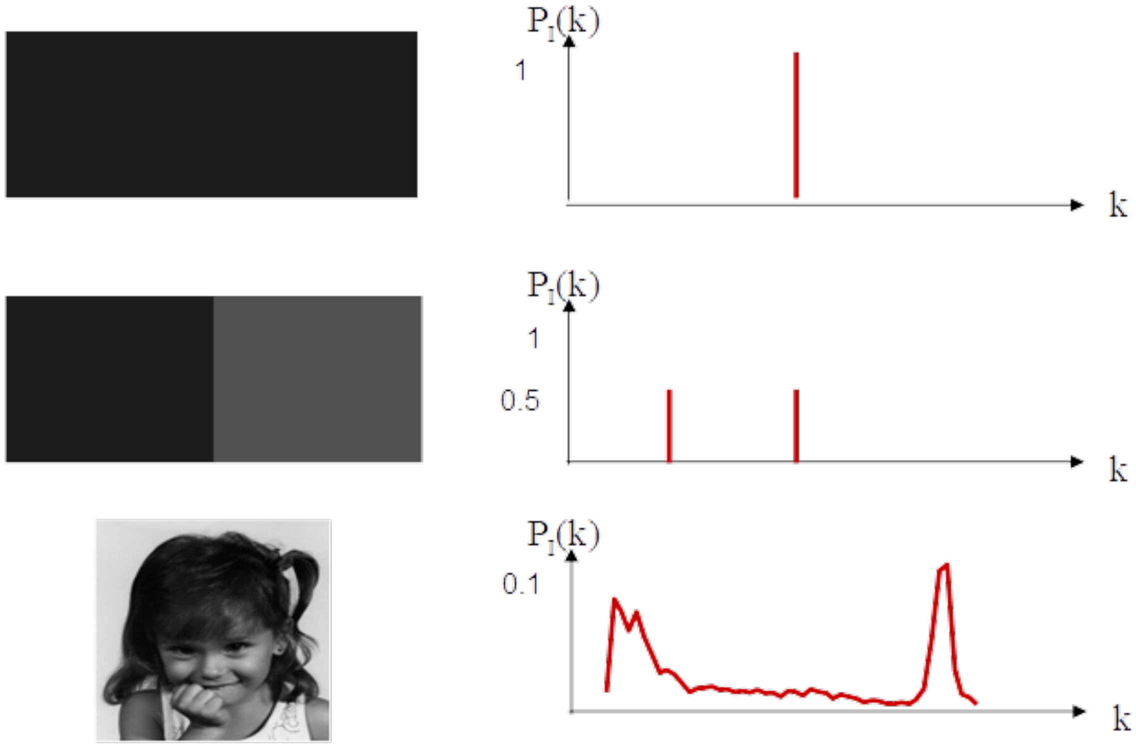
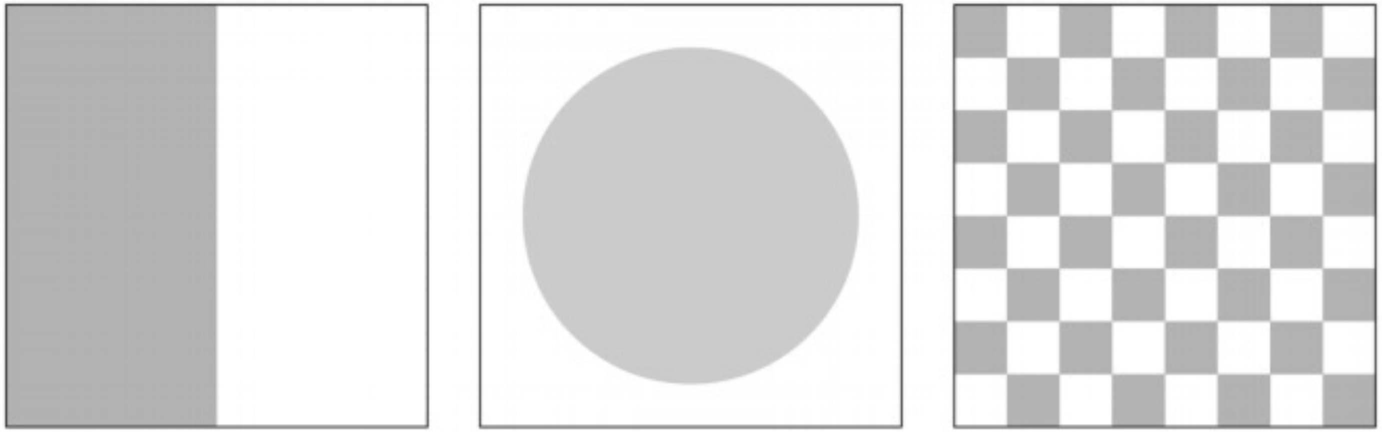


Image dynamic range = [min_value, max_value]

Image histogram

- Histogram
  - Only statistic information
  - No indication about the location of pixel (no spatial information): no information about the **texture** in the image
  - Different image can have the same histogram



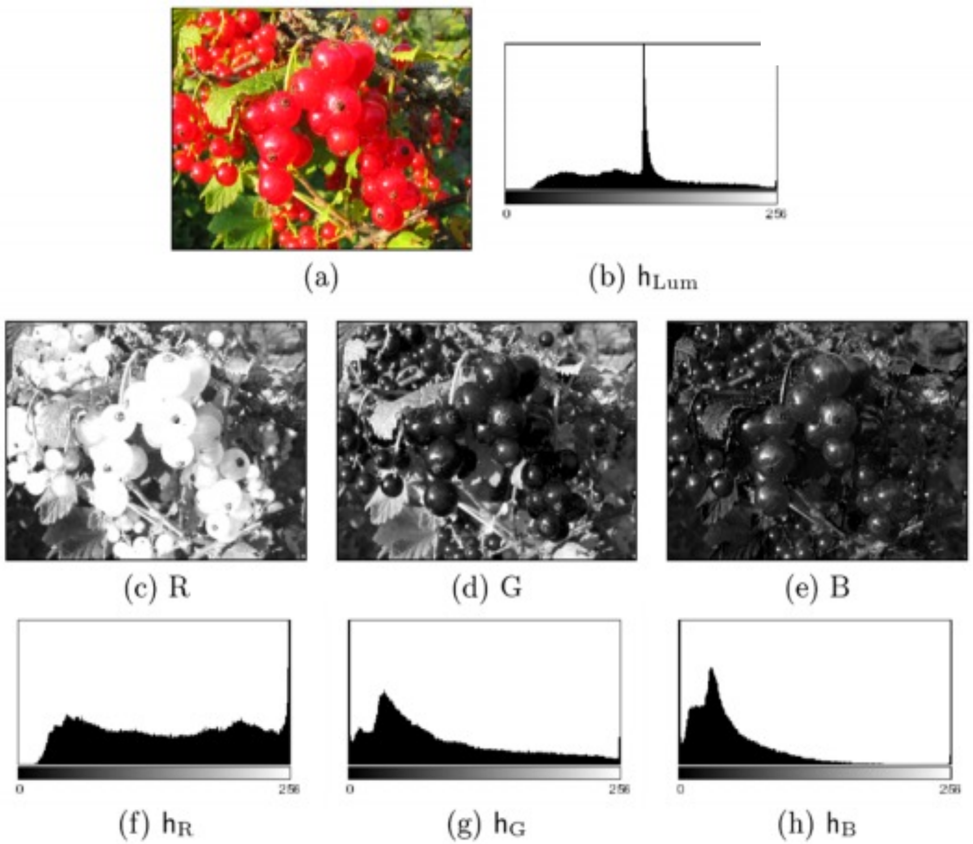
Color Image histogram

- Intensity histogram (b):
  - Convert color image to grayscale

  => Compute histogram of gray scale image
- Individual Color Channel Histograms (f, g, h):

  3 histograms for (R,G,B)
- 3D histogram:

a colour identified by a vector of 3 values, not very commonly used because of its structure size



Source: https://web.cs.wpi.edu/~emmanuel

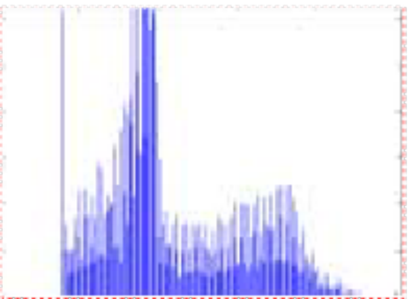
Image Brightness

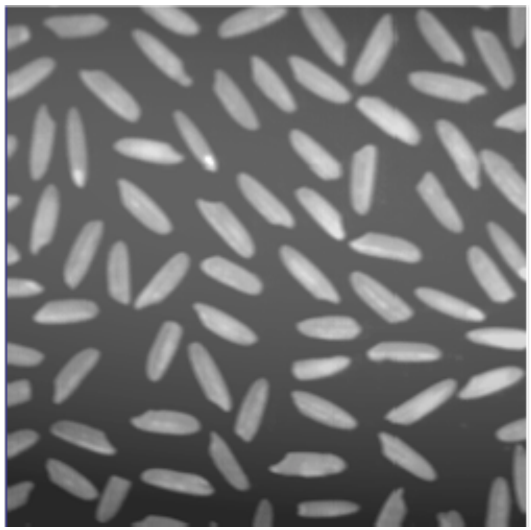
- Brightness of a grayscale image is the average intensity of all pixels in an image
  - refers to the overall lightness or darkness of the image

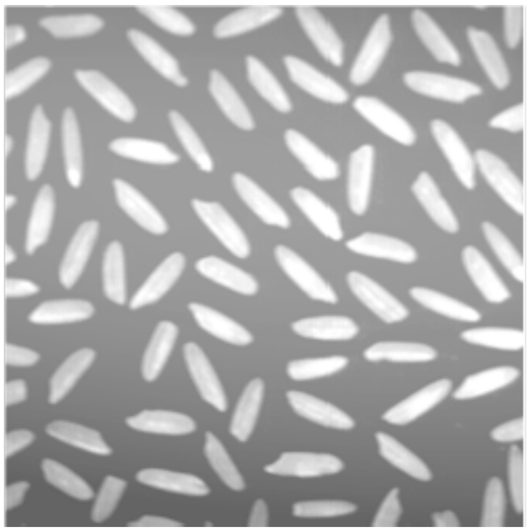
$$B(I) = \frac{1}{wh} \sum_{v=1}^{h} \sum_{u=1}^{w} I(u, v)$$

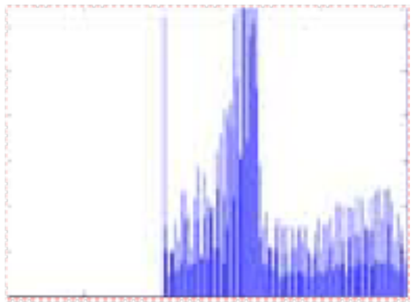
Divide by total number of pixels

Sum up all pixel intensities









Contrast

- The contrast of a grayscale image indicate how easily objects in the image can be distinguished
- Many different equations for contrast exist
  - Standard deviation of intensity values of pixels in the image (N,M size of the image)

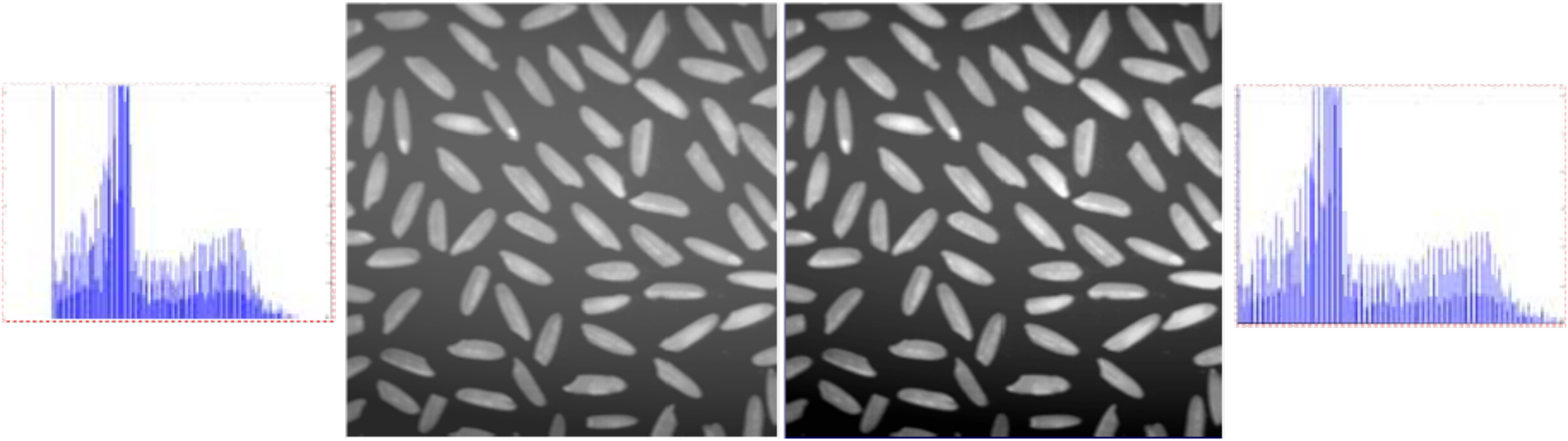
$$C = \sqrt{\frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f(x,y) - Moy)^2}$$

  - Normalized difference between maximum and minimum intensity values

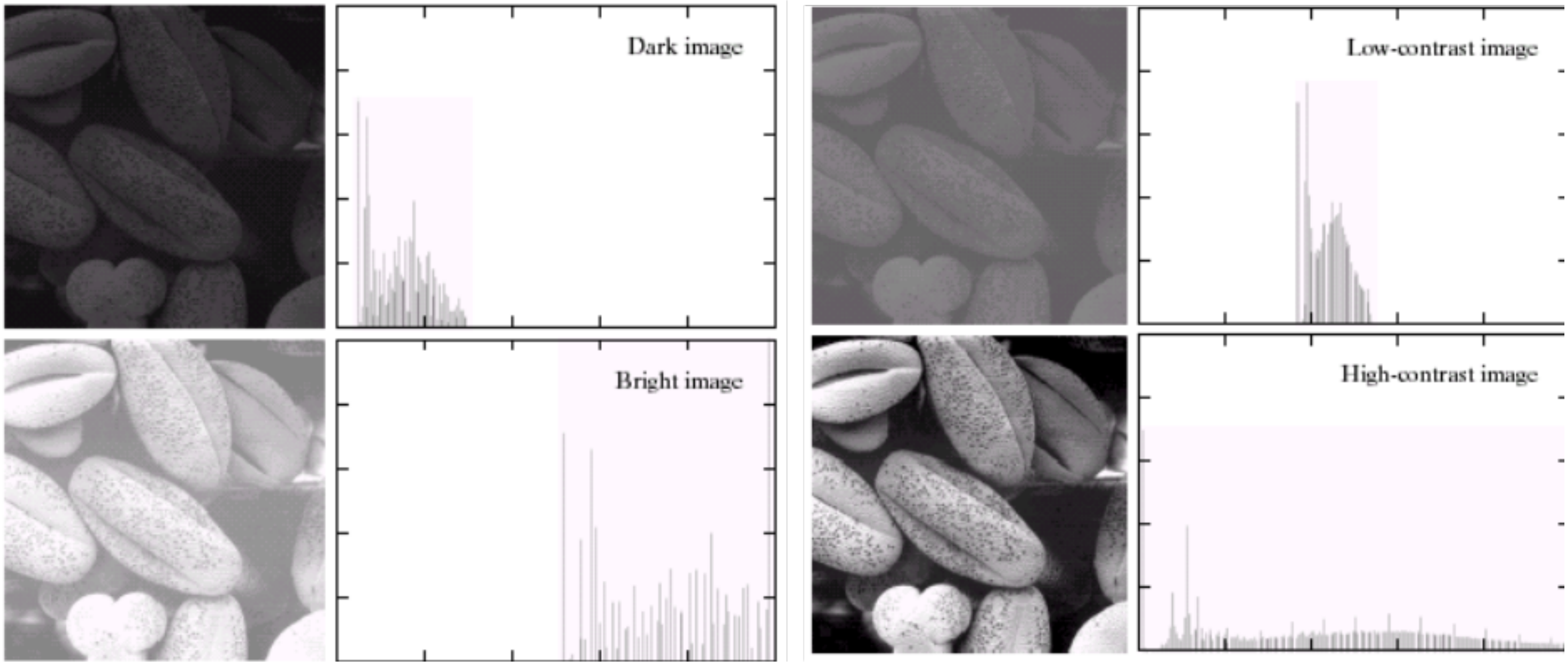
$$C = \frac{\max[f(x,y)] - \min[f(x,y)]}{\max[f(x,y)] + \min[f(x,y)]}$$

Contrast

- Contrast vs histogram
  - More contrast -> “broader” histogram



Examples


Dark image
Bright image
Low-contrast image
High-contrast image

Basic characteristics of digital images

Basic pre-processing typically applied to digital images

Basic image pre-processing

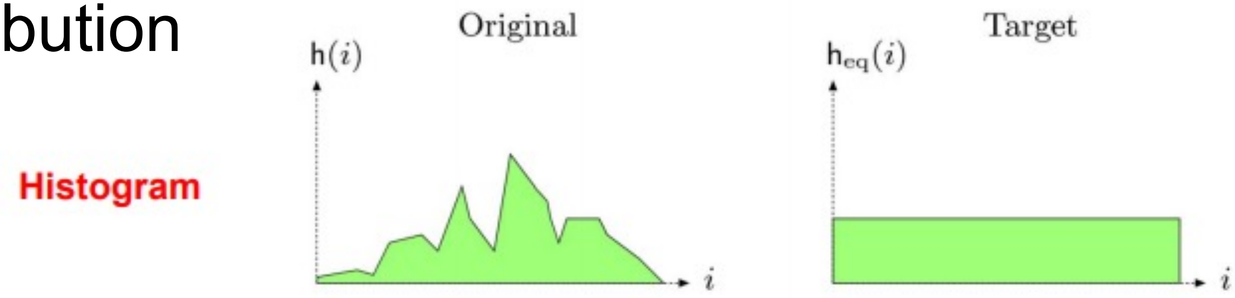
- Just like for most DS applications, one usually needs to pre-process the images before comparing / classifying them
  - Similar to cleaning and normalization of data (see Chapter 3)
- One important pre-processing usually applied to images is contrast enhancement

Basic image pre-processing: contrast enhancement

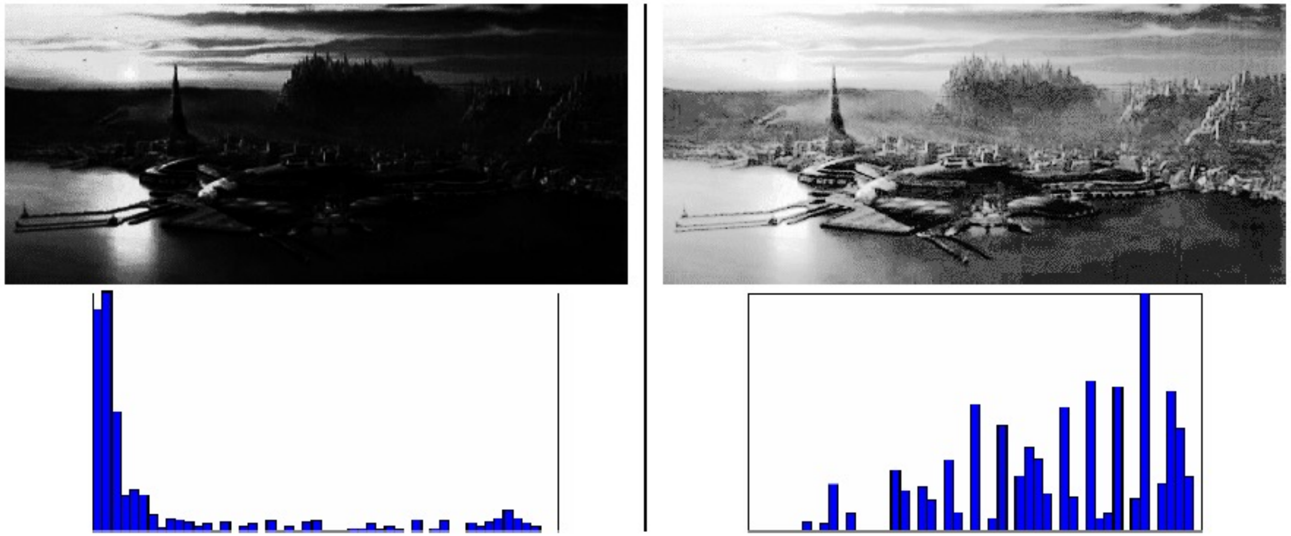
- Modify pixel intensities to obtain higher contrast
- There are several methods:
  - Used in most cameras / smartphones
  - We will illustrate this part with some lines of code using the OpenCV library (available with Python, C++, etc)
  - Linear stretching of intensity range:
  - Non-linear transform (Gama correction)
  - Histogram equalization

Example of histogram equalization

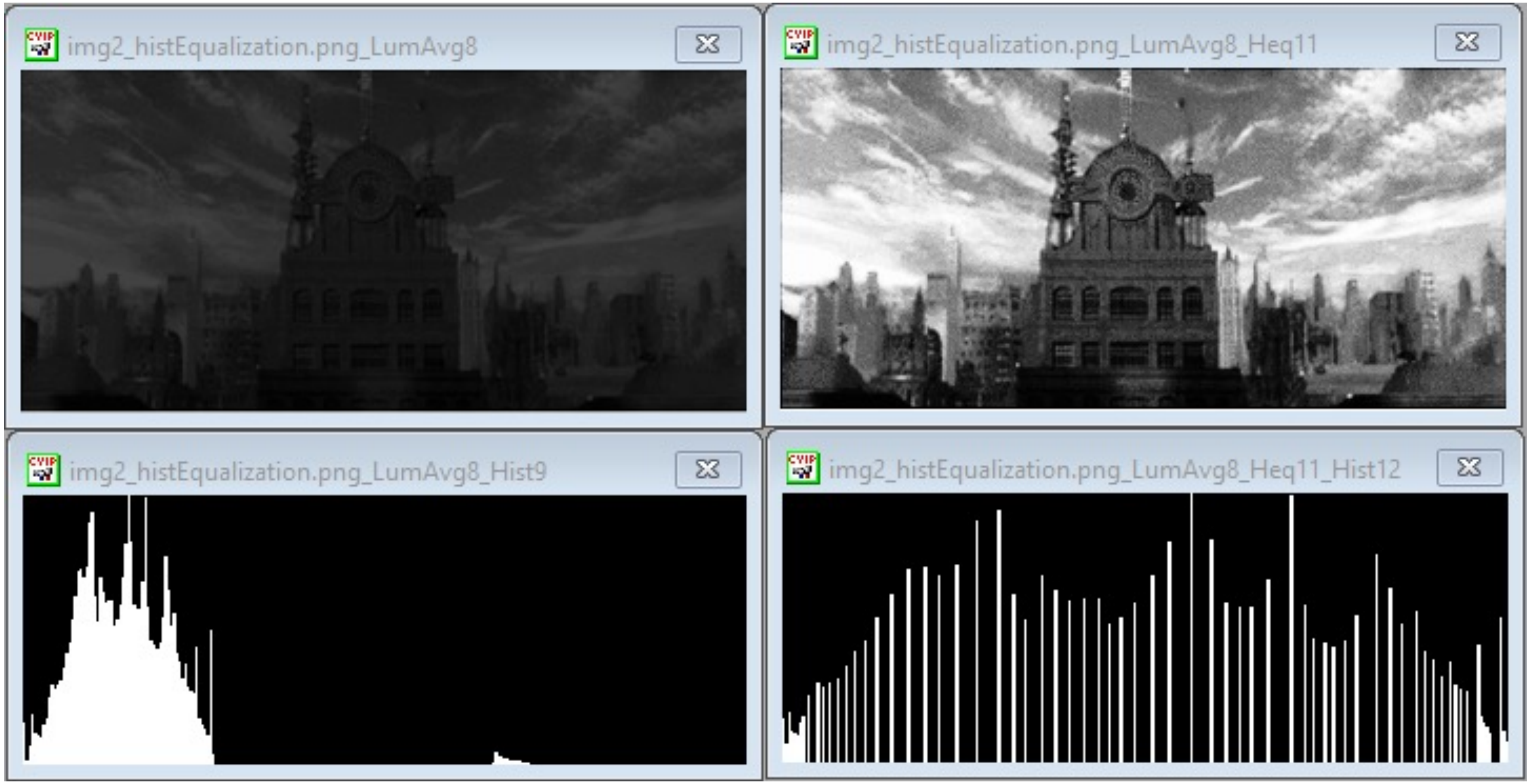
- Change histogram of the image so that it's closer to uniform distribution



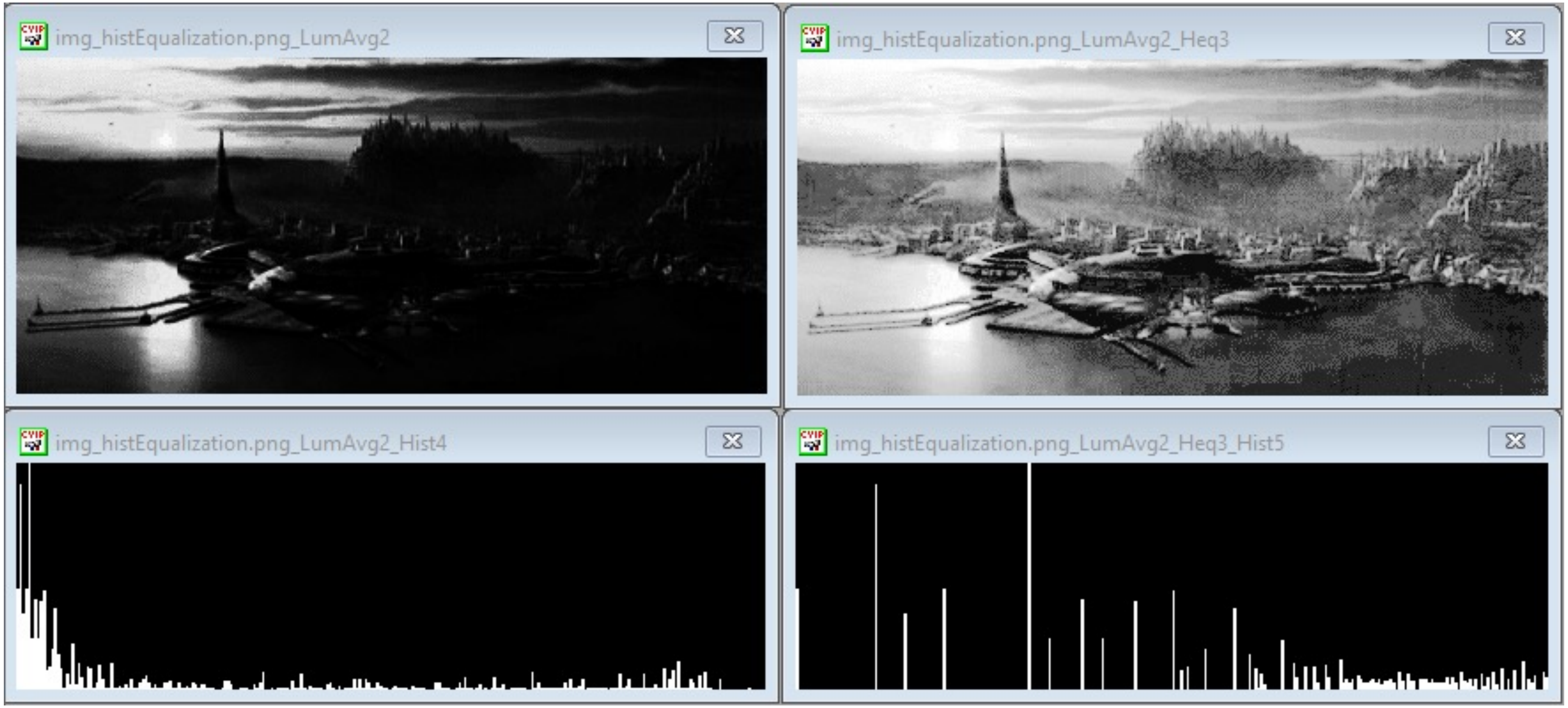
- No parameters. OpenCV:cv2.equalizeHist(img)



Example of histogram equalization



Example of histogram equalization



Convolutions and filters

Spatial convolution

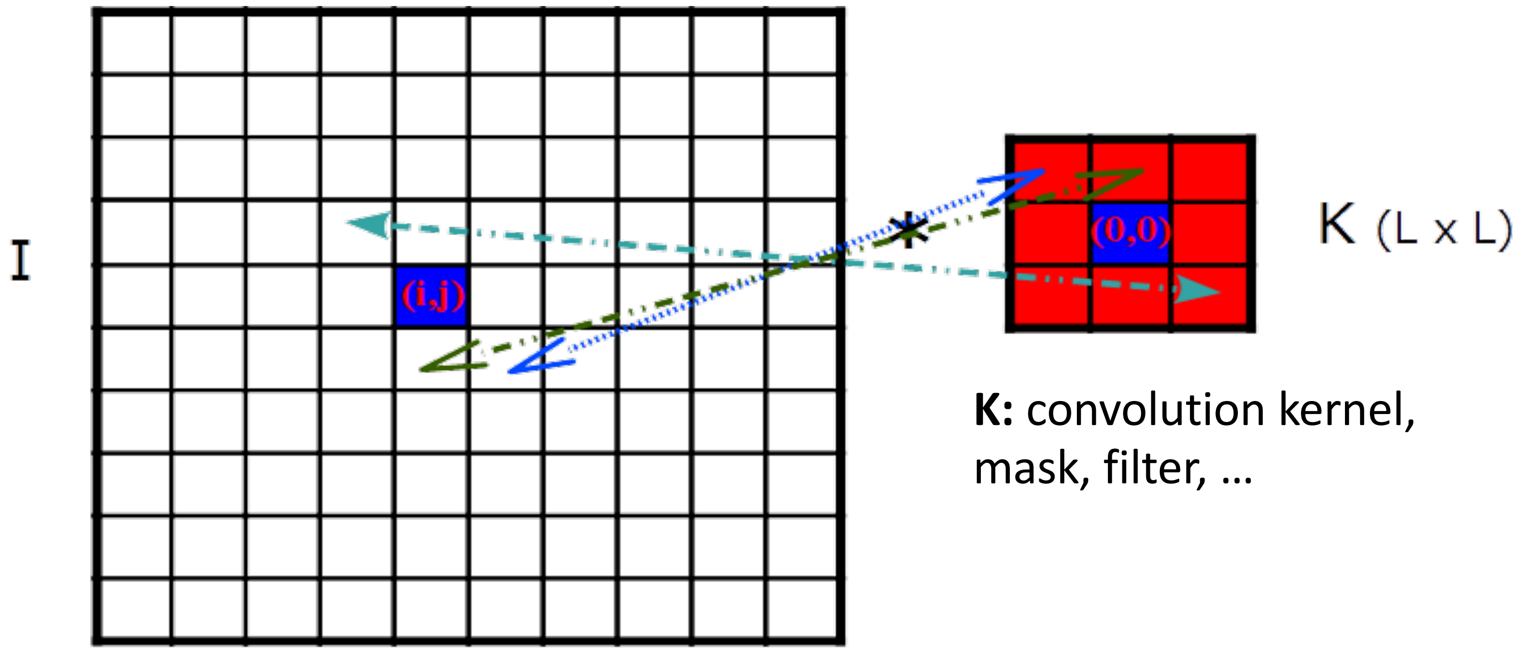
- Image filtering: for each pixel, compute function (kernel K) of local neighbourhood and output a new value
  - Input and output image are typically **the same size** (minus the stride, if no padding)
- Convolution : Linear filtering, function is a weighted sum/difference of pixel values

$$I' = I * K$$

- Spatial convolutions are **great** for images!
  - Can be used for image **pre-processing**: remove the noise, smooth, increase the contrast, etc.
  - Can be used for **computer vision**:
    - To extract information from images:
      - Texture, edges, points of interest, etc.
    - To detect patterns
      - Template matching

Spatial convolution

- New value of a pixel(i,j) is a weighted sum of its neigbors



K: convolution kernel, mask, filter, …

$$I'(i,j) = \sum_{u=-(L-1)/2}^{(L-1)/2} \sum_{v=-(L-1)/2}^{(L-1)/2} I(i-u, j-v)\, K(u,v)$$

Spatial convolution

- The kernel (*a.k.a.* filter*, a.k.a* mask) is applied as a sliding window on the image
  - Example of a 3x3 filter with stride 1
  - Stride = 1 => the filter convolves around the input volume by shifting one unit at a time
  - Kernel in orange, input image in green, convolved image (output) in pink

[https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d]

Some kernels

- 2D spatial convolution
  - is mostly used for image processing or image feature extraction
  - And is also the core block of Convolutional Neural Networks (CNNs)
    - Deep learning for images
- Each kernel has its own effect and is useful for a specific task such as
  - blurring (noise removing),
  - sharpening,
  - edge detection,
  - …..

Some kernels

Original image

$*$

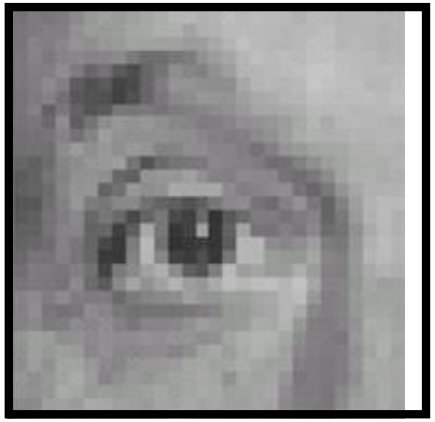
0	0	0
0	1	0
0	0	0

Filtered image
(no change)

Original image

$*$

0	0	0
1	0	0
0	0	0



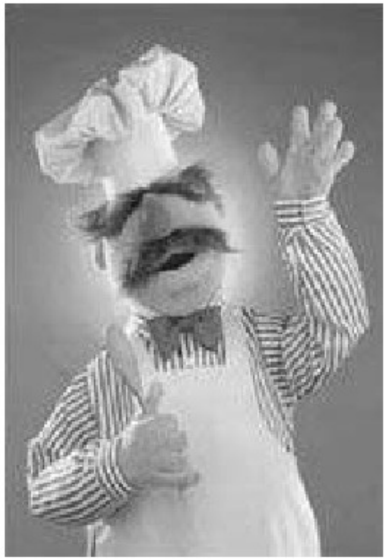
Filtered image
(shifted left by 1 pixel)

Source: David Lowe

Some kernels

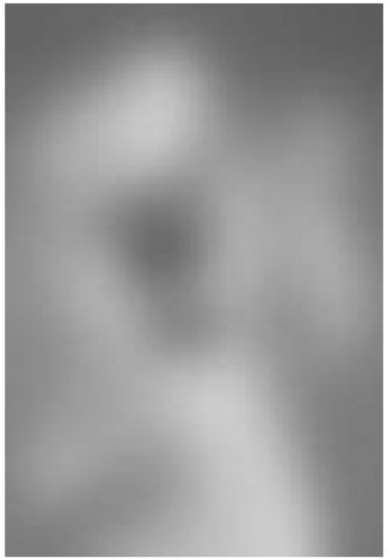
- Box filter (**mean filter**):
  - Replace each pixel with an average of its neigborhood
  - Achieve smoothing effect

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Original image

Filtered image with box size 5x5



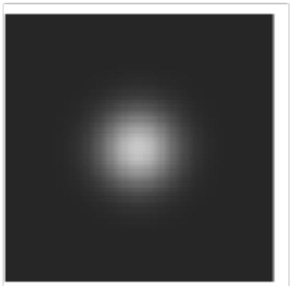
Filtered image with box size 11x11

Some kernels

- Gaussian filter: **Low-pass filter**: Remove high-frequency components from the image (returns a "blurred" image)

Original image

*



Gaussian mask (kernel)

=

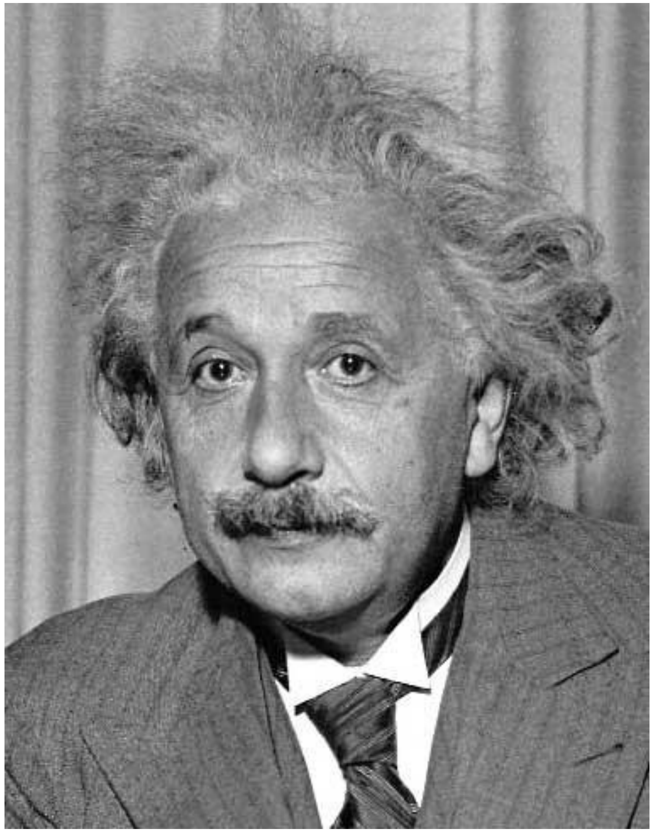
Convolved image -> in that case, the convolved image is Blurred (Gaussian filter)

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

Gaussian filter with size 5 x5 , sigma =1

Some kernels

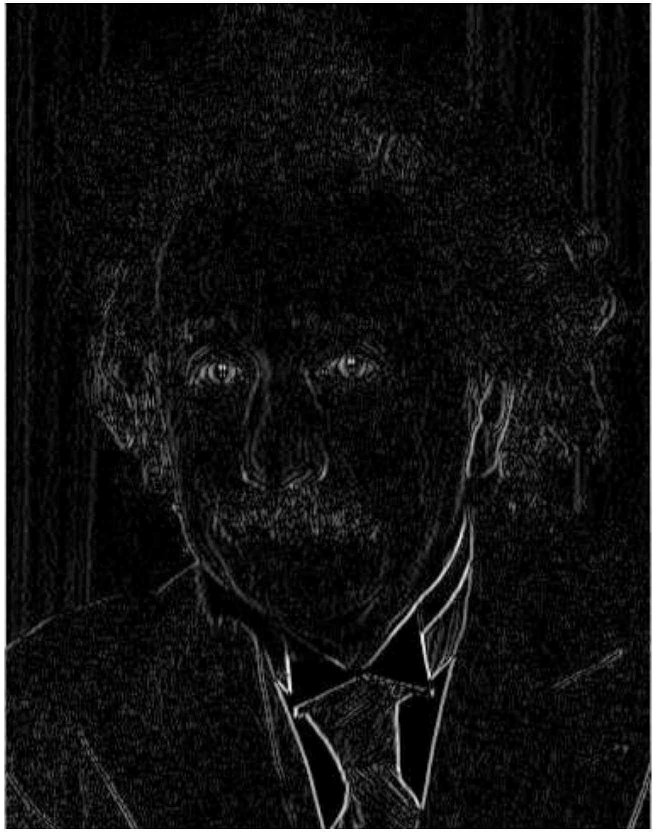
- Sobel for extracting vertical edges



*

-1	0	1
-2	0	2
-1	0	1

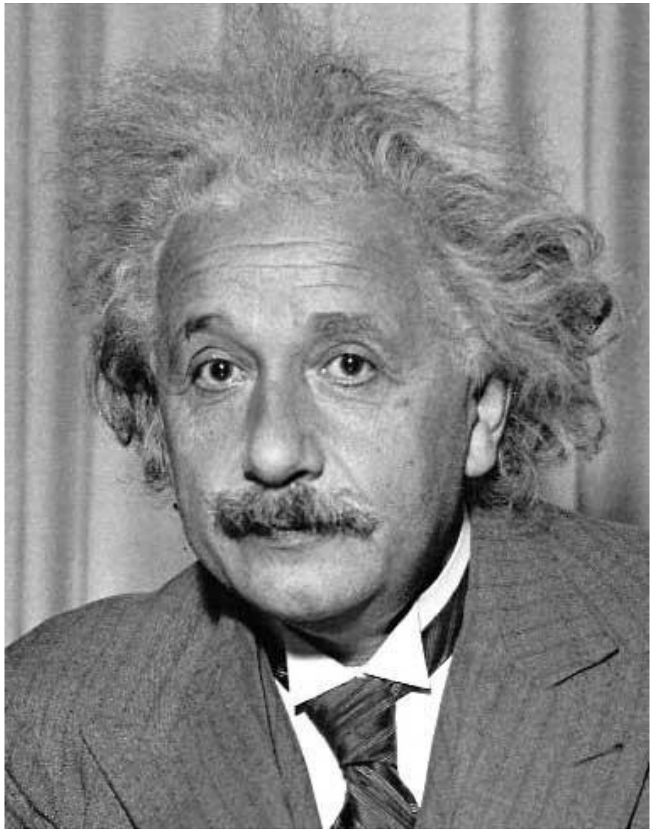
→



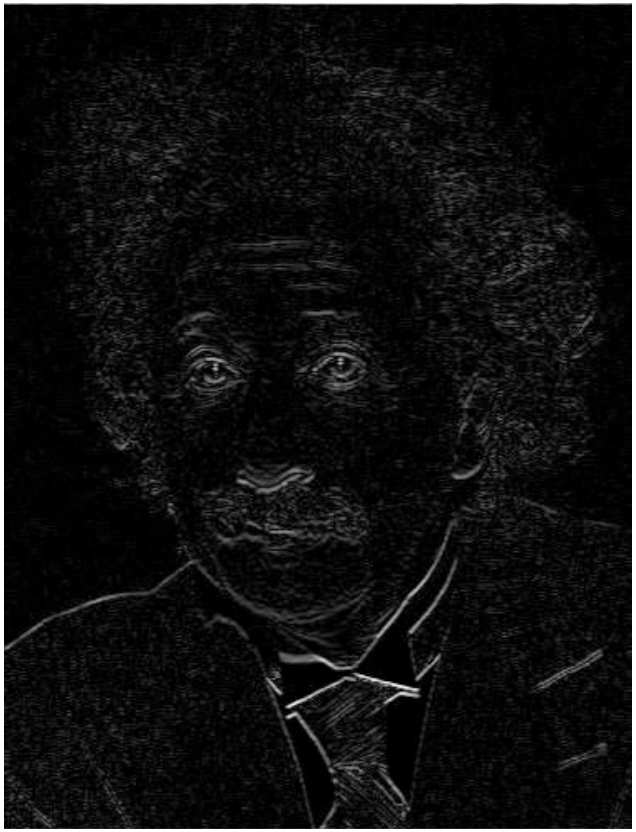
Vertical Edge
(absolute value)

Some kernels

- Sobel for extracting horizontal edges



$$* \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \rightarrow$$

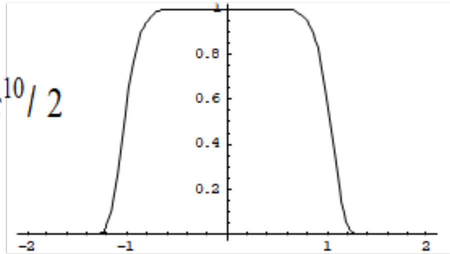


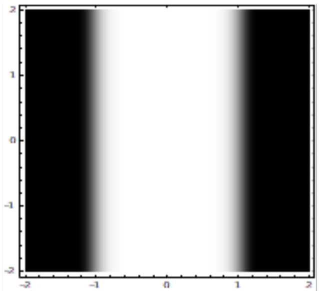
Horizontal Edge
(absolute value)

Edge detection

- Edges are corresponding to:
  - Maximums of the first derivative
  - Zero-crossing in the second derivative

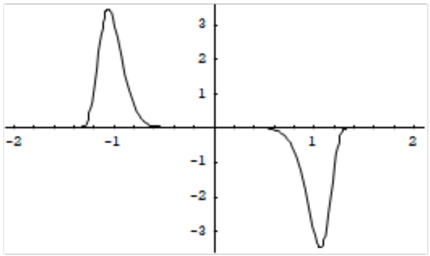
$$f(x,y) = e^{-x^{10}/2}$$

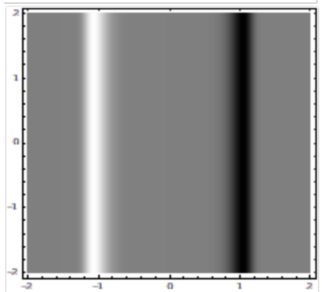




Image

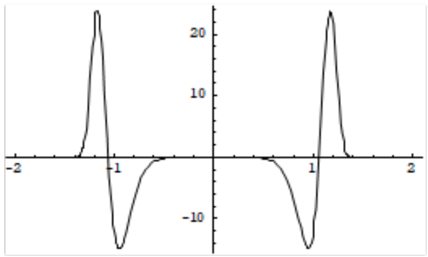
$$\frac{\partial f}{\partial x}$$

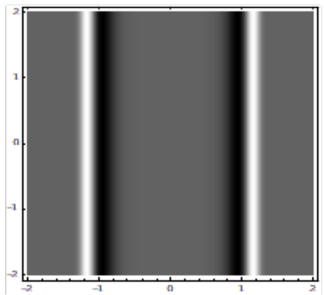




First derivative

$$\frac{\partial^2 f}{\partial x^2}$$





Second derivative

Edge detection using convolutions

- Edge detection can be performed using kernels based on the first derivative or on the second derivative
  - First derivative kernels: **Sobel, Prewitt, Robert**
    - Can **use a threshold** to detect edges (local extrema)
    - Can make several steps to obtain the optimal edge: **Canny detector**
  - Second-order derivative kernels: Laplacian filter, …
  - Everything is implemented in **OpenCV library**

Computer Vision and applications

Computer Vision and Applications

- Images, video are everywhere
- Video, images:
  - Rich information

➔Hot topic, especially

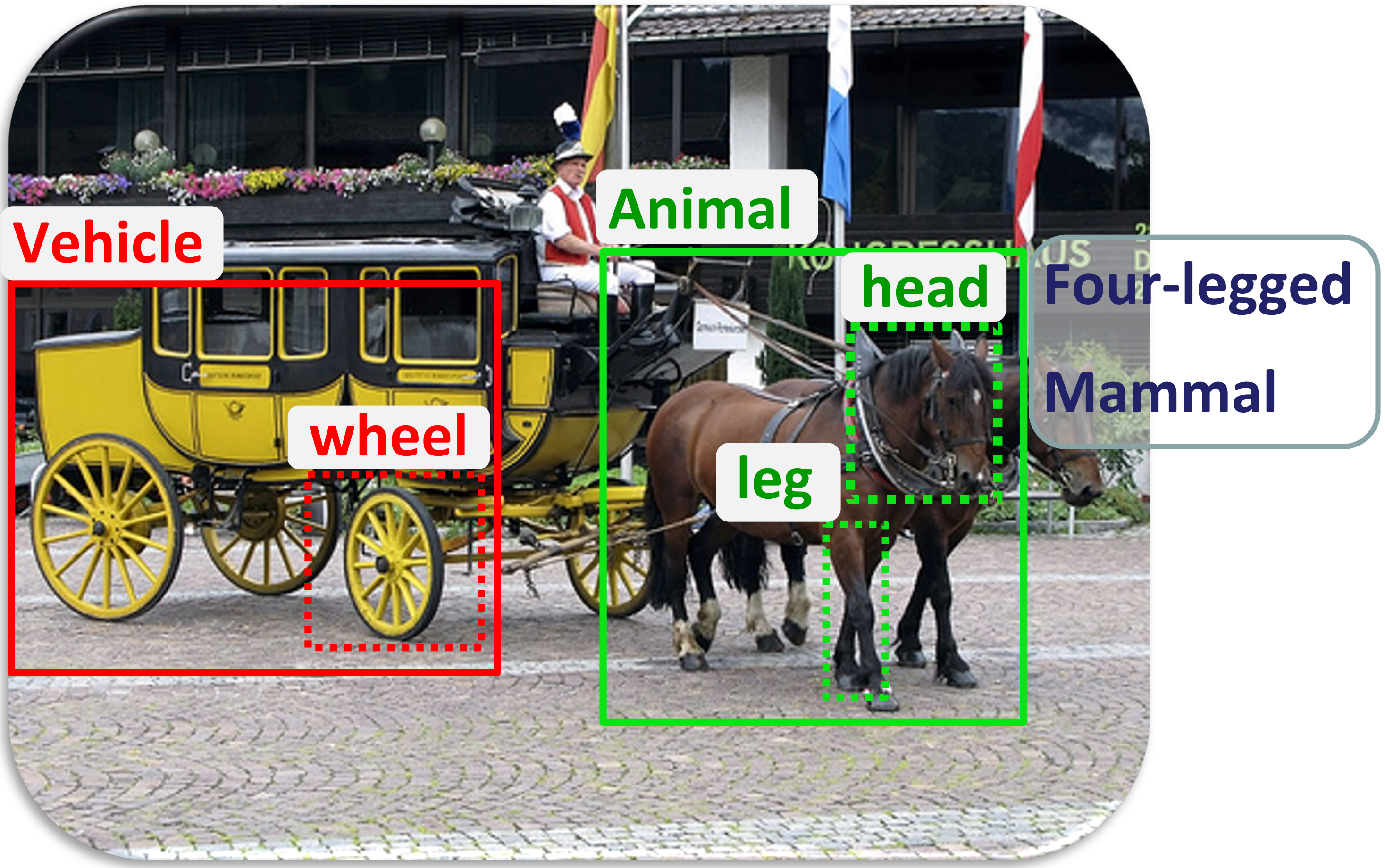
- smart city, smart home,

space, environment,

biometry…

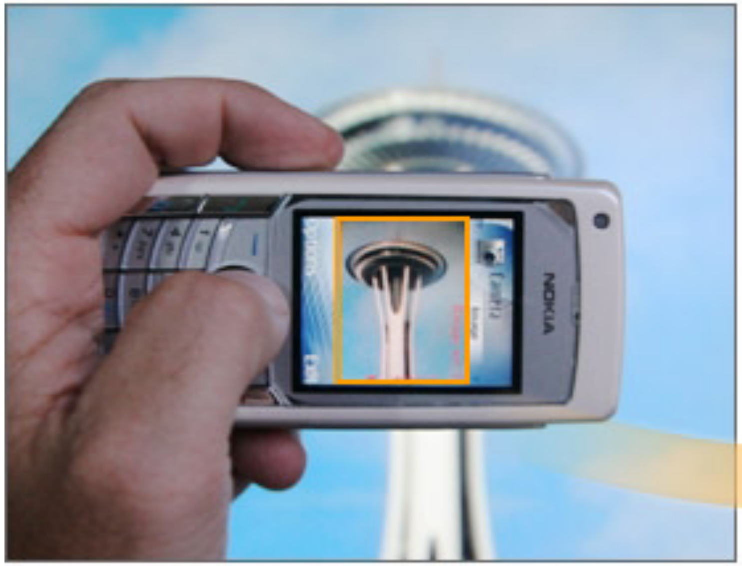
Examples of applications of computer vision

- Object recognition (for understanding the contents of the image)



Examples of applications of computer vision

- Object recognition (on mobile phones)



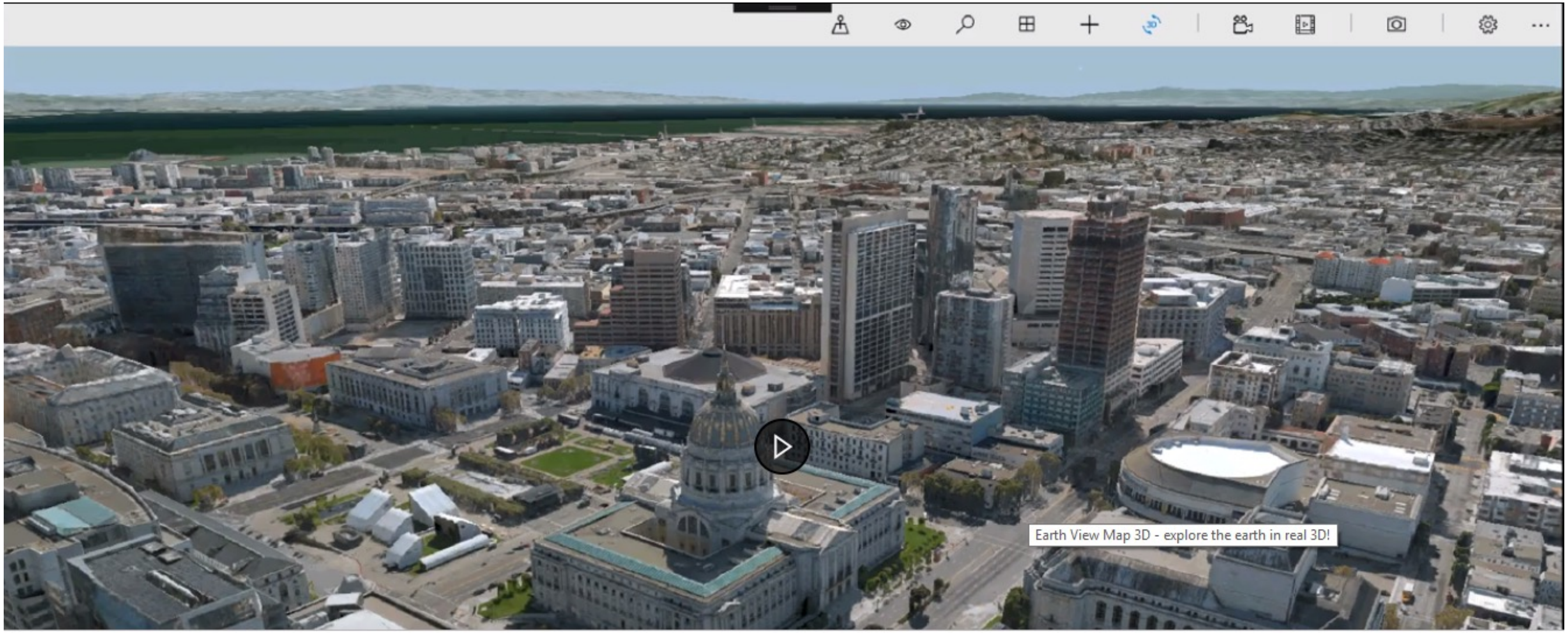
[Point & Find](), [Nokia]()

[Google Goggles]()

Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

Examples of applications of computer vision

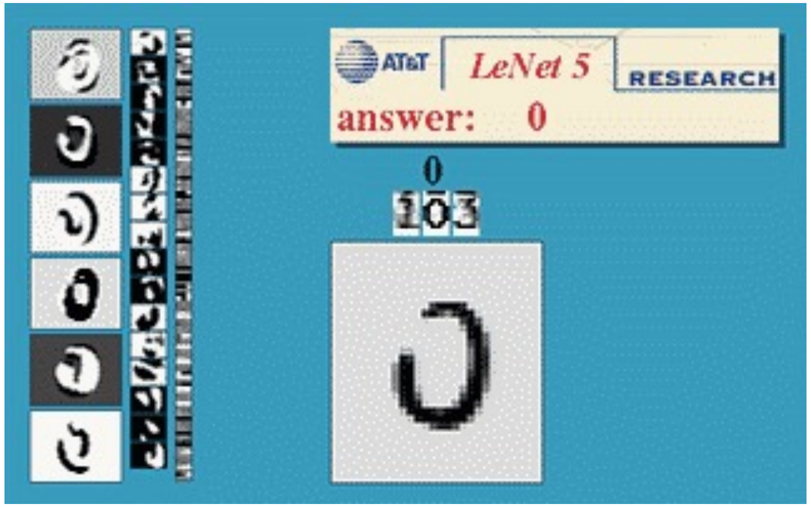
- Earth View, Google earth (3D modeling from lots of 2D images): automatic building generation + hand modeled buildings (Golden Gate bridge or Sydney Opera house)



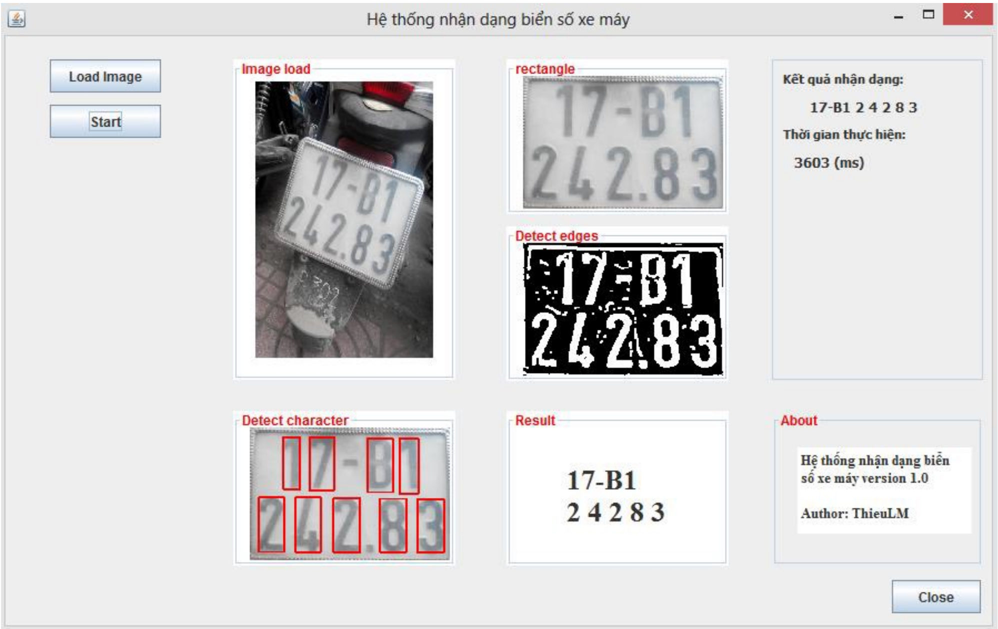
Microsoft's Virtual Earth

Examples of applications of computer vision

- OCR (Optical character recognition)
  - – Technology to convert scanned documents to text
    - For printed, high-quality characters: commercial / Opensource products work very well on most languages



Digit recognition, AT&T labs
http://yann.lecun.com/exdb/lenet/



Licence plate detection and character recognition

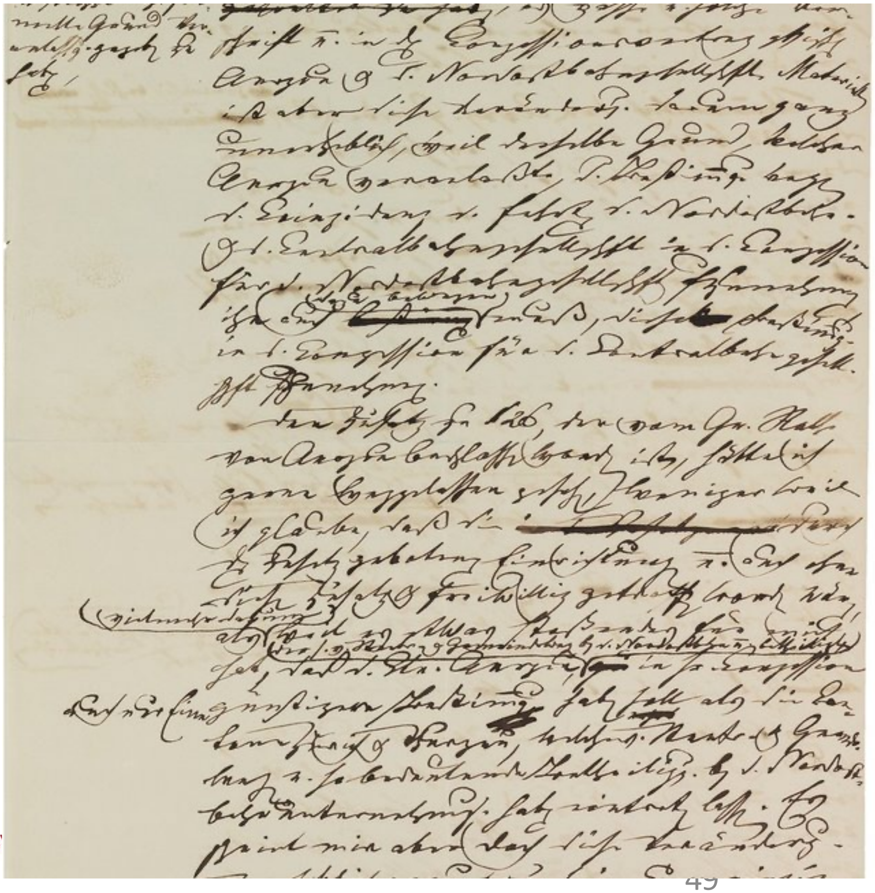
Examples of applications of computer vision

- Handwritten character recognition
  - Still very challenging, especially for old/damaged documents

  ⇒Researchers gather every 2 years at a dedicated conference to present / discuss their advances

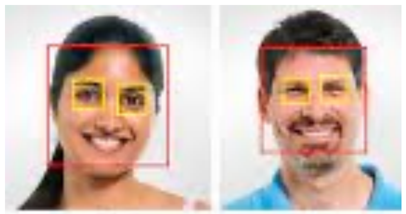
  - International Conference on Frontiers of Handwriting Recognition

    http://icfhr2022.org/



Examples of applications of computer vision

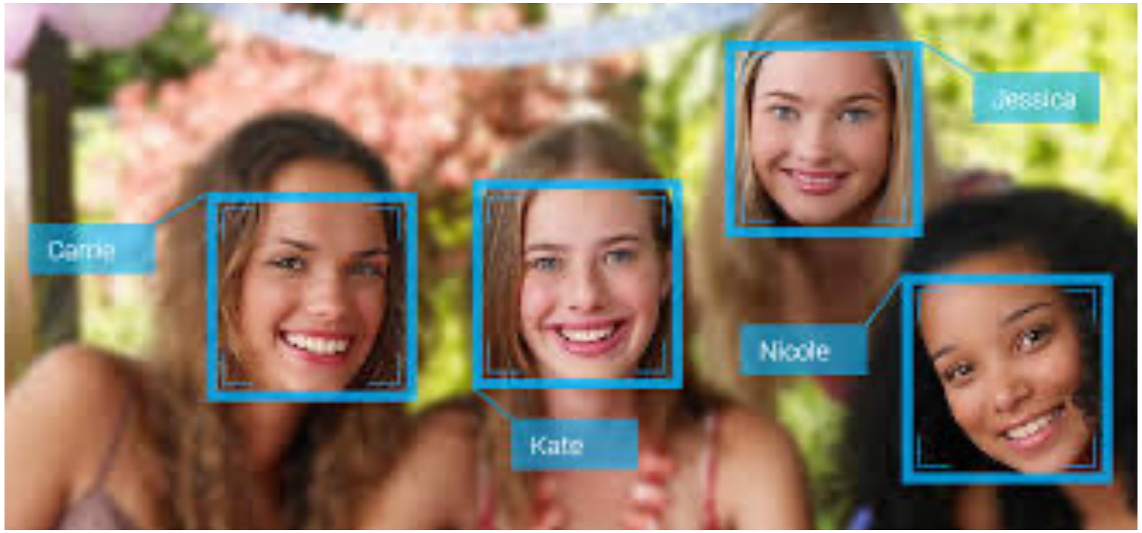
- Face detection: most cameras detect faces nowadays
  - Face detection gives as output a “bounding box” containing the faces and / or some facial features (*e.g.* eyes)
    - Requires a HUGE training dataset containing faces / not faces
    - Can serve as pre-processing to face recognition (see next slide)



Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

Examples of applications of computer vision

- Face recognition
  - Assign a name from the output bounding boxes of face detection
  - Obviously, we need a training dataset with images of all the persons that the machine needs to recognize!
  - Nowadays, face recognition problem is almost solved (many commercial products) -> researchers focus on face anti-spoofing!



Automatic face recognition

Examples of applications of computer vision

- Anti-spoofing: fighting against Presentation Attacks
  - Impostors trying to impersonate someone else

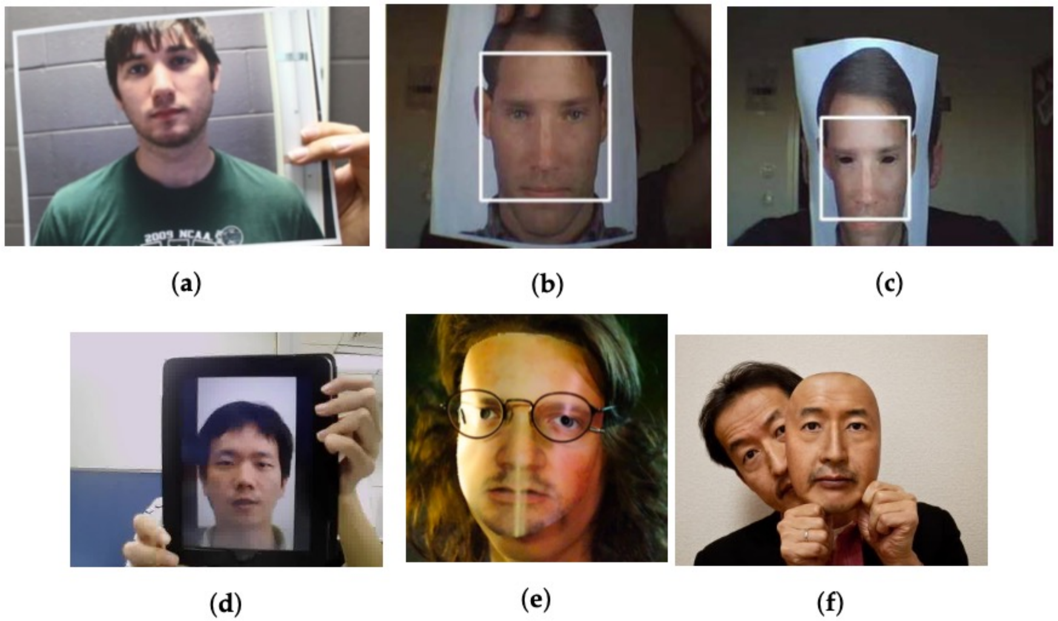
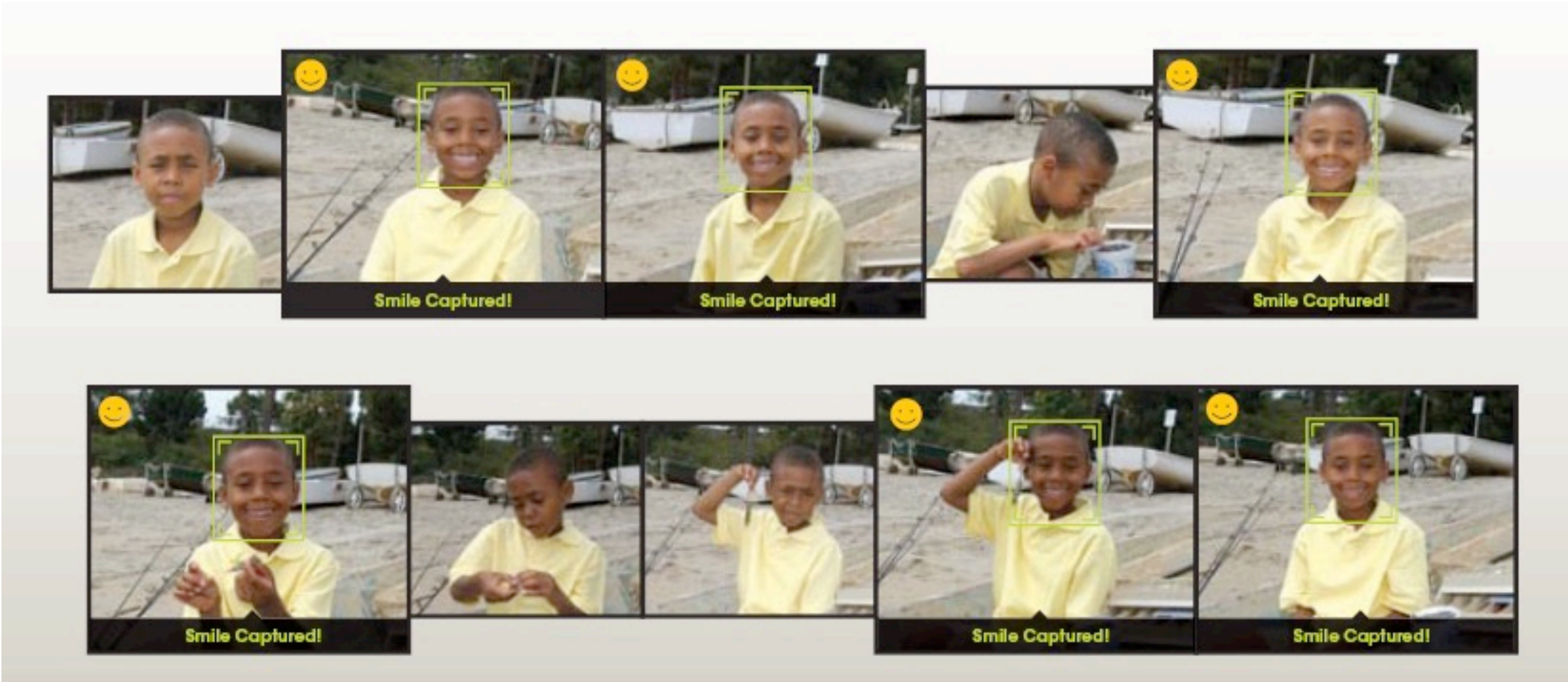


Figure 2. Examples of common facial presentation attacks: (**a**) a printed photo attack from the SiW dataset [17]; (**b**) an example of a warped photo attack extracted from [18]; (**c**) an example of a cut photo attack extracted from [18]; (**d**) a video replay attack from the CASIA-FASD dataset [19]; (**e**) a paper-crafted mask from UMRE [20]; and (**f**) a high-quality 3D mask attack from REAL-f [17].

*Z. Ming, **M. Visani**, M. M. Luqman and J. C. Burie, "A Survey on Anti-Spoofing Methods for Facial Recognition with RGB Cameras of Generic Consumer Devices", Journal of Imaging, 2020

Examples of applications of computer vision

- Smile detection: smart camera
  - Camera can automatically trip the shutter at the right instant to catch the perfect expression



Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

Examples of applications of computer vision

- More generally: facial expression recognition
  - Can be used for many final applications, such as such as human-machine interaction, face anti-spoofing or mental diseases diagnosis
  - Main difficulty: the expressions vary widely with the geographic region / culture

Examples of applications of computer vision

- Biometry: *e.g.* for login without a password, but with biometrics (fingerprint, iris, face,…)
  - Authentication (matching) problem: classif. with 2 classes

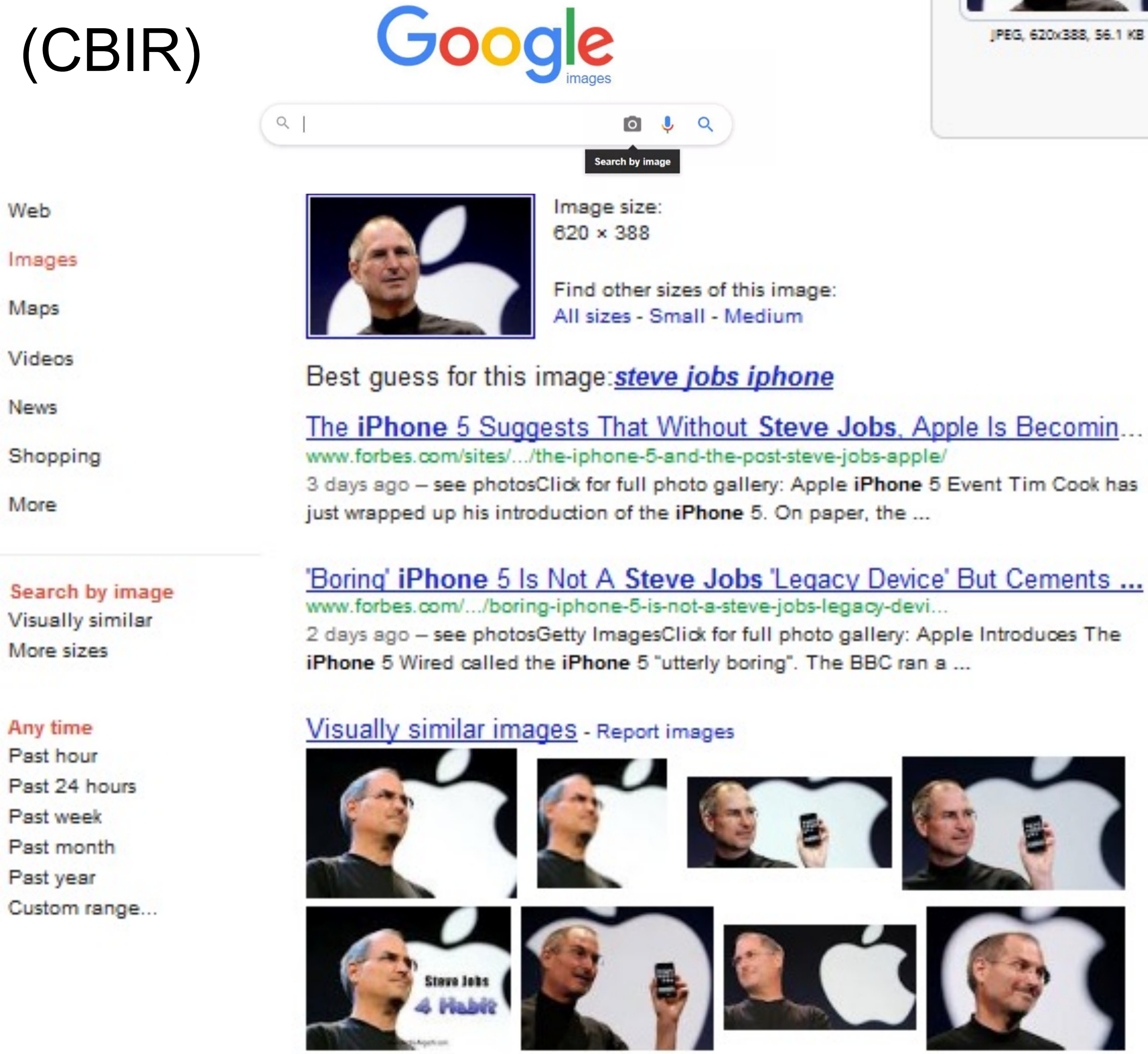
Fingerprint scanners on many new laptops, other devices

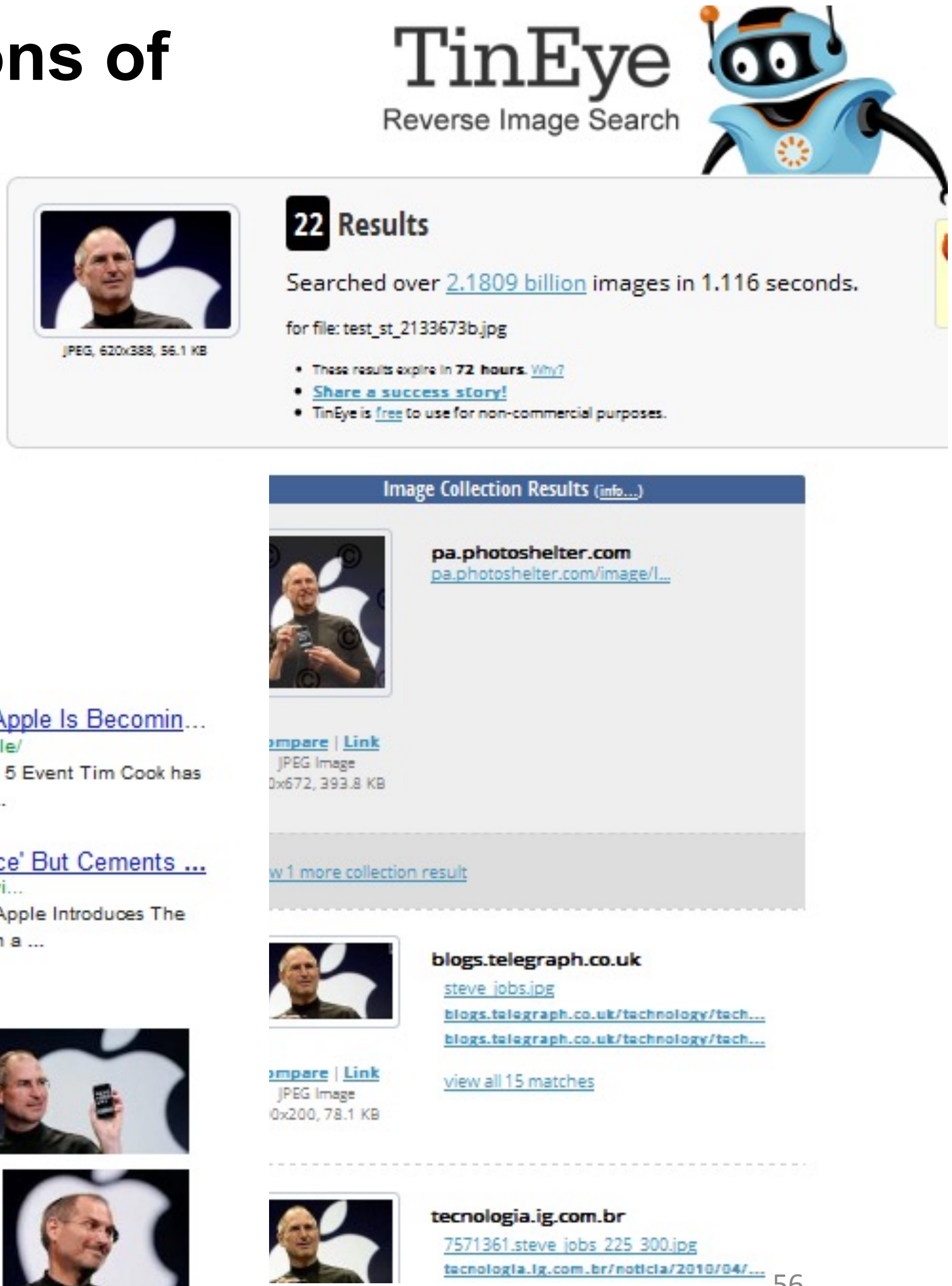
Face recognition systems now beginning to appear more widely
http://www.sensiblevision.com/

Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

Examples of applications of computer vision

- Content-Based Image Retrieval (CBIR)

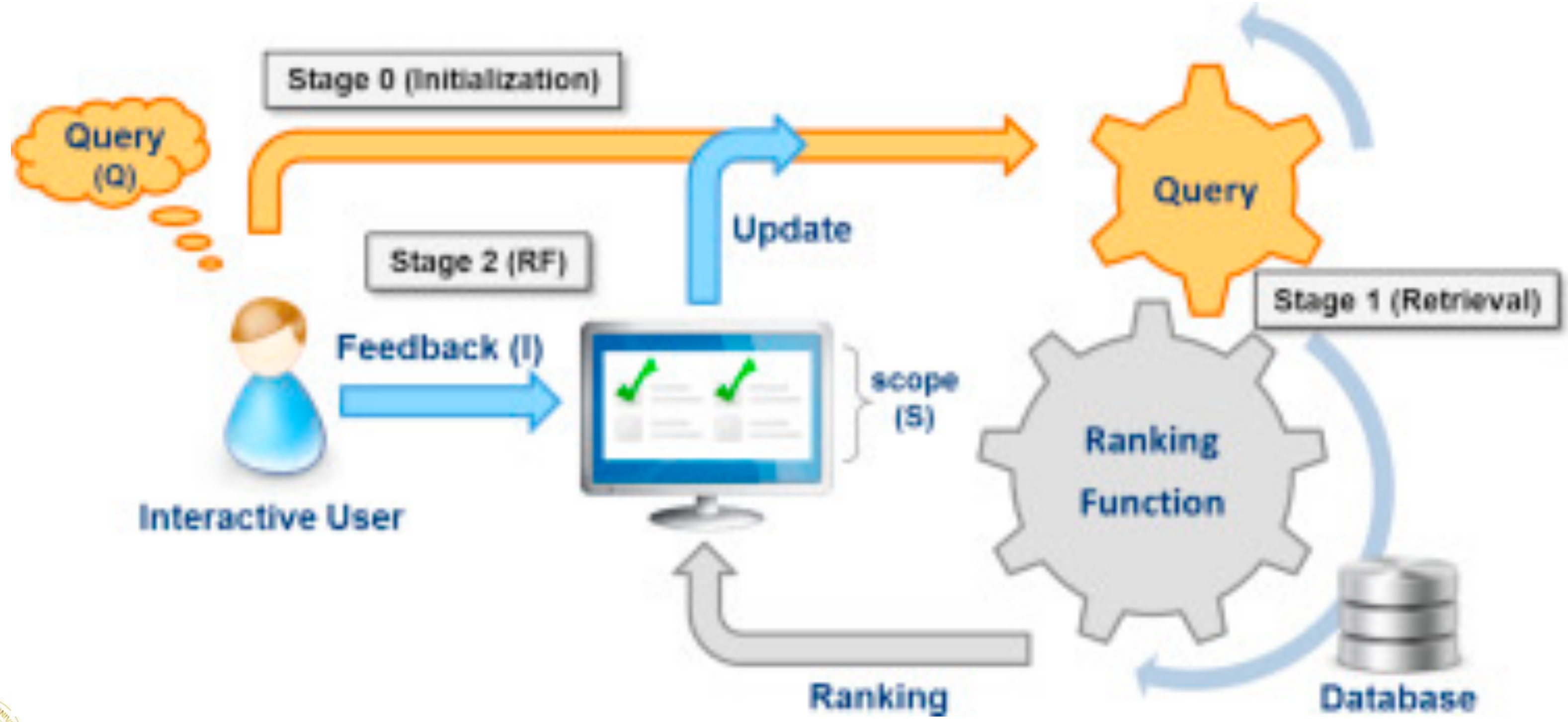




Examples of applications of computer vision

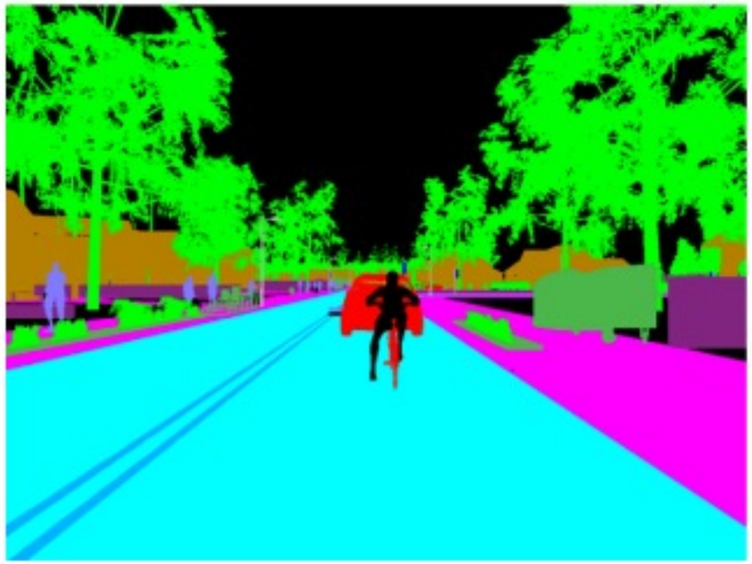
- Nowadays, CBIR is almost a solved problem
  - Many commercial products are working quite well for general images

  => Researchers focus on **relevance feedback** and specialization of the retrieved images for a specific user profile



Examples of applications of computer vision

- Smart cars → autonomous vehicles
  - Requires recognizing / getting a precise location for each object on the road
  - Still an active research area (on-going NAVER project in the BK.AI research center)



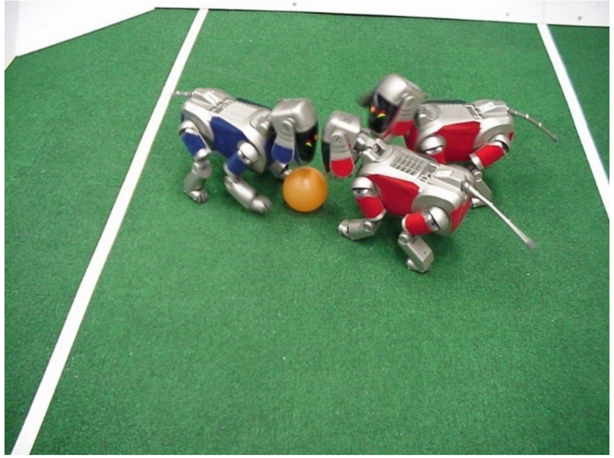
Examples of applications of computer vision

- Games / robots:

Vision-based interaction game (Microsoft's Kinect)

The devices generally contain RGB cameras + infrared projectors / detectors that estimate depth

Robot vacuum cleaner



http://www.robocup.org/

Examples of applications of computer vision

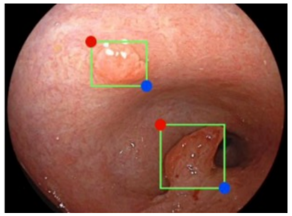
- To learn more about vision commercial applications in 2021:

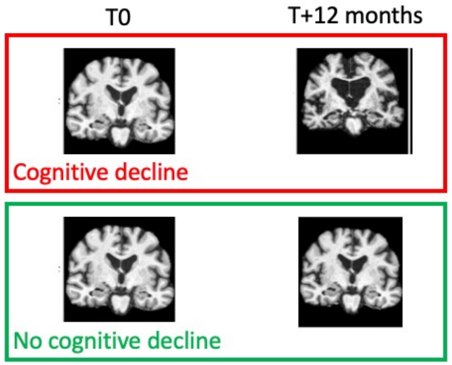
https://itrexgroup.com/blog/computer-vision-applications-in-different-industries/#

Some widely used tasks in Computer Vision

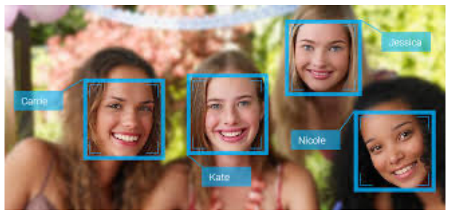
Some major tasks in Computer Vision

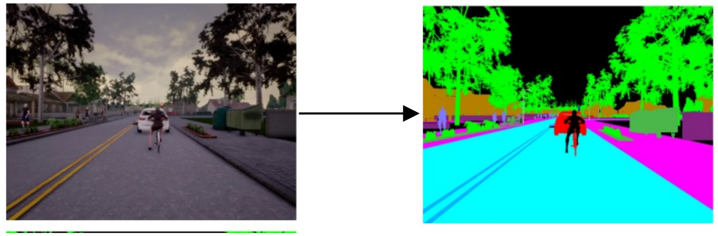
- Main tasks considered in this lecture
  - Classification / recognition
  - Detection
  - Semantic segmentation

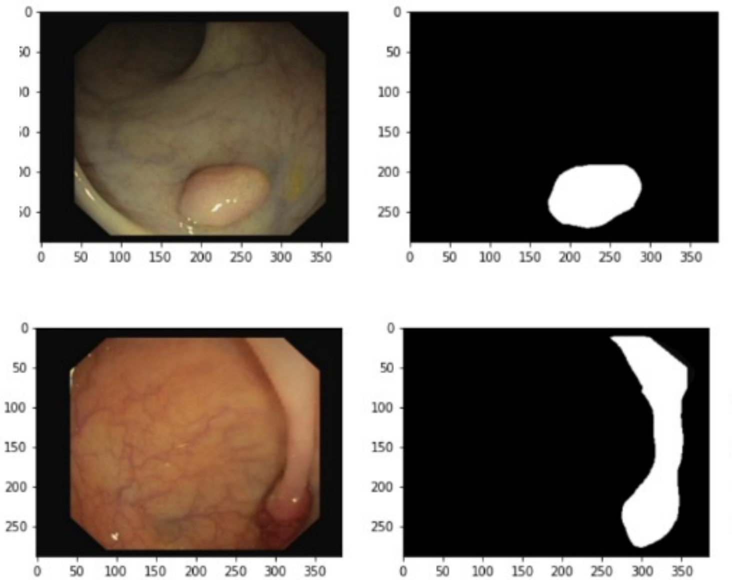




Example of **fine-grained** recognition







Some major tasks in Computer Vision

- Main tasks considered in this lecture
  - Detection
  - Classification / recognition (including fine-grained recognition)
  - Semantic segmentation
- These main task rely on **supervised classification** (see Chapter 6)

Supervised classification for Computer Vision

Supervised classification for Computer Vision

- « **Traditional** » classification methods for images
- **Before** the deep learning ''revolution'' in the 2010's (still active research)

Conception/use of ad-hoc image features: global/local ; color / shape / texture:

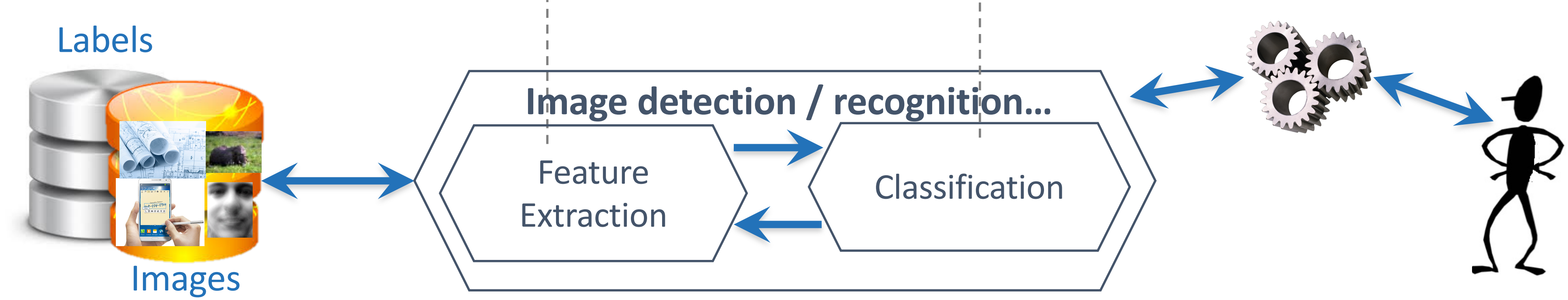
Color histogram, Hu moments, SIFT…

Depending on the type of images and application

Often, using off-the-shelf clustering methods or classifiers

SVMs, random forests, Multi-Layer Perceptrons…

Indexing
Retrieval (CBIR)
Detection
Recognition
(Semantic) segmentation
…



Training dataset

Supervised classification for Computer Vision

- In the 2010's, Deep Learning **revolutionised** Image Recognition
  - In **2012**, AlexNet won the ImageNet Large Scale Visual Recognition Challenge
    - 10,000,000 labeled images depicting **10,000+ object categories**
    - … Outperforming the other models by almost **11% accuracy gain**

Supervised classification for Computer Vision

- In the 2010's, Deep Learning **revolutionised** Image Recognition
  - In **2012**, AlexNet won the ImageNet Large Scale Visual Recognition Challenge..
  - In **2014-2015**, Deepface and FaceNet surpassed human-level recognition accuracy for face recognition
    - On very challenging face benchmarks, *e.g.*
      - Labeled Faces in the Wild (LFW)
        - More than 13,000 images of faces collected from the web
      - YouTube Faces (YTF)

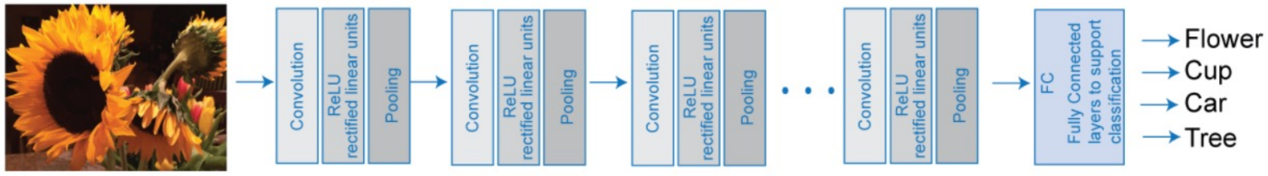
Supervised classification for Computer Vision

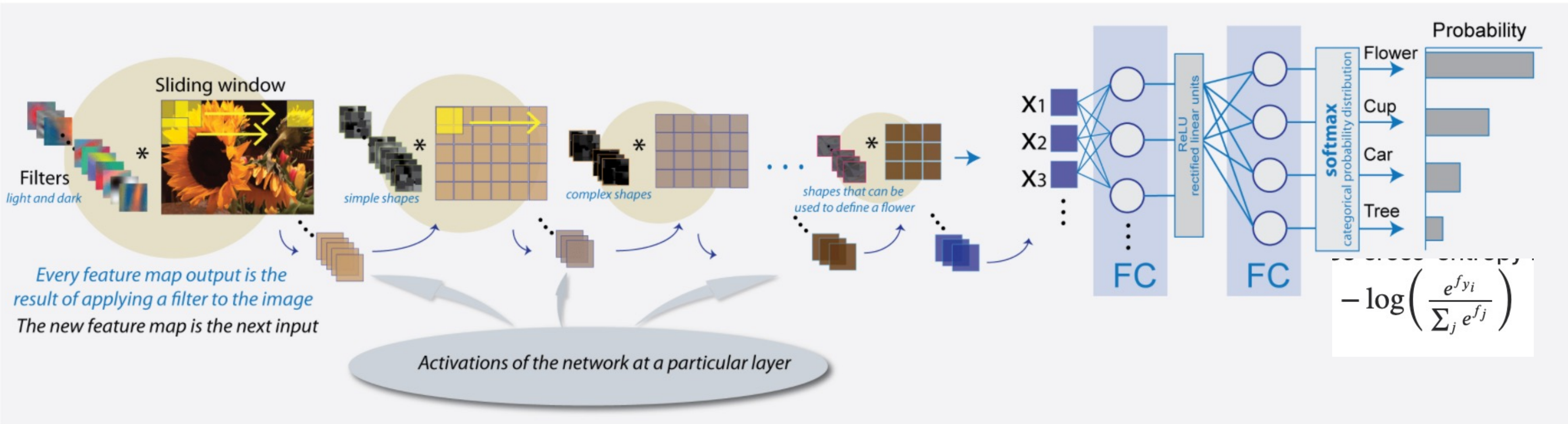
- In the 2010's, Deep Learning **revolutionised** Image Recognition
  - In **2012**, AlexNet won the ImageNet Large Scale Visual Recognition Challenge..
  - In **2014-2015**, Deepface and FaceNet surpassed human-level recognition accuracy for face recognition

- The above methods are based on **Convolutional Neural Networks**
  - In the rest of this lecture, we will focus on Convolutional Neural Networks, even if other Deep Neural Networks techniques are also useful for image analysis
    - For instance, recurrent Neural Networks (in particular LSTM), especially in the presence of a sequence of images / handwriting recognition

Supervised classification using deep learning for Computer Vision

- Convolutional Neural Networks
  - Regularized versions of Multi-Layer Perceptrons
  - Can be applied directly to images
  - Convolutions *a.k.a* kernels *a.k.a.* filters *a.k.a.* weights: http://setosa.io/ev/image-kernels/
    - **Learnt** by the model





[https://fr.mathworks.com/help/deeplearning/ug/introduction-to-convolutional-neural-networks.html]

Supervised classification using deep learning for Computer Vision

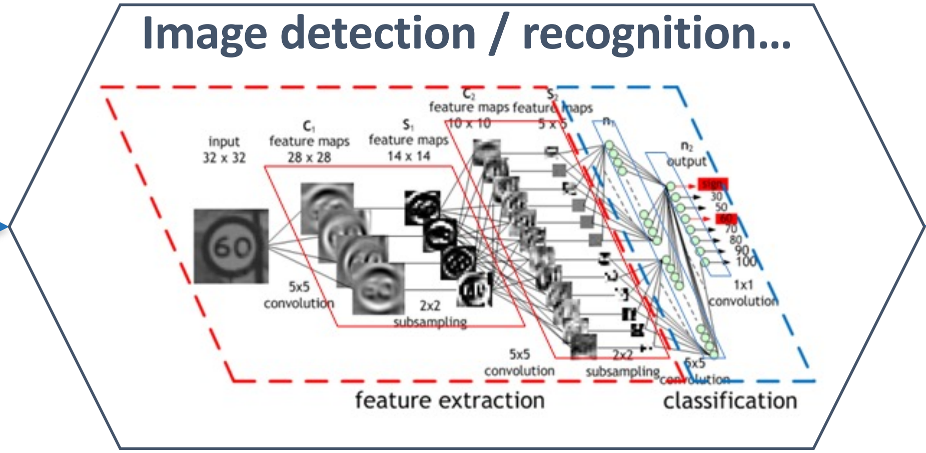
- **After** the deep learning ''revolution'' in the 2010's
  - For image analysis, one of the most used types of deep learning models are Convolutional Neural Networks (CNN)

- Why are CNN so popular in Computer Vision?
  - Reach excellent performances in general
  - Enable **end-to-end processing** of the images
    - Convolution layers **learn** filters that become more specialized when progressing in the network
    - No need to design / extract hand-crafted features

Labels

Images

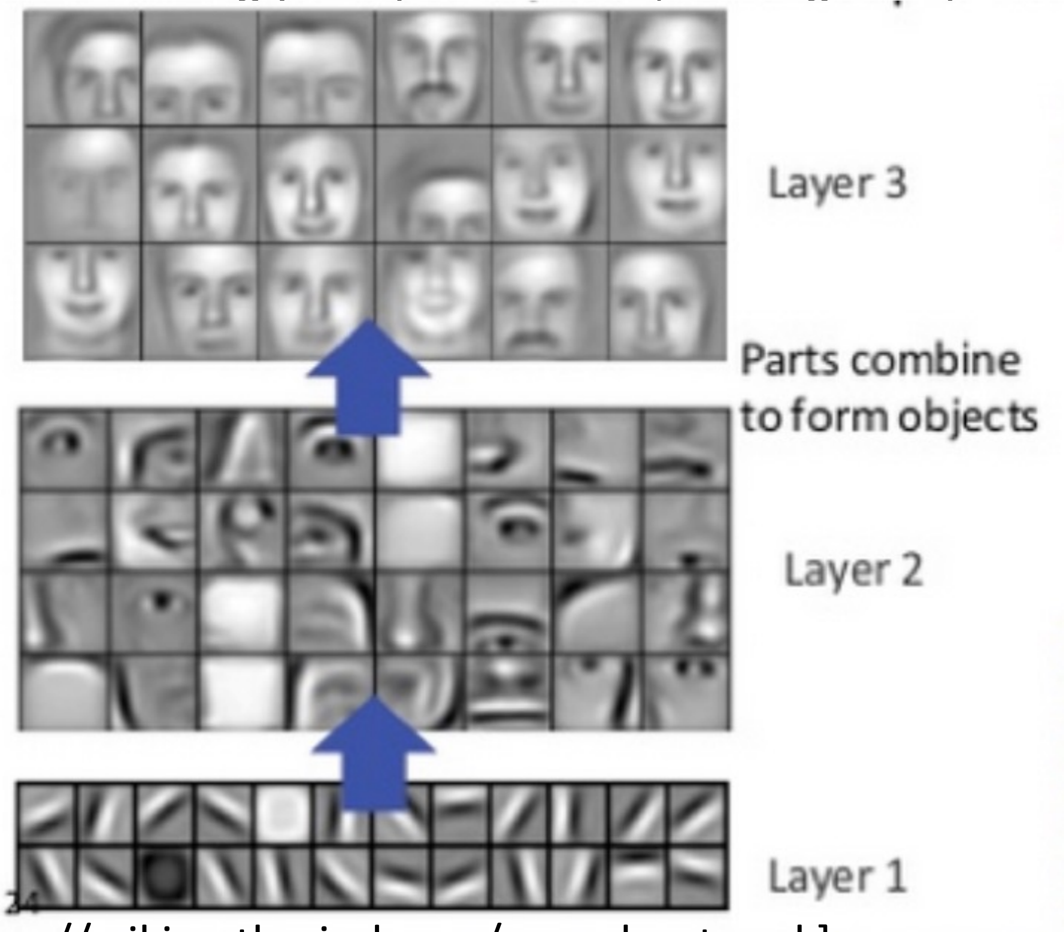
Training dataset



Indexing
Retrieval (CBIR)
Detection
Recognition
(Semantic) segmentation
…

Supervised classification using deep learning for Computer Vision

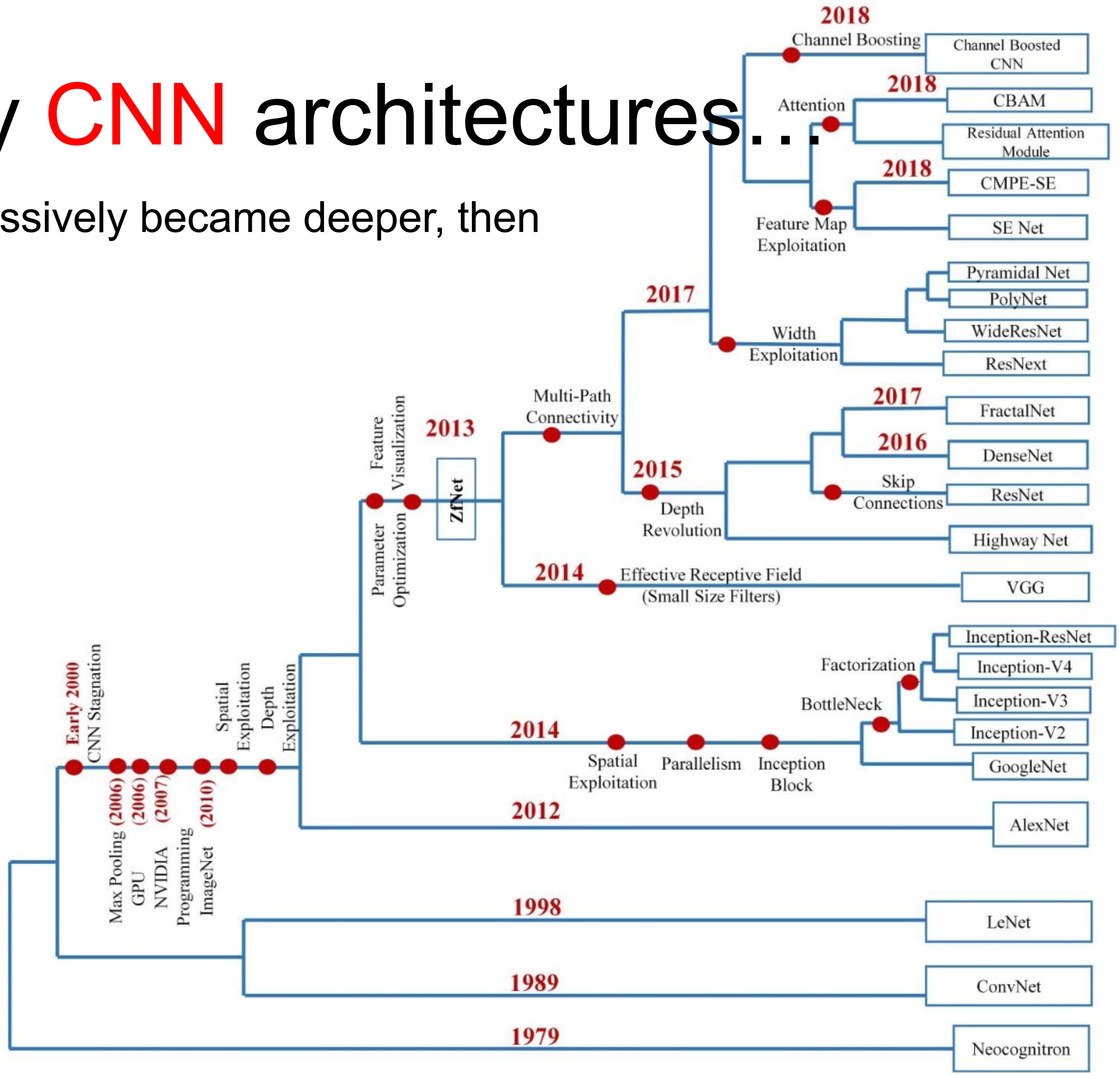
- Convolutional Neural Networks (CNN)
  - Extraction of increasingly complex features, as we go deeper in the network…



[https://wiki.pathmind.com/neural-network]

So many CNN architectures…

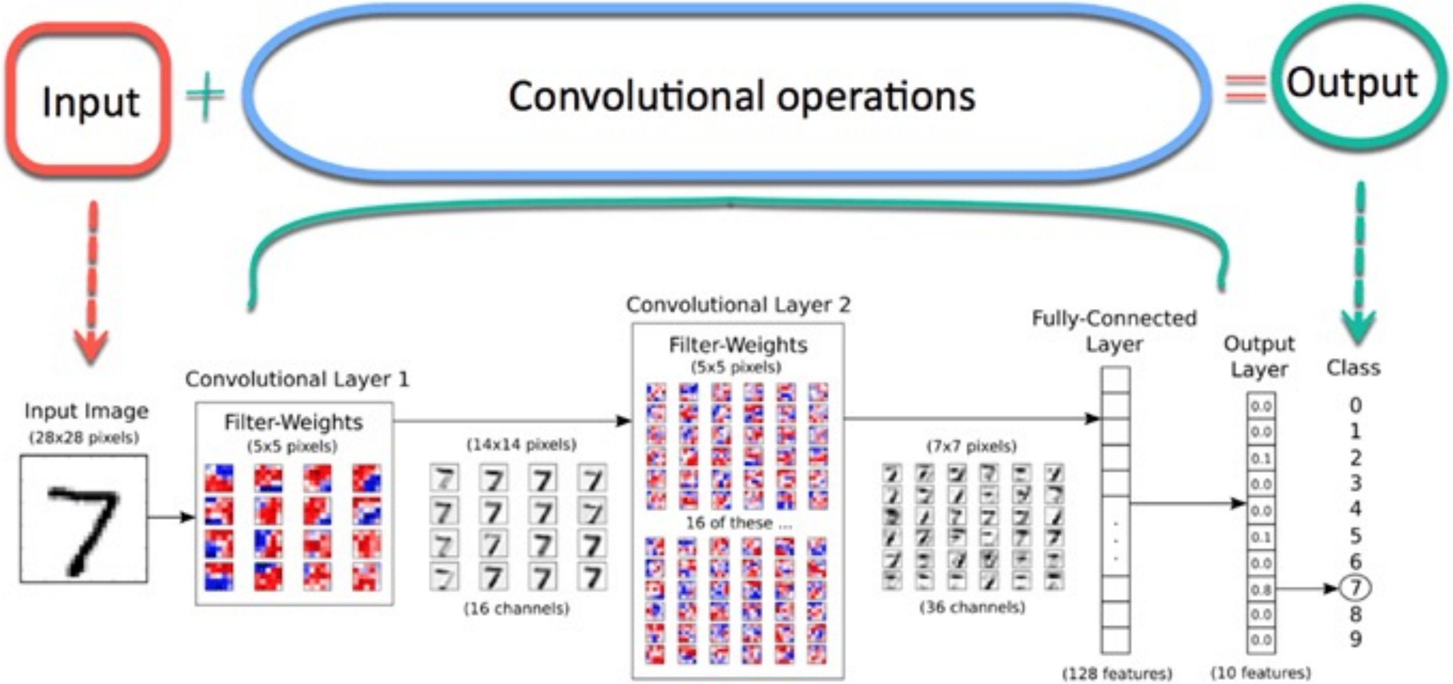
- Networks progressively became deeper, then wider



[https://towardsdatascience.com/top-10-cnn-architectures-every-machine-learning-engineer-should-know-68e2b0e07201]

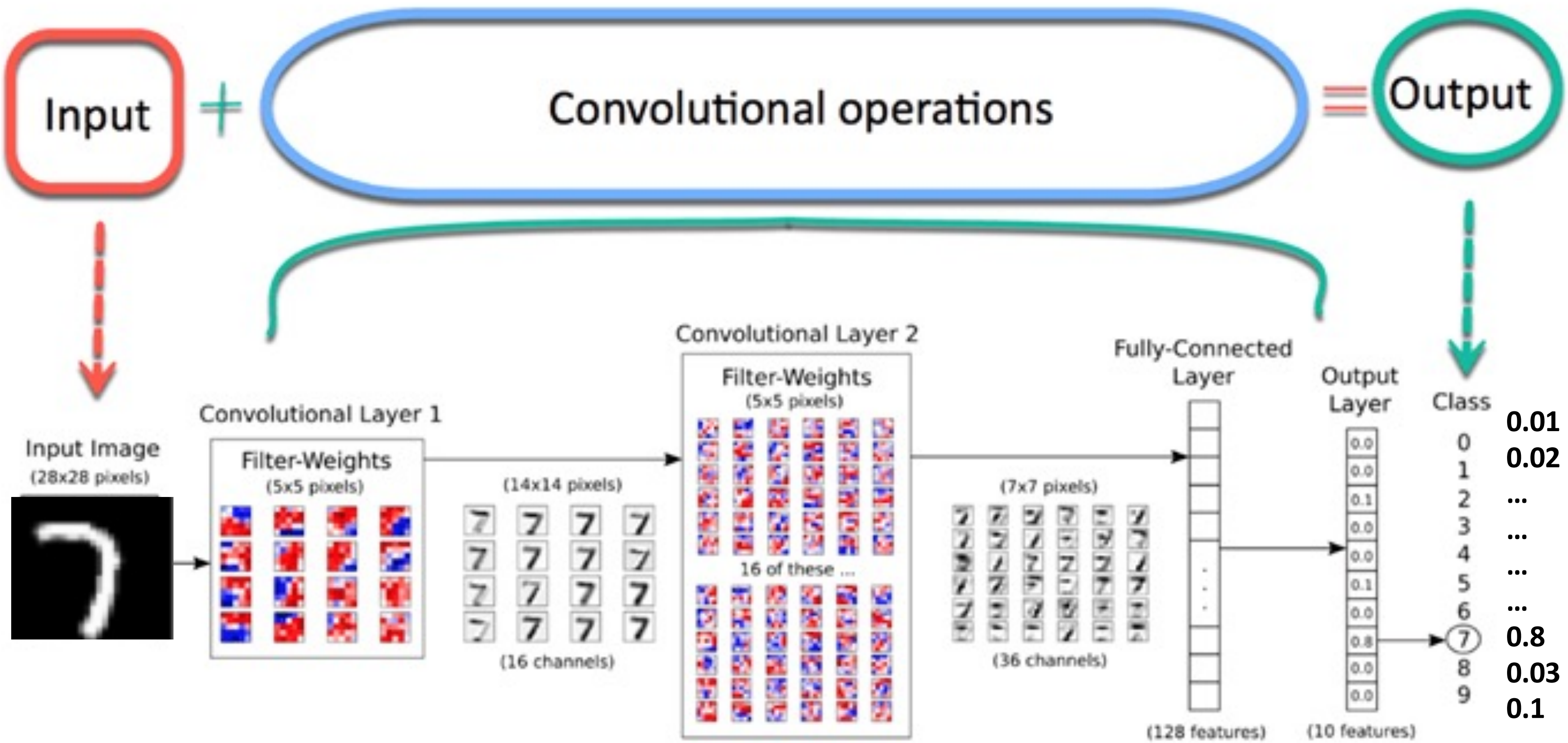
Supervised classification using deep learning for Computer Vision

- **First step**: training the CNN (supervised)
  - In general, requires huge amount of labelled data!!
  - We give the training images (with labels) as batches
  - Thanks to the true labels and a loss function, the network can evaluate its errors in the output layer and back-propagate the error in the network to adjust its filters
    - using some **backpropagation** technique: https://medium.com/@pavisj/convolutions-and-backpropagations-46026a8f5d2c
  - Full training generally requires several **epochs**



Supervised classification using deep learning for Computer Vision

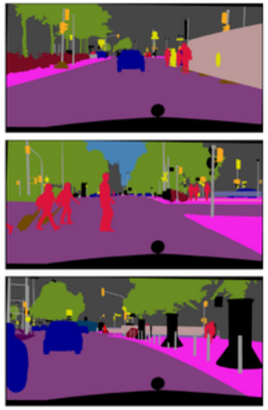
- **Second step:** using the CNN for inference on new images
  - The image is pre-processed in the same way as the training dataset
  - The kernels are frozen
  - The input image goes through the network, and the output will be a **probability** value for each class -> we chose the class with highest probability



Supervised classification using deep learning for Computer Vision

- Deep neural networks for image categorization
  - Generally require learning **millions** of parameters
    - Because they learn their own filters, depending on the objective!
  - Therefore, require **a lot** of data **in order to avoid overfitting**…

- But, especially for supervised tasks, it's often too costly/time-consuming to get so much labelled images!!!
  - Example of semantic segmentation task



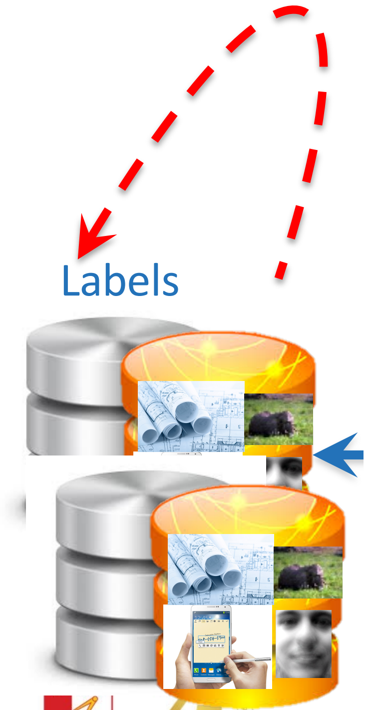
Supervised classification using deep learning for Computer Vision

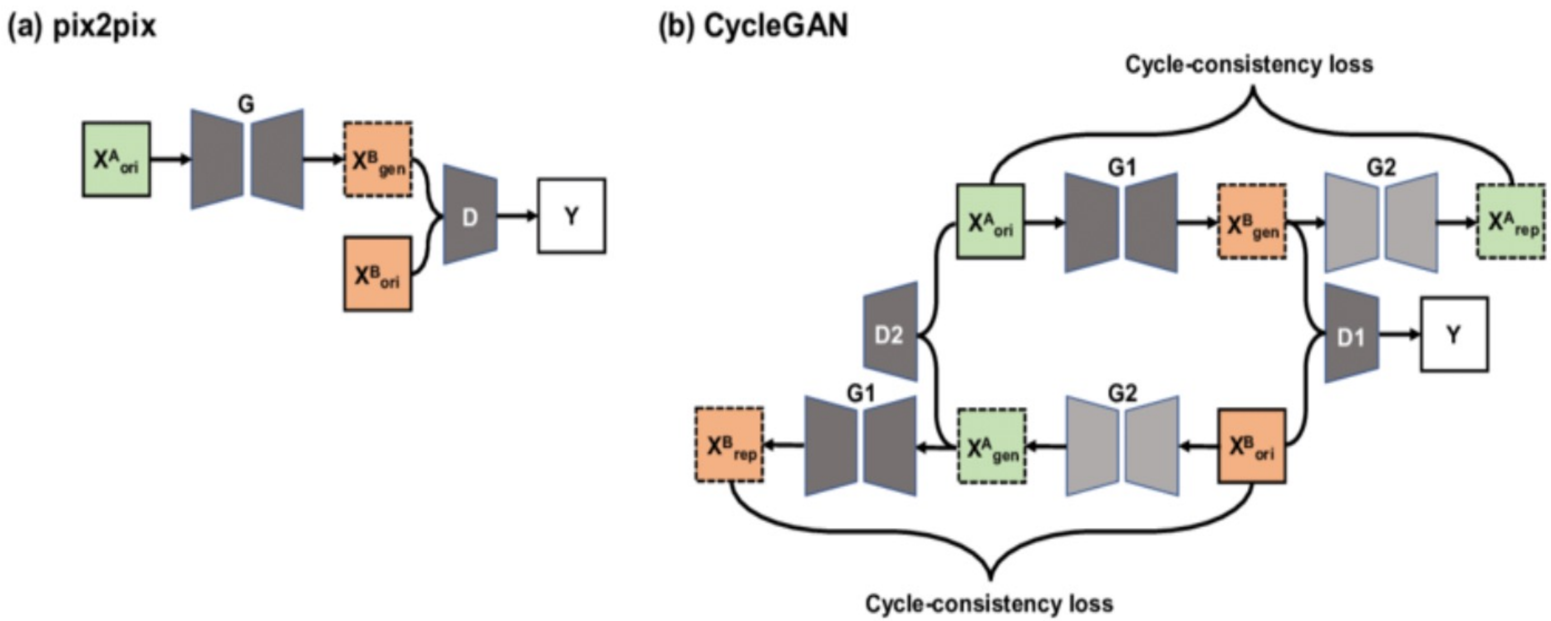
- Not really a problem!

Solution 1- Deep neural networks can be used to **generate** training data

- Example: GAN [Goodfellow *et al.*, NIPS 2014]

Generative Adversarial Networks



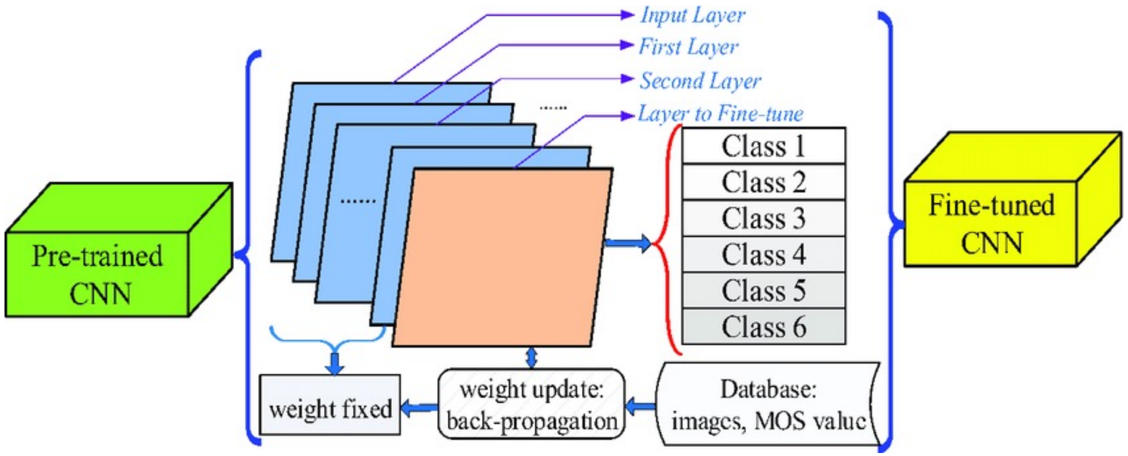


Supervised classification using deep learning for Computer Vision

- Not really a problem!

  Solution 2- Usual deep neural networks are **pre-trained** on some big labelled dataset, and then adapated for our dataset / task

  - either using fine-tuning: freezing the convolutional layers and retraining the last, (usually fully connected), layers, with our data



  - **or, using Domain adaptation**: several solutions, including adversarial learning (more complex)

Summary

Summary

- Computer Vision is a very hot topic
  - Both in industry and for academic researchers
  - Used in many many commercial applications
- Computer Vision relies heavily on Image Processing and Machine Learning
  - Can be seen as data science with a specific kind of data
  - Lately, deep learning is very trendy
  - The field of Computer Vision is constantly evolving
  - Very exciting research field!!!
- I am the leader of the Computer Vision Group inside the BK.AI research center
  - If you're interested in working with us for graduation thesis/projects, check our activities on https://bkai.ai/research/cv-team/ and contact me!

Some tutorials

Tutorial 1: CNN for total beginners

- CNNs for object recognition
  - Link for better understanding convolutions:
    - http://setosa.io/ev/image-kernels/
  - Tutorial for better understanding CNNs, and implementing them in Python (need to install **TensorFlow** and **Keras**)
    - https://www.kdnuggets.com/2019/07/convolutional-neural-networks-python-tutorial-tensorflow-keras.html

Tutorial 2: CNN for people who already used/trained CNNs

- Hot topic 1: Adversarial learning
  - Tutorial from NAVER (please put on a headset if you listen to the videos)
    - https://europe.naverlabs.com/eccv-2020-domain-adaptation-tutorial/
  - (Shorter) Tutorial about deep domain adaptation
    - https://towardsdatascience.com/deep-domain-adaptation-in-computer-vision-8da398d3167f
  - Tutorial (in Python) to apply adversarial learning (but not to images, this would take too much time)
    - https://realpython.com/generative-adversarial-networks/

Tutorial 3: for advanced students (who already know about reinforcement learning)

- Hot topic 2: Neural Architecture Search (NAS)

  - A short introduction about the fundamentals of NAS
    - https://towardsai.net/p/machine-learning/the-fundamentals-of-neural-architecture-search-nas

  - (Longer, more complete) tutorial about NAS
    - https://www.youtube.com/watch?v=wL-p5cjDG64

  - Tutorial (in Python) to implement NAS with reinforcement learning
    - https://lab.wallarm.com/the-first-step-by-step-guide-for-implementing-neural-architecture-search-with-reinforcement-learning-using-tensorflow-99ade71b3d28/

Questions

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!!!

soict.hust.edu.vn/ fb.com/groups/soict