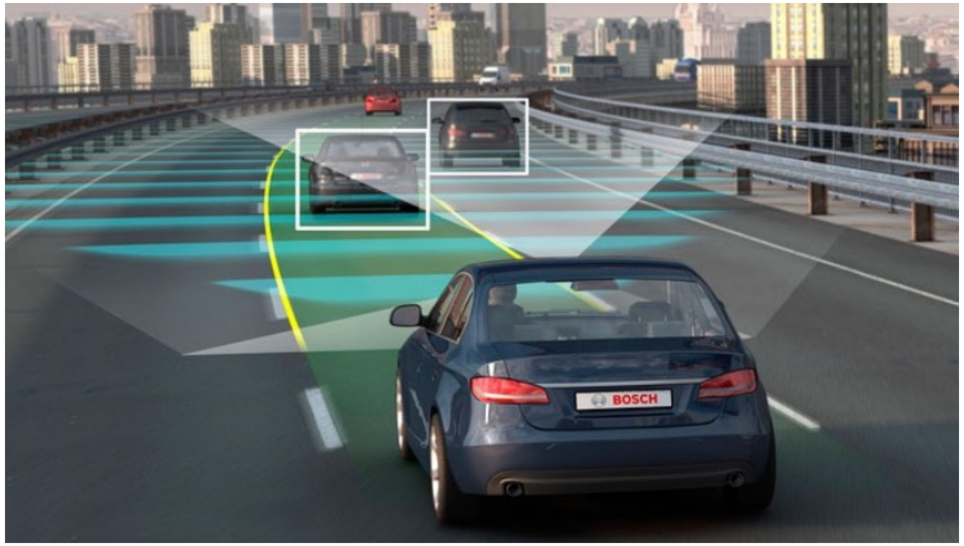
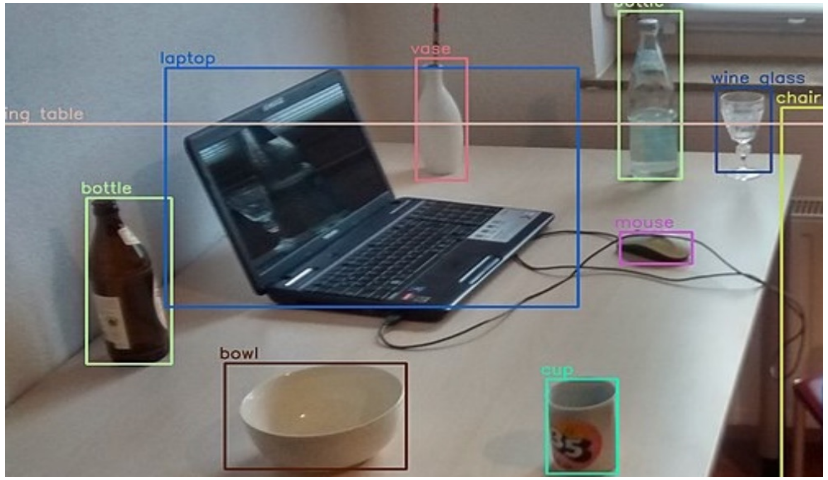
Lecture 11 - Vision

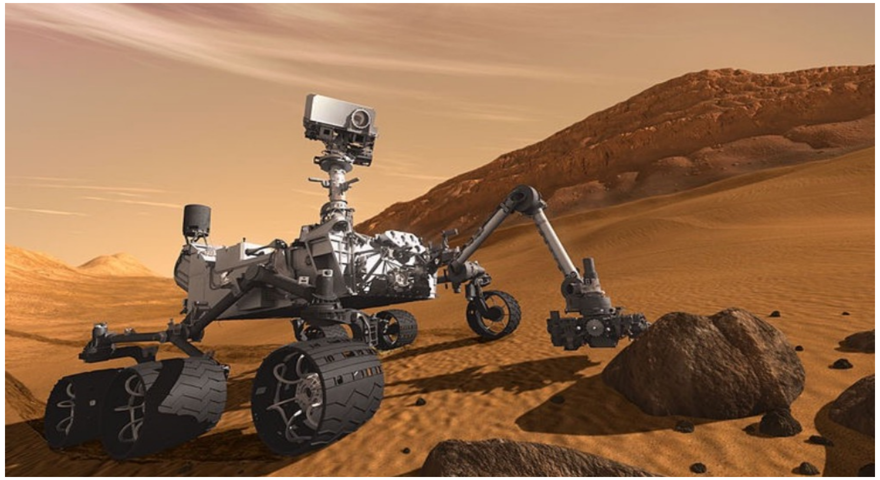
Introduction to Computer Vision, Image Processing and Analysis

Alexandros Lioulemes, PhD

What is Computer Vision







Goal of Computer Vision

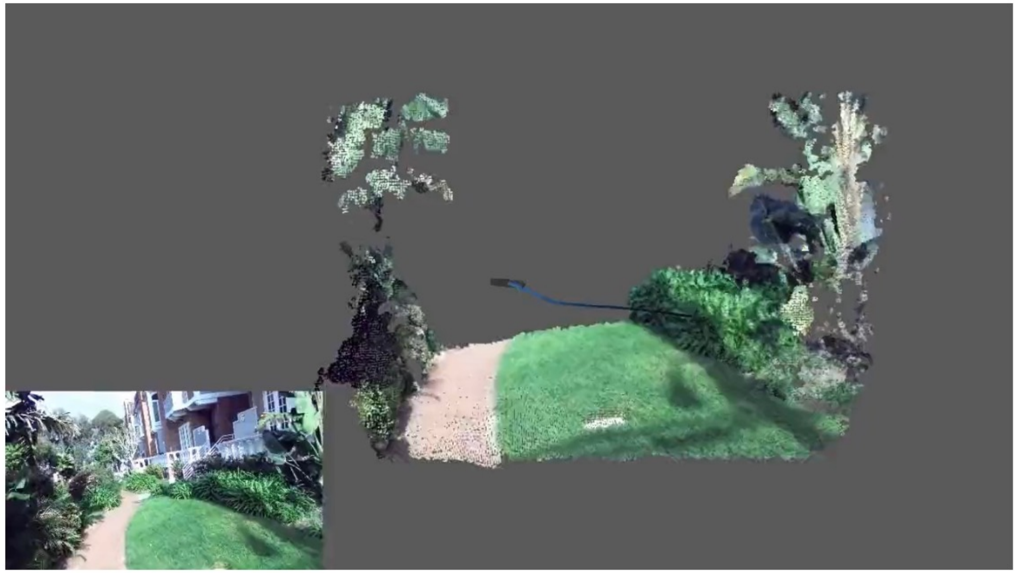
Goal of Computer Vision

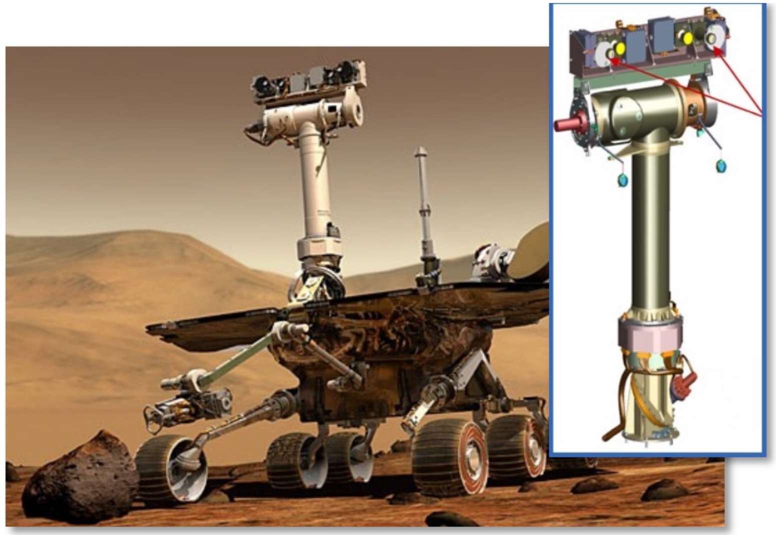
1. Image Recognition
2. Object Detection
3. Image Segmentation
4. Image Generation
5. Optical Character Recognition (OCR)
6. Pose Estimation
7. Scene Understanding
8. Action Recognition

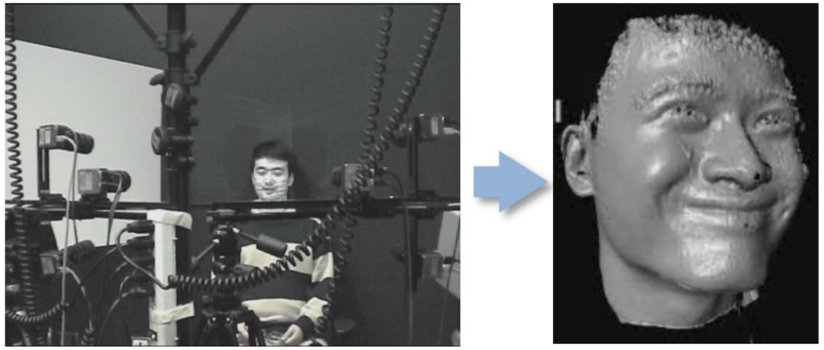
Goal of Computer Vision

- Compute the 3d shape of the world

ZED 2i Camera

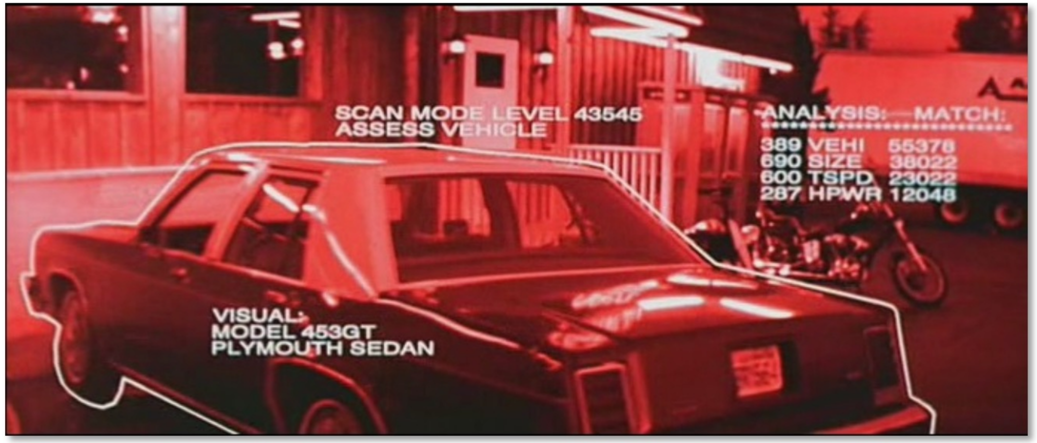






Goal of Computer Vision

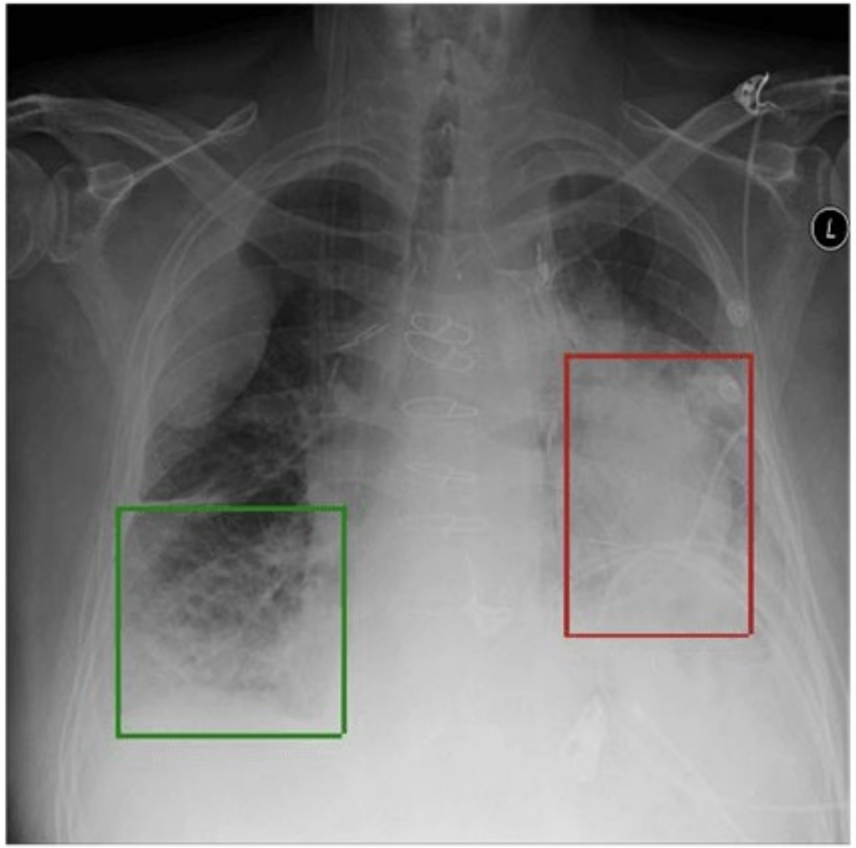
- Recognize objects and people



Terminator 2, 1991

Goal of Computer Vision

- “Enhance” images



Goal of Computer Vision

- Improve photos ("Computational Photography")

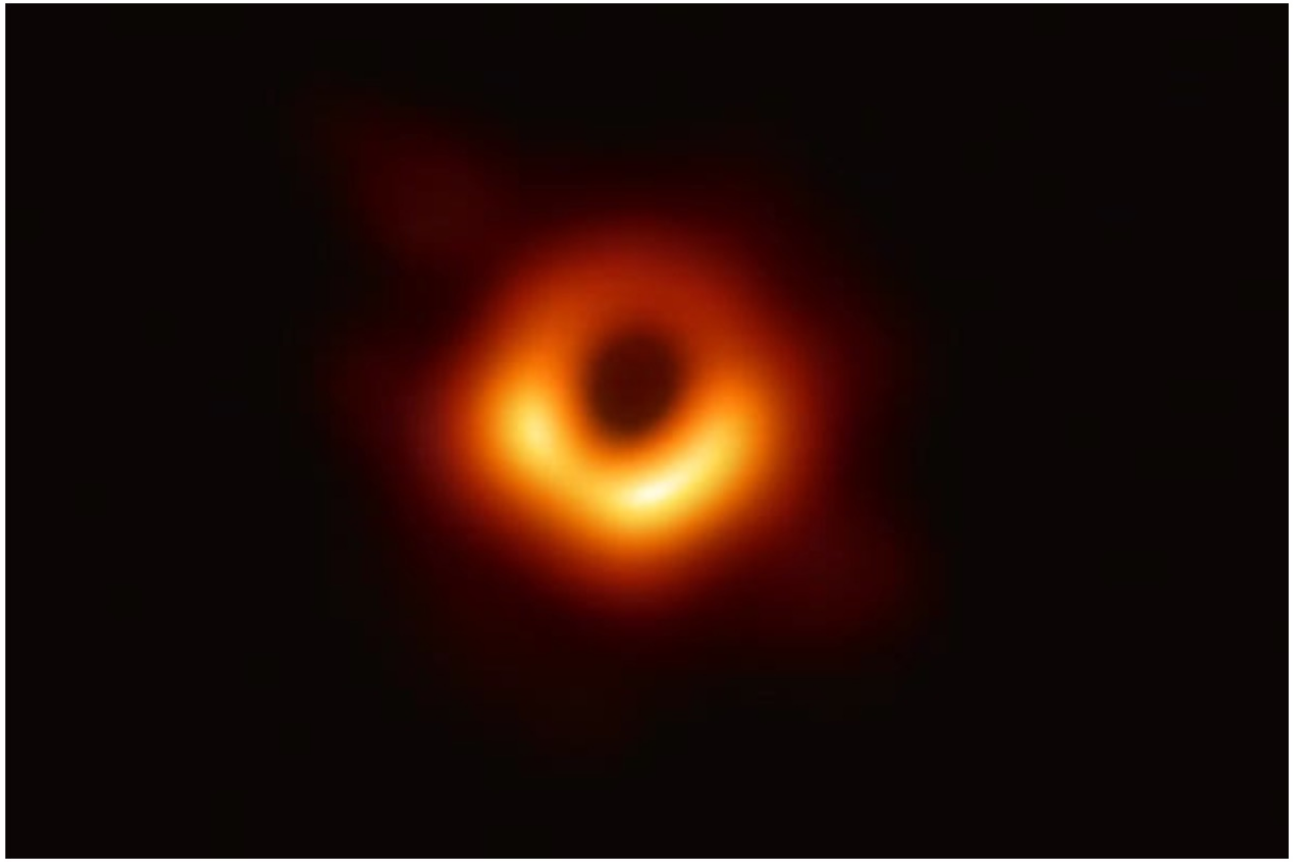
Super-resolution (source: 2d3)

Depth of field on cell phone camera

Removing objects

Goal of Computer Vision

- Reconstruction of objects and “black holes” – April 10, 2019



Why study Computer Vision

- Understanding the world through visual data
- Automation and efficiency
- Enhancing human-computer interaction
- Medical applications
- Images and video analysis
- Research and Innovation
- Ethical considerations
- Cross-disciplinary impact
- Career opportunities

State of the art: Optical character recognition

- If you have a scanner, it probably came with OCR software

Automatic check processing

Sudoku grabber

http://sudokugrab.blogspot.com/

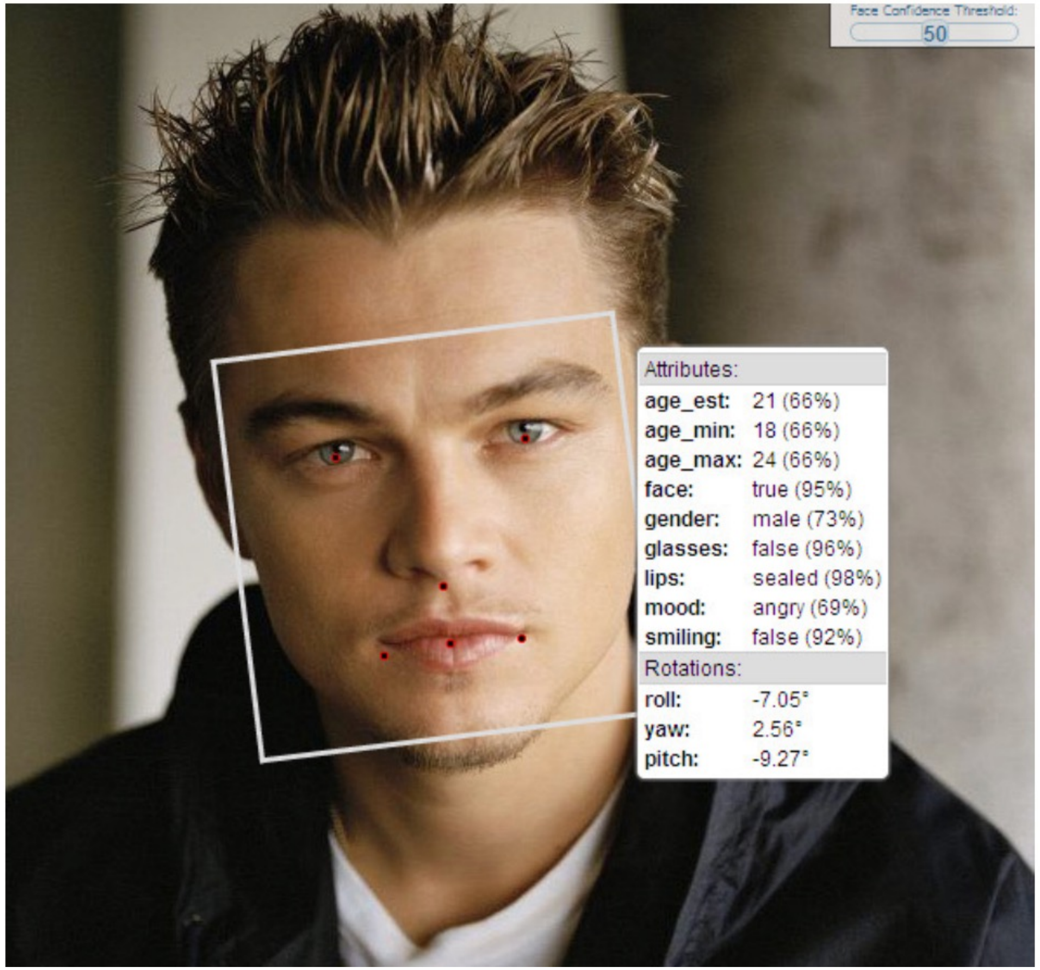
License plate readers

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

State of the art: Face detection

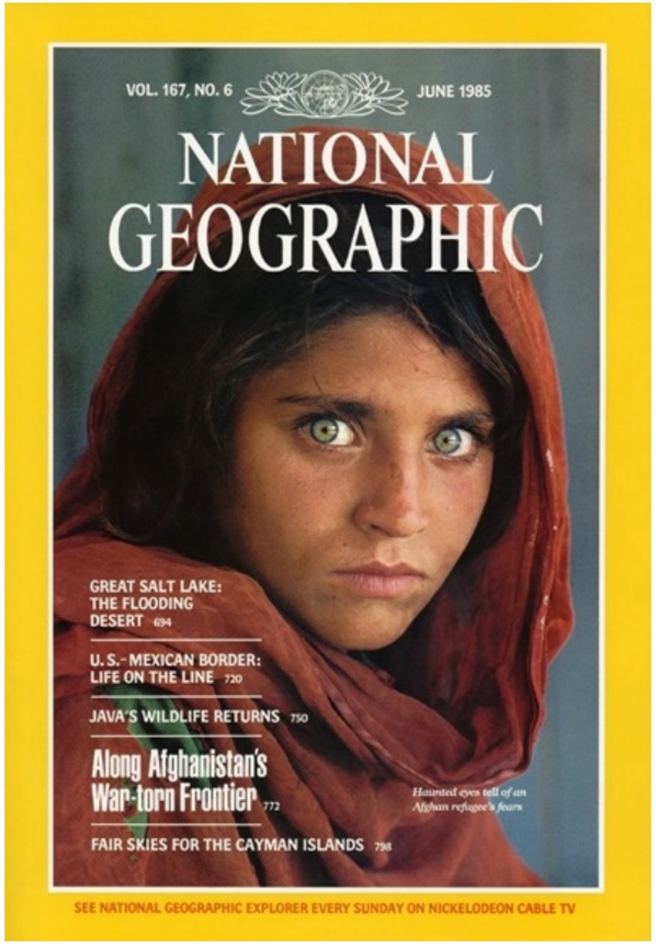
- Nearly all cameras detect faces in real time

State of the art: Face analysis and recognition

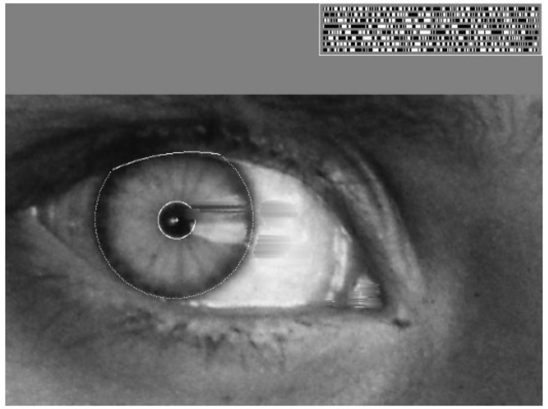


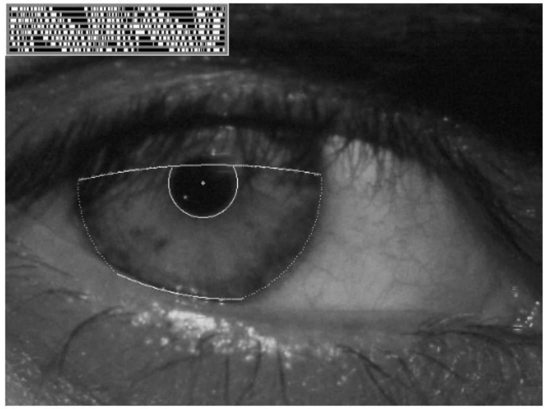
State of the art: Vision-based biometrics

- Who is she?



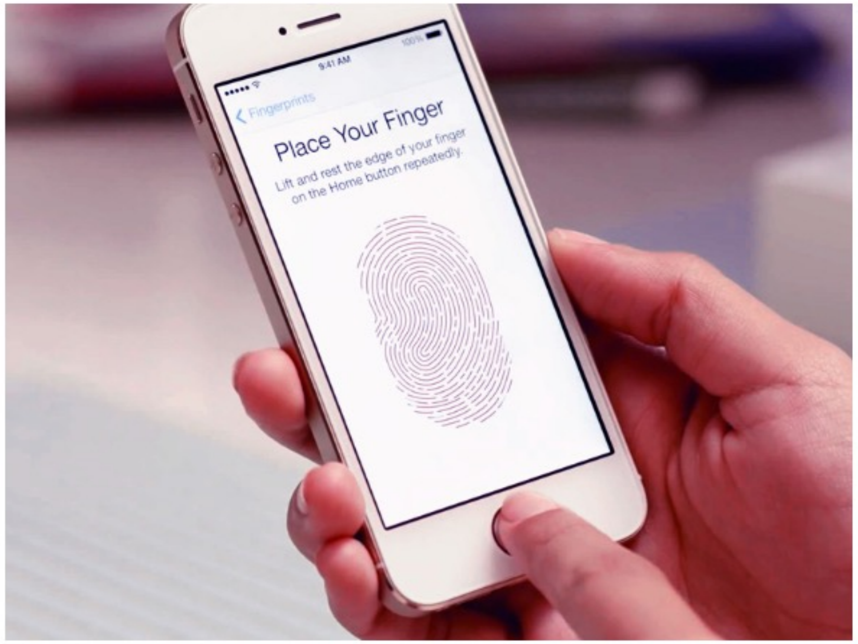
"How the Afghan Girl was Identified by Her Iris Patterns"





https://www.cl.cam.ac.uk/~jgd1000/afghan.html

State of the art: Login without a password

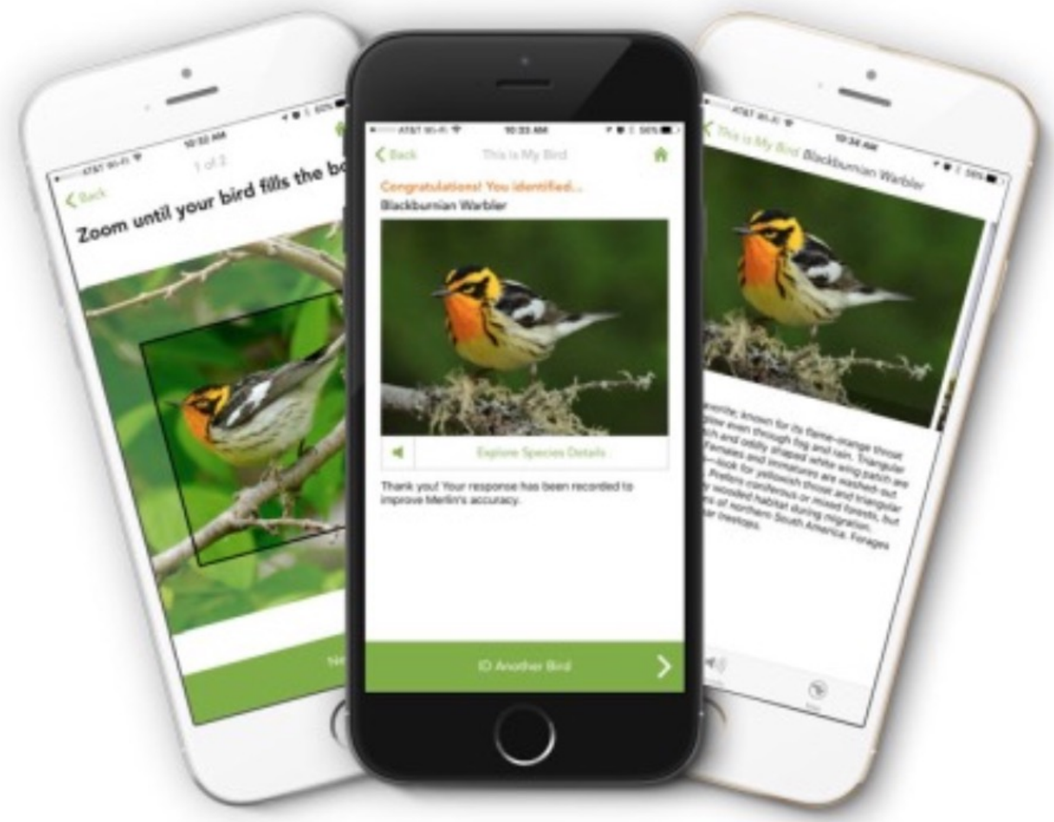


Fingerprint scanners on many new smartphones and other devices

Face unlock on Apple iPhone X

http://www.sensiblevision.com/

State of the art: Bird identification



Merlin Bird ID (based on Cornell Tech technology!)

State of the art: Special effects - shape capture

The Matrix movies, ESC Entertainment, XYZRGB, NRC

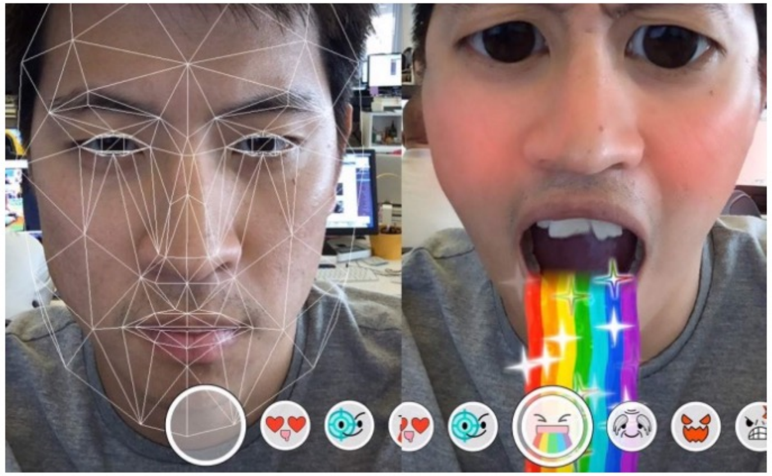
State of the art: Special effects - motion capture

Pirates of the Carribean, Industrial Light and Magic

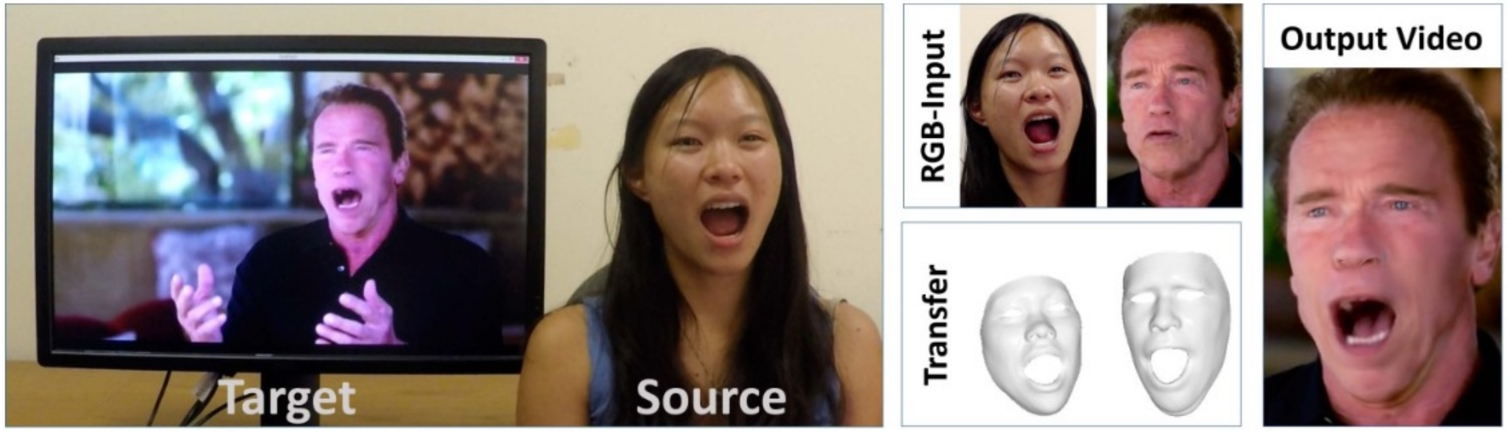
State of the art: No green screen - motion capture

https://www.vulture.com/2020/01/how-the-irishman-used-cgi-and-special-effects-on-actors.html

State of the art: 3D face tracking

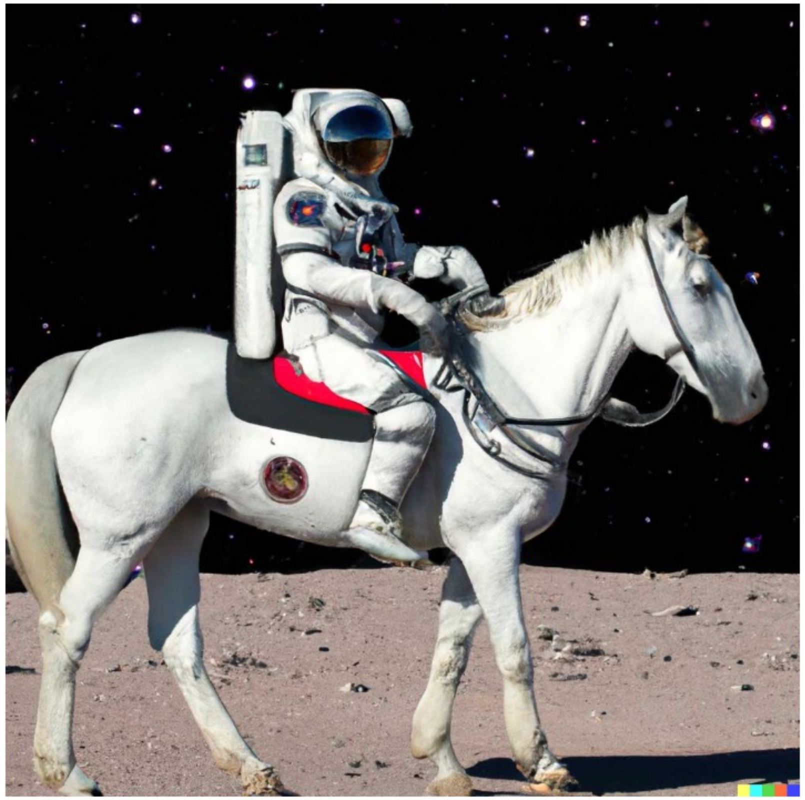


Snapchat Lenses



http://www.graphics.stanford.edu/~niessner/thies2016face.html

State of the art: Image synthesis



"An astronaut riding a horse in a photorealistic style" – DALL-E 2

https://openai.com/dall-e-2

"A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat" – Imagen

https://imagen.research.google

State of the art: Sports

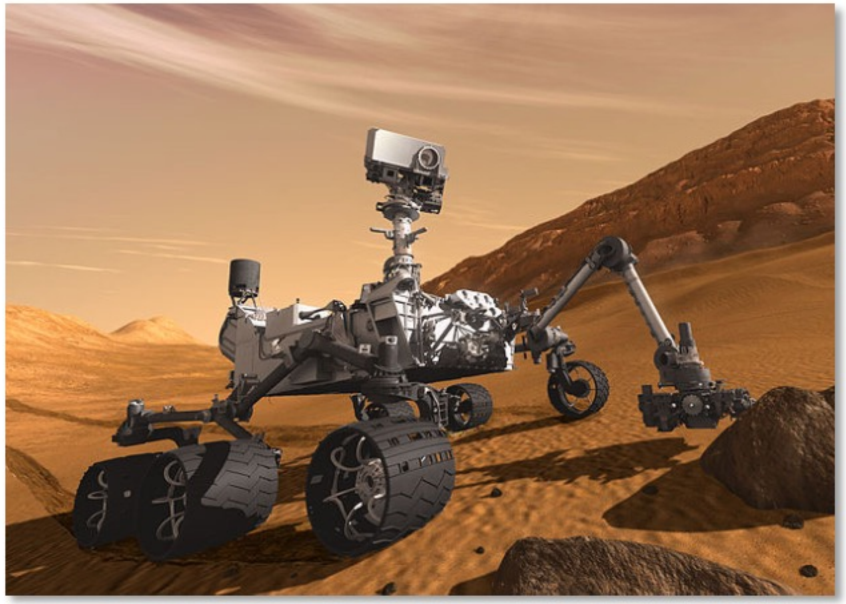
State of the art: Smart cars

- Advanced driver assistance systems (ADAS)
- Tesla Autopilot
- Safety features in many cars
- Self-driving cars

https://www.synopsys.com/automotive/what-is-adas.html

State of the art: Robotics

https://www.robocup2016.org/en/events/amazon-picking-challenge/



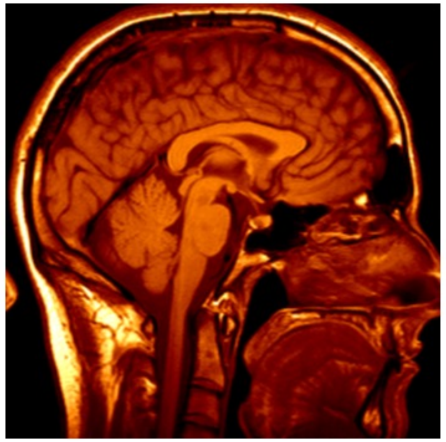
NASA's Mars Curiosity Rover

https://en.wikipedia.org/wiki/Curiosity_(rover)

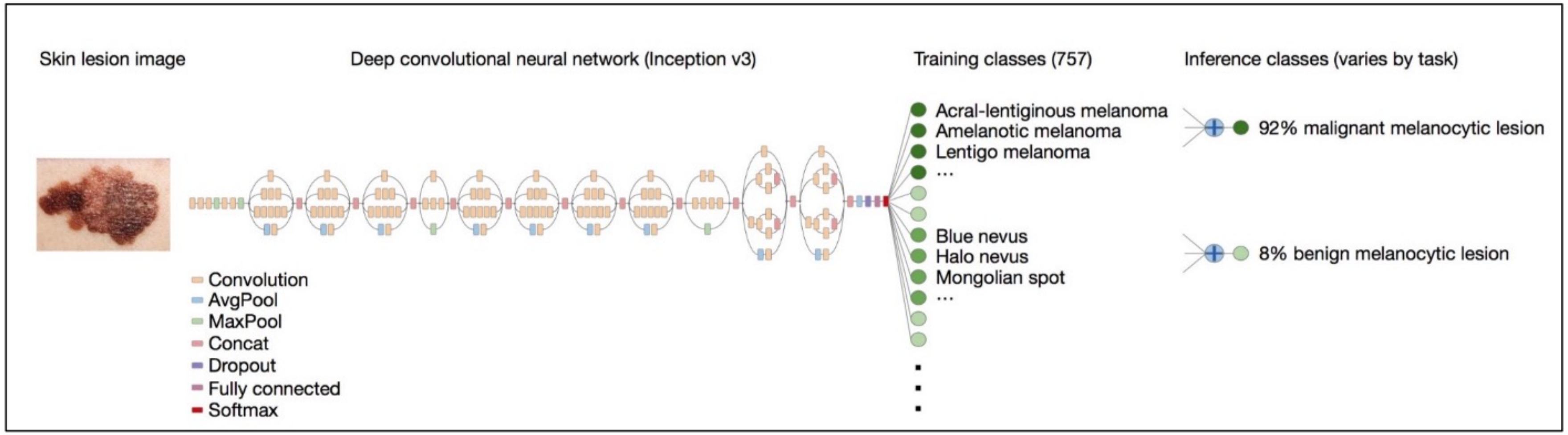
Amazon Picking Challenge

https://www.aboutamazon.com/news/transportation/meet-scout Amazon Scout

State of the art: Medical imaging

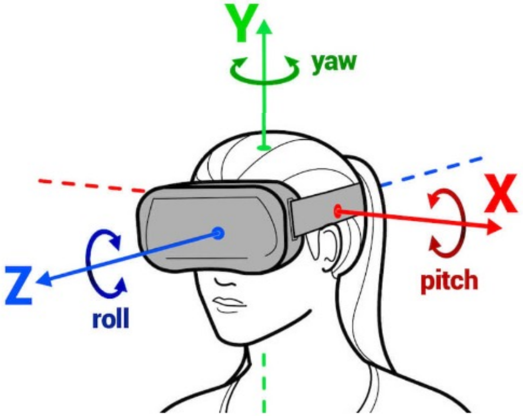


3D imaging
(MRI, CT)



Skin cancer classification with deep learning
https://cs.stanford.edu/people/esteva/nature/

State of the art: Spatial Computing



6DoF head tracking


Hand & body tracking


3D scene understanding


3D-360 video capture

Current state of the art

- You just saw many examples of current systems.
  - Many of these are less than 5 years old
- Computer vision is an active research area, and rapidly changing
  - Many new apps in the next 5 years
  - Deep learning powering many modern applications
- Many startups across a dizzying array of areas
  - Deep learning, robotics, autonomous vehicles, medical imaging, construction, inspection, VR/AR, ...

Why Computer Vision is difficult?

Viewpoint variation

Illumination

Credit: Flickr user michaelpaul

Scale

Why Computer Vision is difficult?

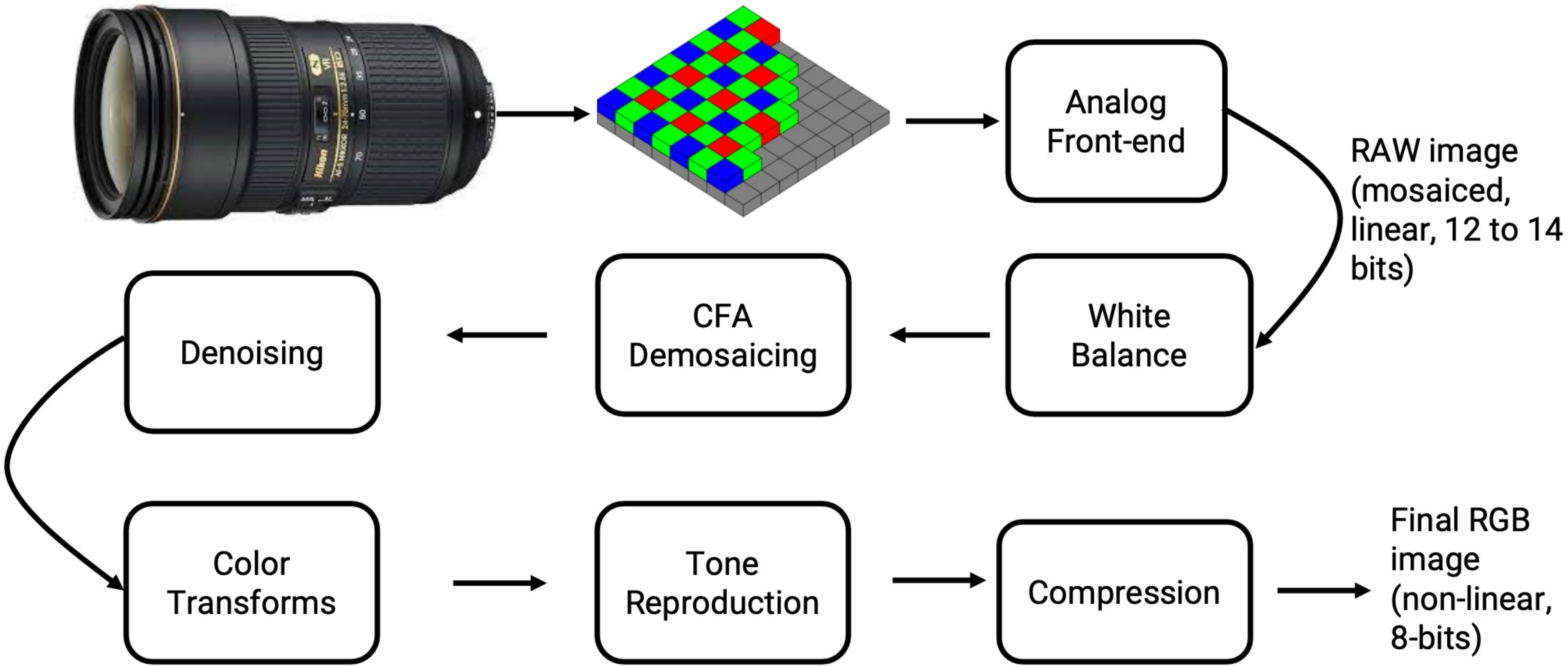
Intra-class variation

Motion (Source: S. Lazebnik)

Background clutter

Occlusion

What it takes to capture an image?



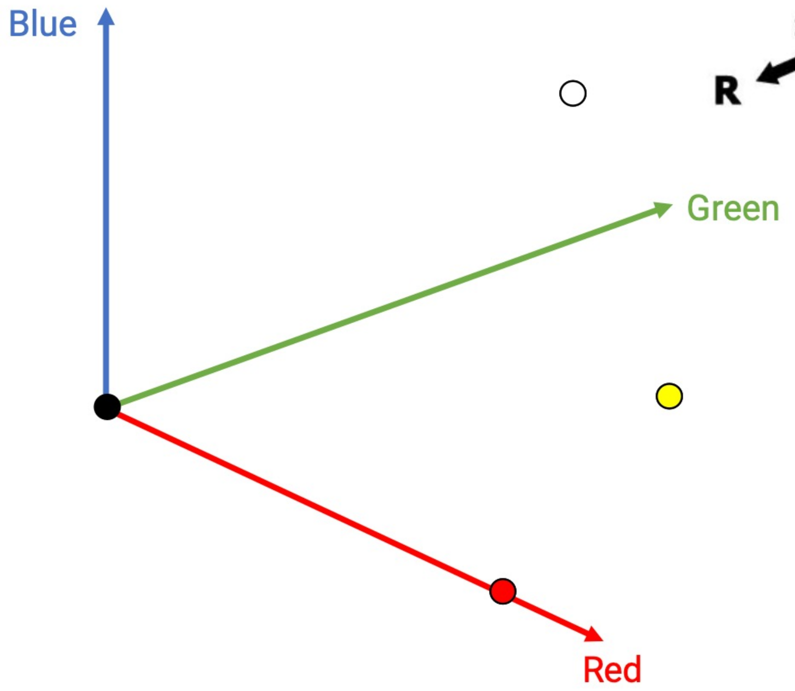
https://en.wikipedia.org/wiki/Bayer_filter

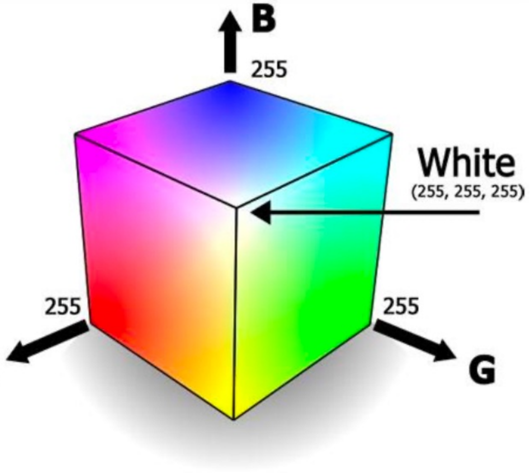
Color - Red/Blue/Green (RGB)

RGB Color Cube

$\mathbf{x} = [r \quad g \quad b]$

Why is this a cube and not any other shape?
Cartesian Coordinates





Color - Hue/Saturation/Value (HSV)

$$R' = \frac{R}{255}$$

$$G' = \frac{G}{255}$$

$$B' = \frac{B}{255}$$

$$C_{max} = \max\left(R', G', B'\right)$$

$$C_{min} = \min\left(R', G', B'\right)$$

$$\Delta = C_{max} - C_{min}$$

Hue is calculated as follows:

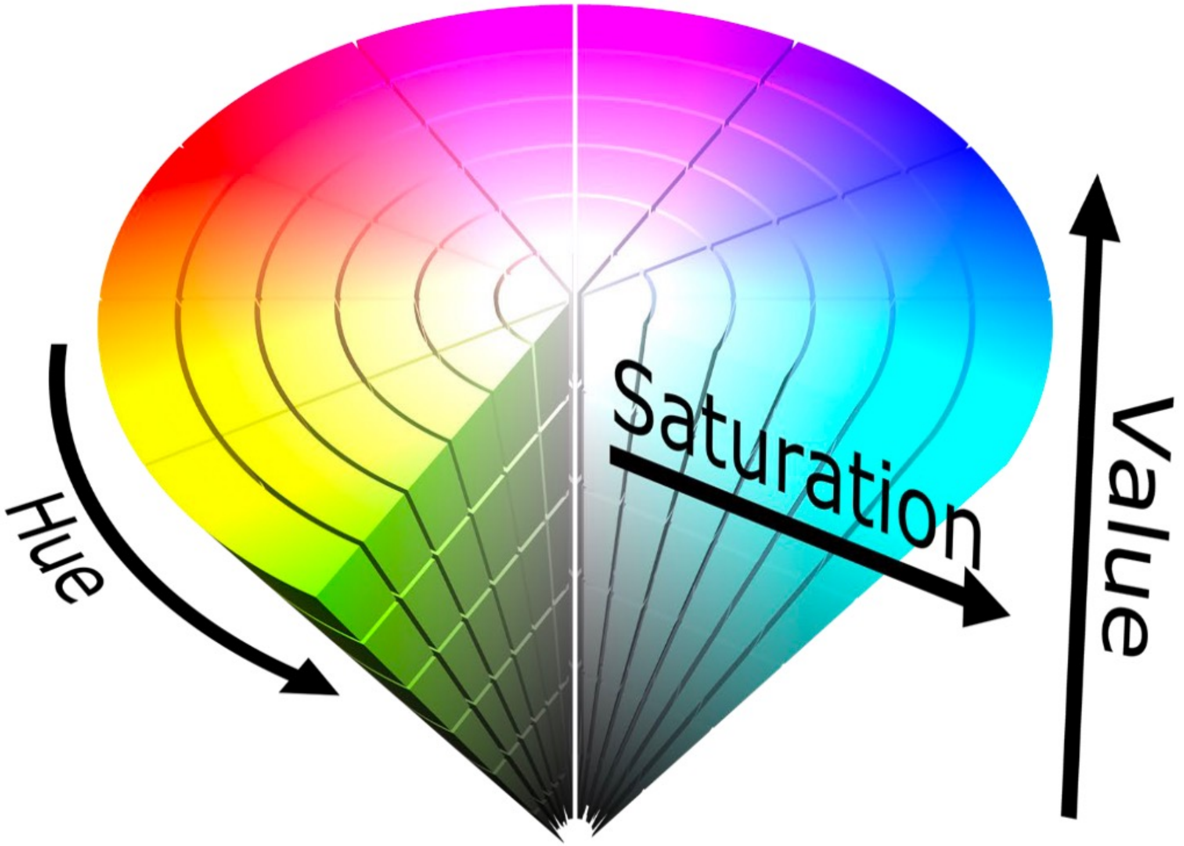
$$\begin{cases} 0^\circ & \text{if } \Delta = 0 \\ 60^\circ \times \frac{G'-B'}{\Delta} 60^\circ \times \frac{B'-R'}{\Delta} + 2 & \text{if } C_{max} = G' \\ 60^\circ \times \frac{R'-G'}{\Delta} + 4 & \text{if } C_{max} = B' \end{cases}$$

Saturation is calculated as follows:

$$\begin{cases} 0 & \text{if } C_{max} = 0 \\ \frac{\Delta}{C_{max}} & \text{if } C_{max} \neq 0 \end{cases}$$

Value is calulated as follows:

$$V = C_{max}$$



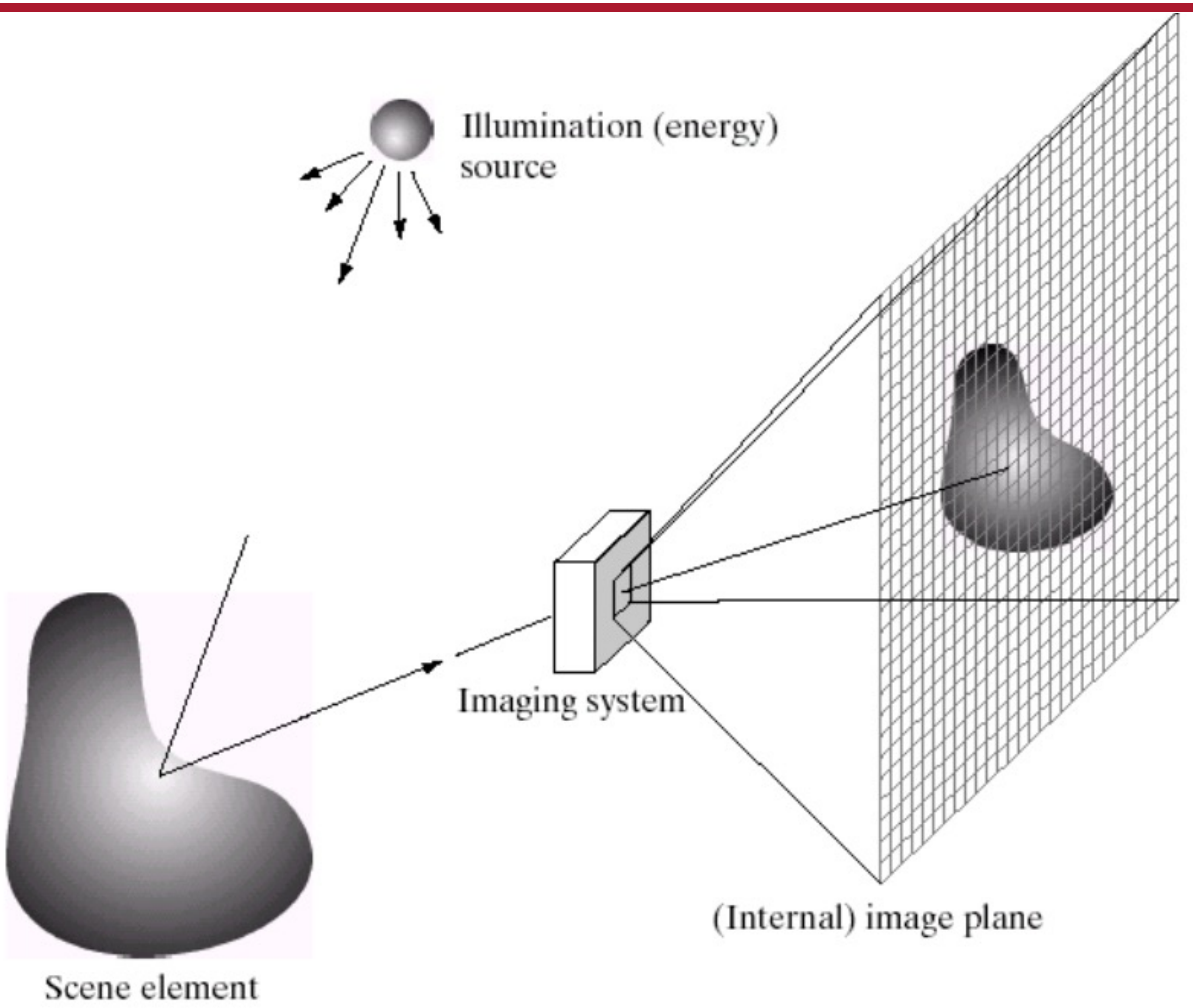
https://en.wikipedia.org/wiki/HSL_and_HSV

Canonical Image Processing problems

- 1. Image Filtering *(Apply filters or convolution kernels)*
- 2. Image Enhancement *(Adjust contrast, brightness, color balance)*
- 3. Image Restoration *(Denoising, Deblurring)*
- 4. Image Segmentation *(Partition an image)*
- 5. Image Registration *(Align images in a scene)*
- 6. Object Detection *(Identify objects or region of interest "ROI")*
- 7. Image Compression *(Reduce size - "JPEG, HEIF, MPEG")*
- 8. Image Recognition *(Classifying objects or patterns)*
- 9. Image Super-Resolution *(Increase the resolution)*
- 10. Image Synthesis *(Generate new images)*

What is an image?

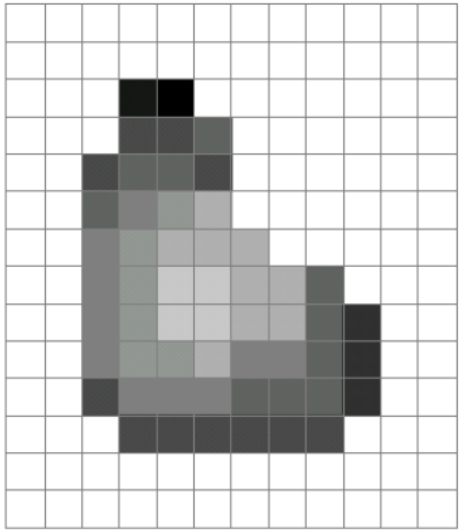
What is an image?



https://en.wikipedia.org/wiki/Digital_imaging

What is an image?

- A grid (matrix) of intensity values



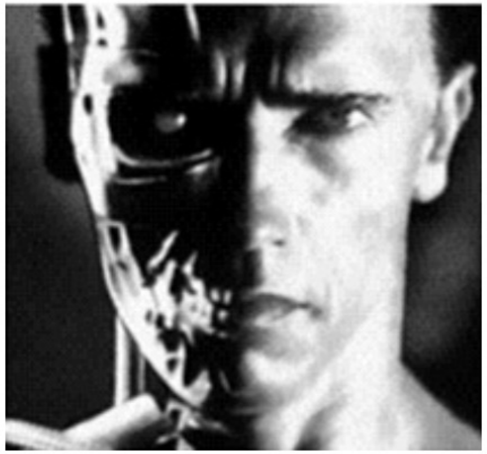
=

255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

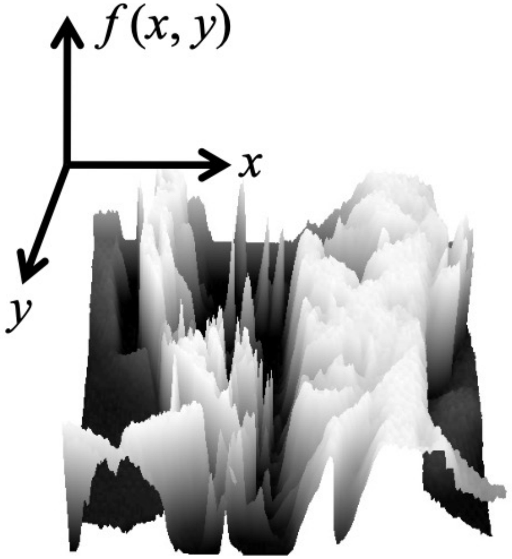
  – (common to use on byte per value: 0 = black, 255 = white)

What is an image?

- Can think of a (grayscale) image as a **function** *f* from R^2 to R:
  - $f(x,y)$ gives the **intensity** at position (x,y)



snoop

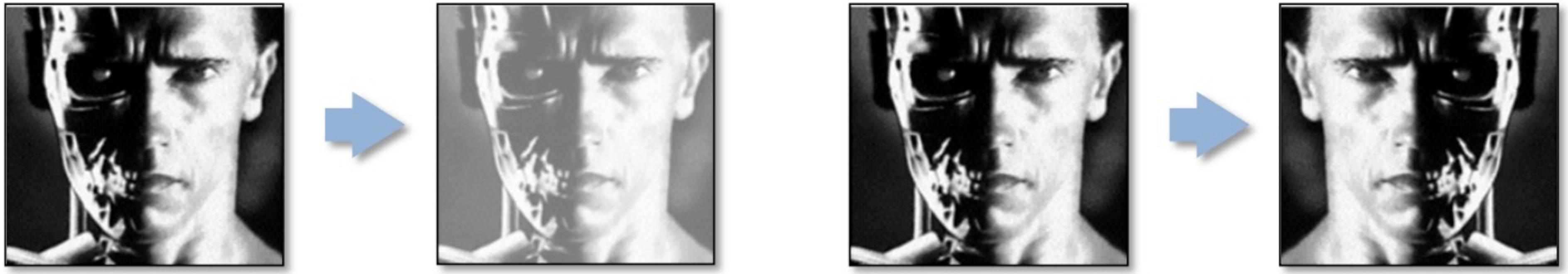


3D view

  - A **digital** image is a discrete (**sampled**, **quantized**) version of this function

Image transformations

- As with any function, we can apply operators to an image



$g(x,y) = f(x,y) + 20$ $\qquad$ $g(x,y) = f(-x,y)$

- Today we'll talk about a special kind of operator, *convolution* (linear filtering)

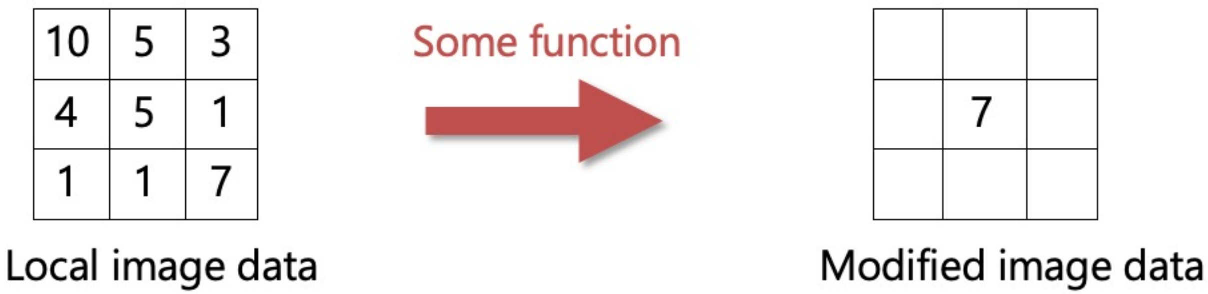
Filters

- Filtering
  - Form a new image whose pixel values are a combination of the original pixel values
- Why?
  - To get useful information from images
    - E.g., extract edges or contours (to understand shape)
  - To enhance the image
    - E.g., to remove noise
    - E.g., to sharpen and "enhance image"

https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html

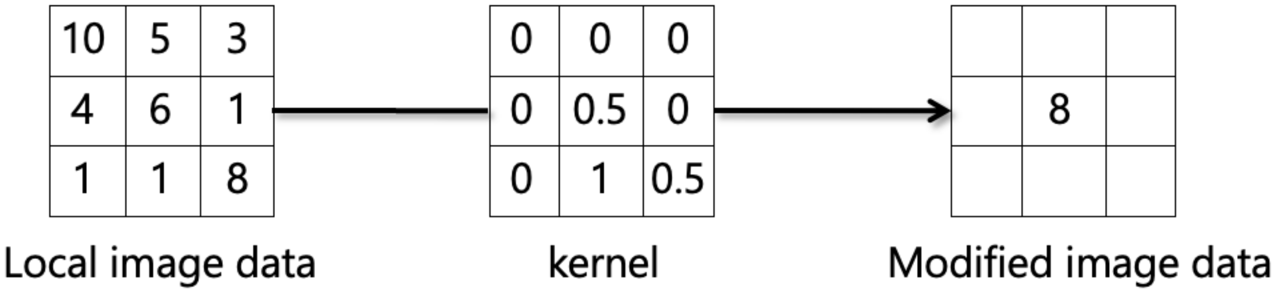
Image filtering

- Modify the pixels in an image based on some function of a local neighborhood of each pixel



Linear filtering

- One simple version of filtering: linear filtering (cross-correlation, convolution)
  - Replace each pixel by a linear combination (a weighted sum) of its neighbors
- The prescription for the linear combination is called the “kernel” (or “mask”, “filter”)



Cross-correlation

- Let F be the image, H be the kernel (of size 2*k*+1 x 2*k*+1), and G be the output image

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v] F[i+u, j+v]$$

This is called a **cross-correlation** operation:

$$G = H \otimes F$$

- Can think of as a “dot product” between local neighborhood and kernel for each pixel

Convolution

- Same as cross-correlation, except that the kernel is "flipped" (horizontally and vertically)

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

This is called a **convolution** operation:

$$G = H * F$$

- Convolution is **commutative** and **associative**

Mean filtering

H

$*$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

F

$=$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

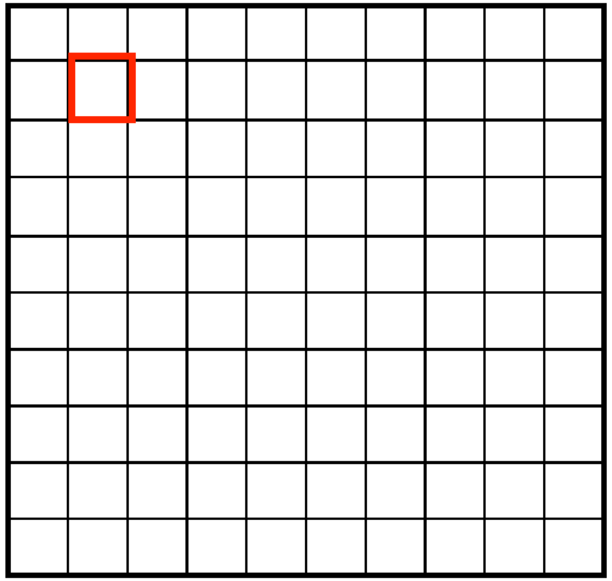
G

Mean filtering / Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$



Mean filtering / Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10							

Mean filtering / Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20						

Mean filtering / Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30					

Mean filtering / Moving average

$$F[x, y]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$G[x, y]$$

	0	10	20	30	30				

Mean filtering / Moving average

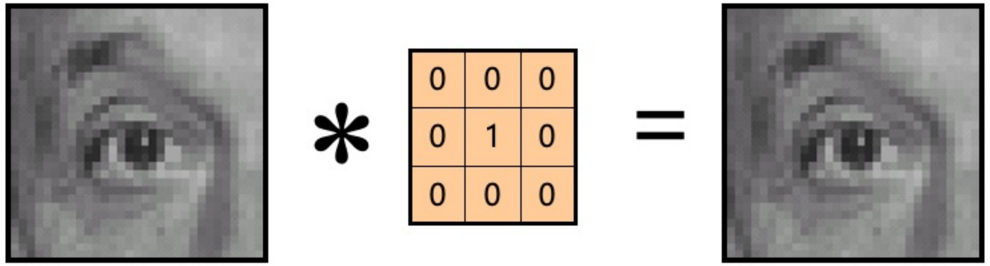
$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Linear filters - Examples



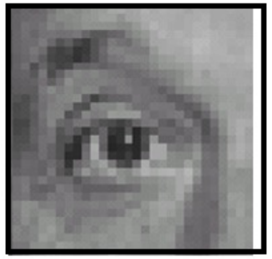
Original

Identical image

Linear filters - Examples

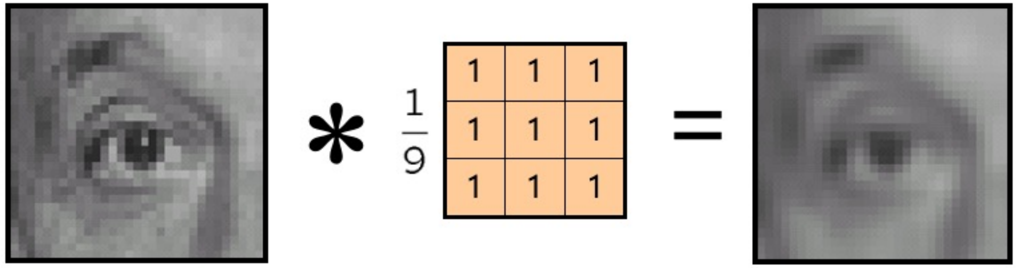
Original

$$* \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$



Shifted left by 1 pixel

Linear filters - Examples



Original

Blur (with a mean filter)

https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html

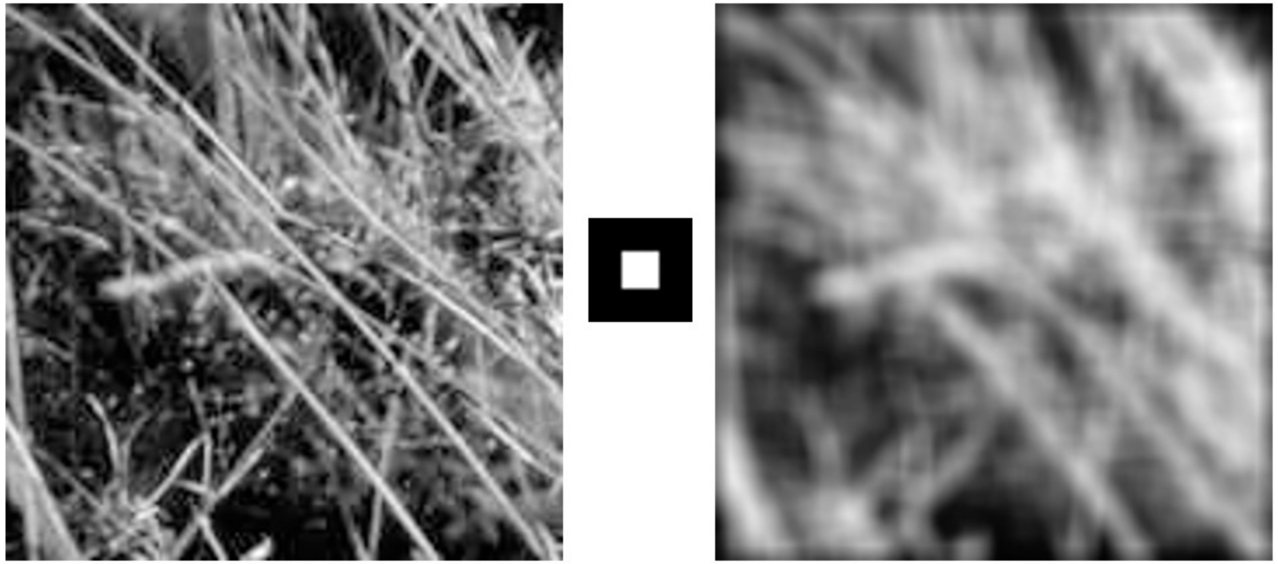
Linear filters - Examples

Original

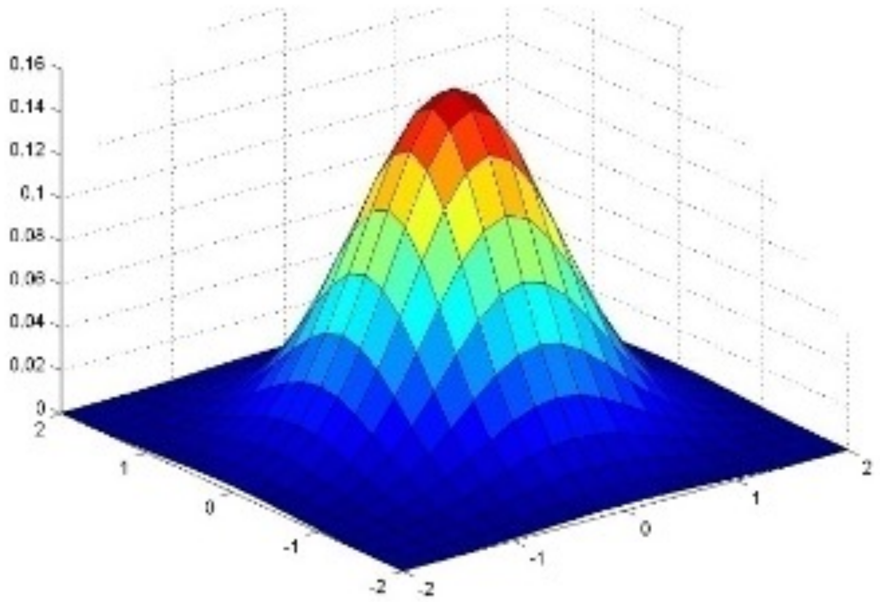
$$* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) =$$

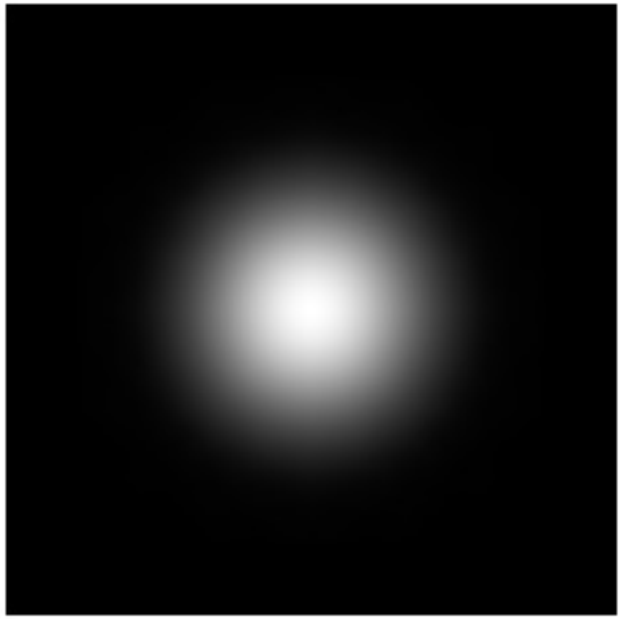
Sharpening filter
(accentuates edges)

Smoothing with box filter revisited



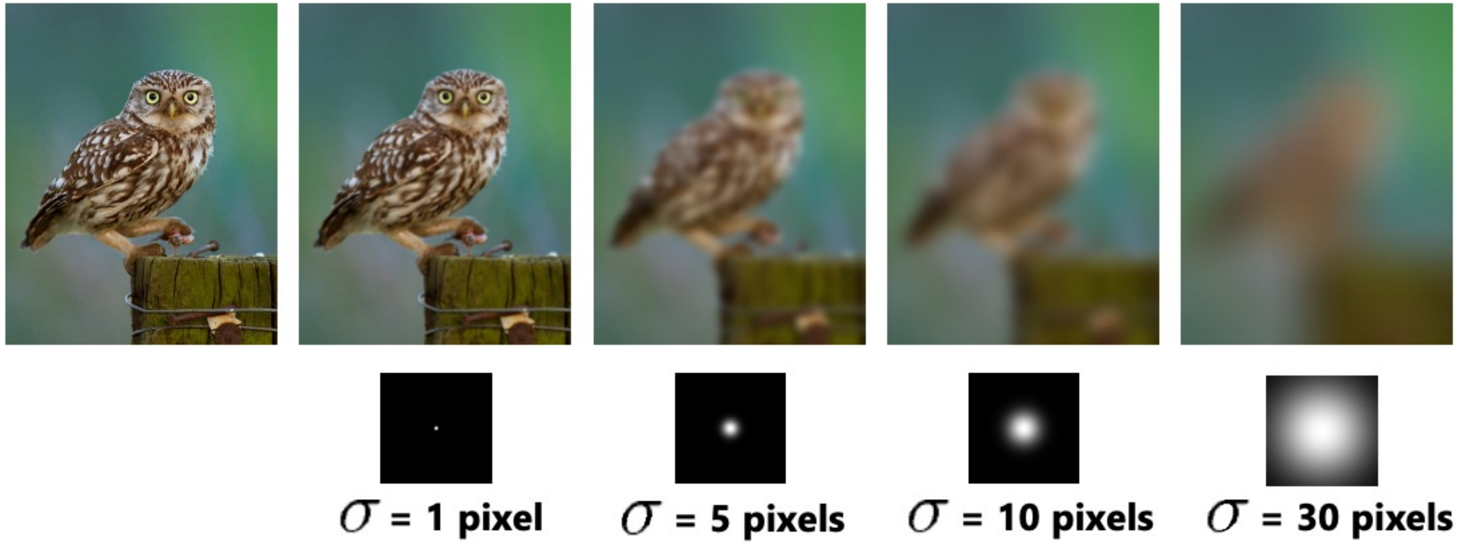
Gaussian kernel





$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Gaussian filters



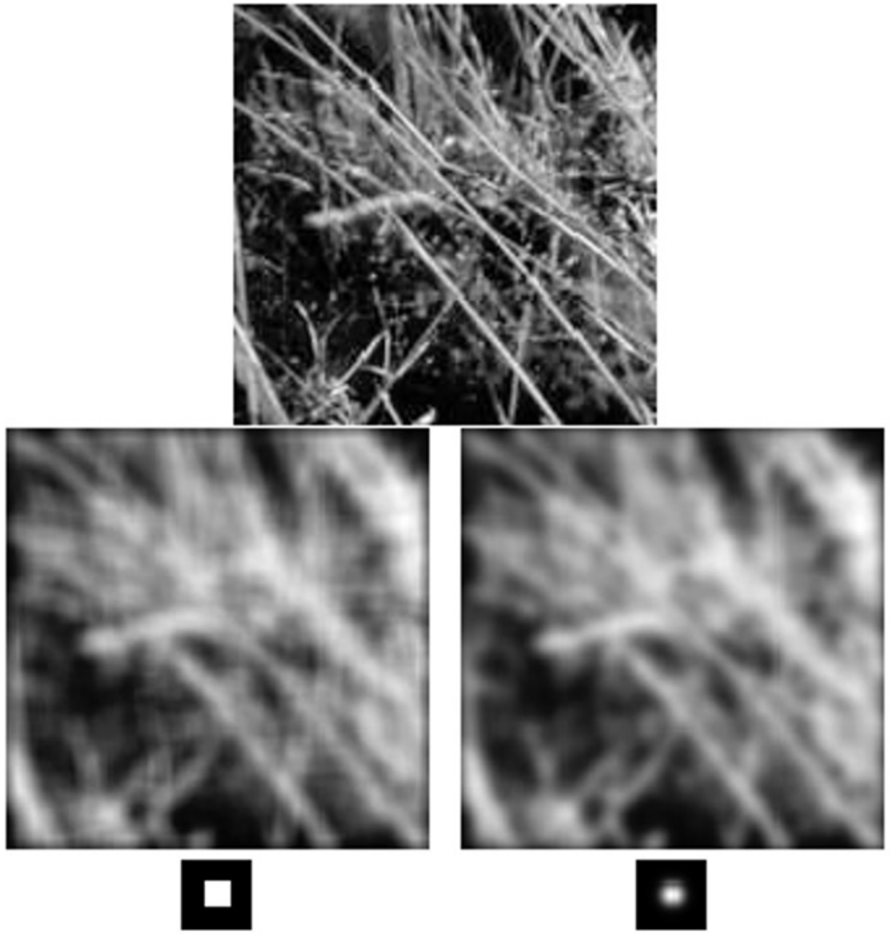
https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

Gaussian filter

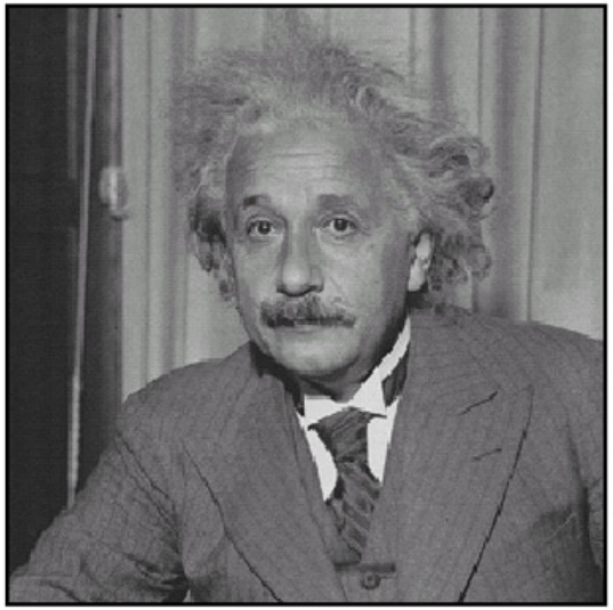
- Removes "high-frequency" components from the image (low-pass filter)
- Convolution with self is another Gaussian

- Convolving twice with Gaussian kernel of width σ
  = convolving once with kernel of width $\sigma\sqrt{2}$

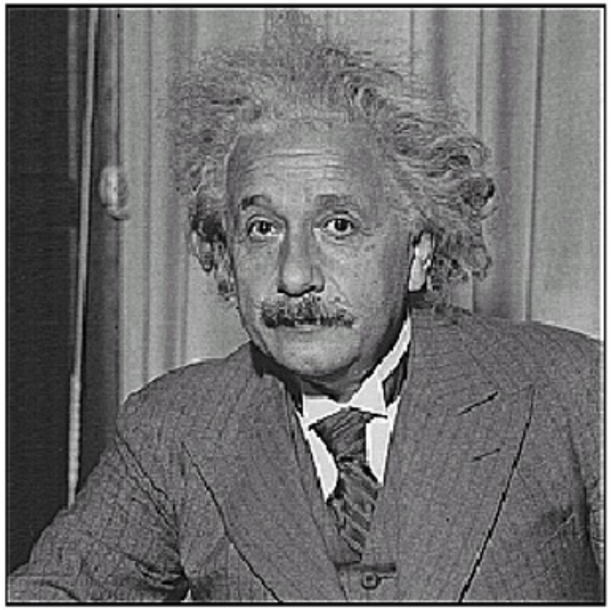
Mean vs. Gaussian



Sharpening



before



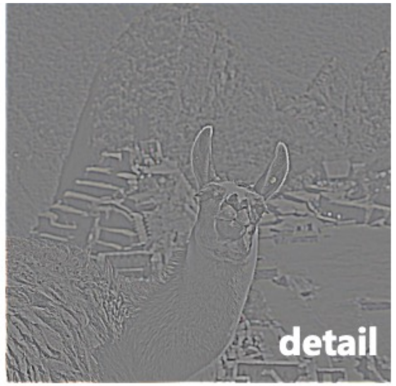
after

Sharpening revisited

- What does blurring take away?

–

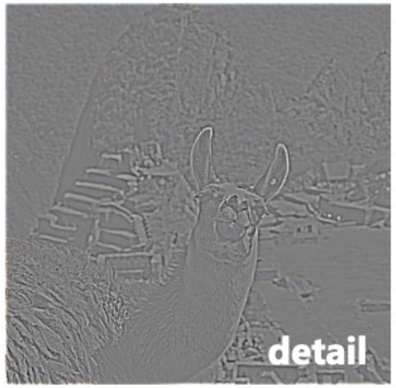
=



(This "detail extraction" operation is also called a ***high-pass filter***)

Let's add it back:

+ α



=

Thresholding filter

$$g(m,n) = \begin{cases} 255, & f(m,n) > A \\ 0 & otherwise \end{cases}$$

... end of Lecture 1