COMPUTER VISION

•••

Session - 1, Winter School '16

Introduction and Basic Operations

Some Fun Facts

- 40% of all nerve fibers connected to the brain are linked to the retina.
- More of our neurons are dedicated to vision than the other four senses *combined*.
- 2/3rds of the brain is used for processing visual data

So What is CV?

Teaching computers how to make sense of images

Robotic Vision

Computer
Intelligence

Control
Robotics

Multi-variable SP

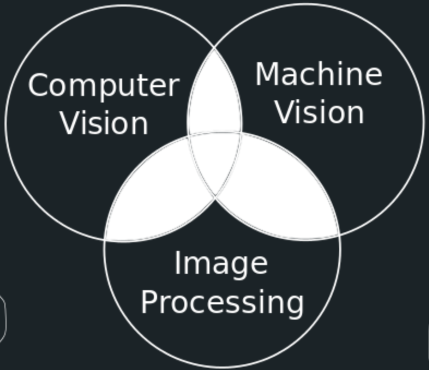
Non-linear SP

Artificial
Intelligence

Signal Processing

Cognitive
Vision

Machine
Learning



Physics

Optics

Statistics

Mathematics

Geometry

Imaging

Smart
Cameras

Optimization

There is SOMETHING for ALL !!

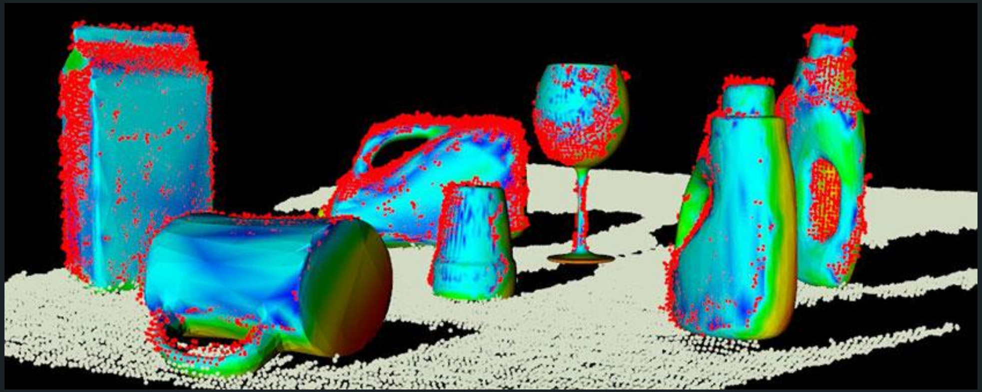
Neurobiology

Biological
Vision

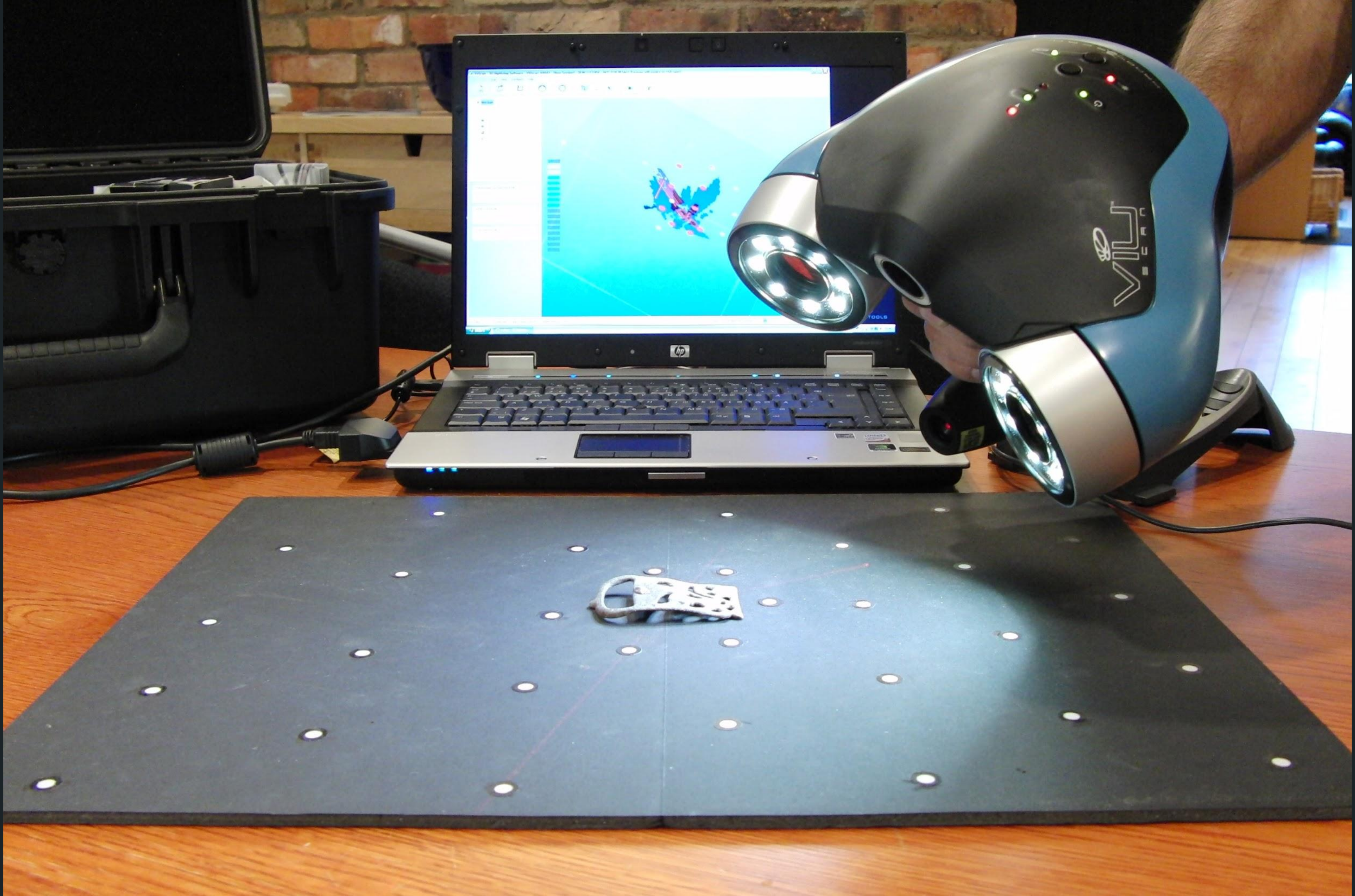
Where is CV used?

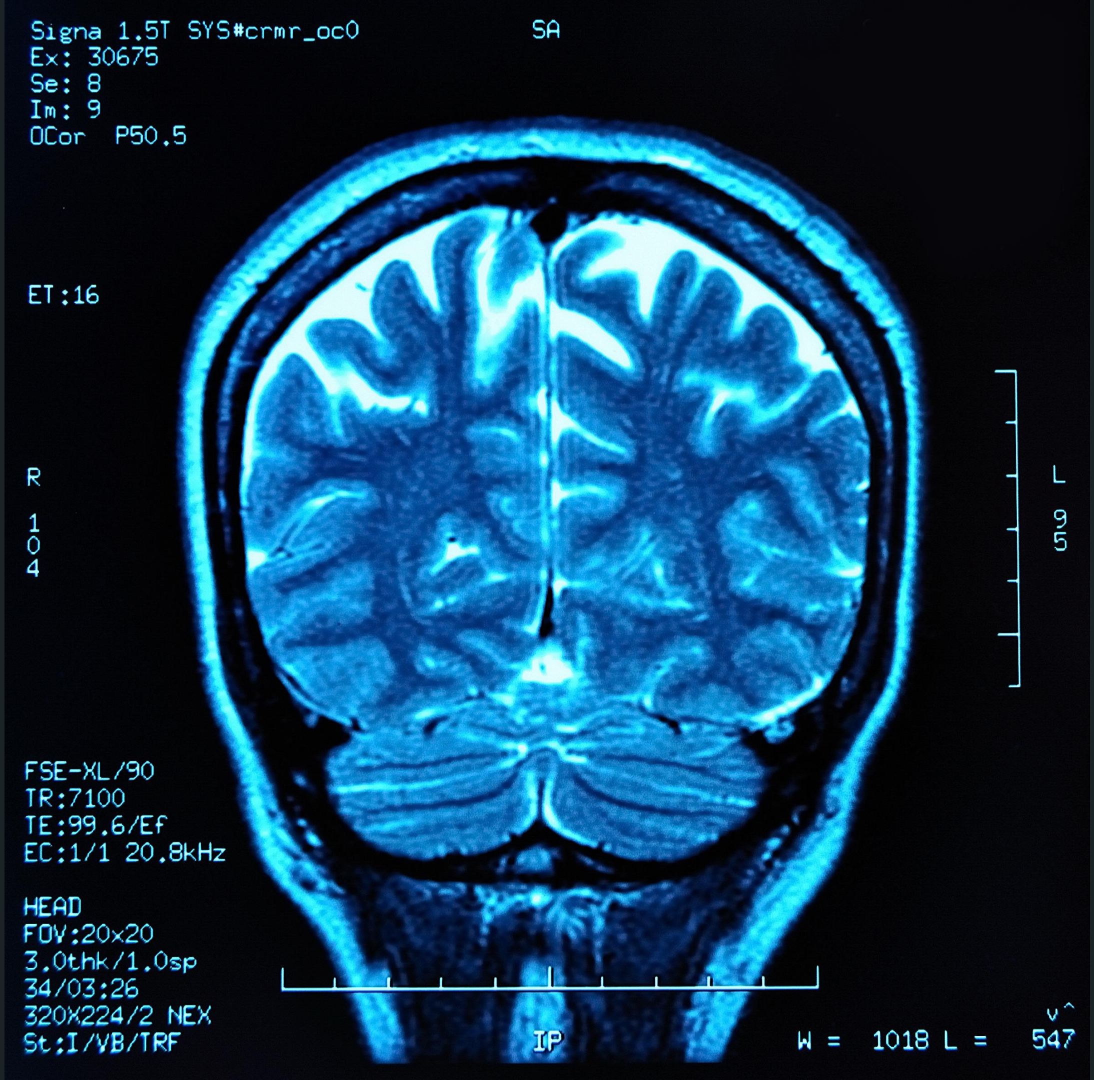
- In Robotics
- Autonomous Cars
- In cameras
- In face recognition
- Image search
- Gesture Recognition,etc
- Cam Scanner

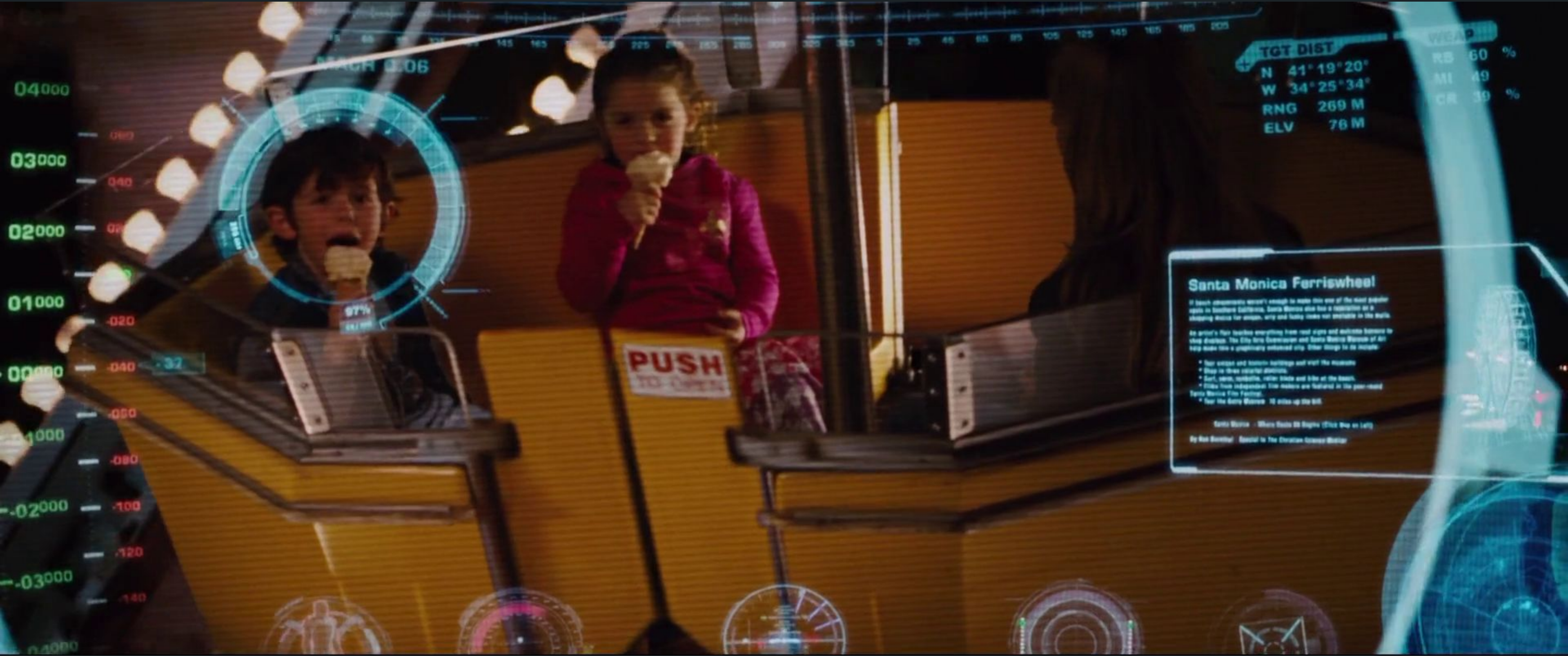

UBER
UBER



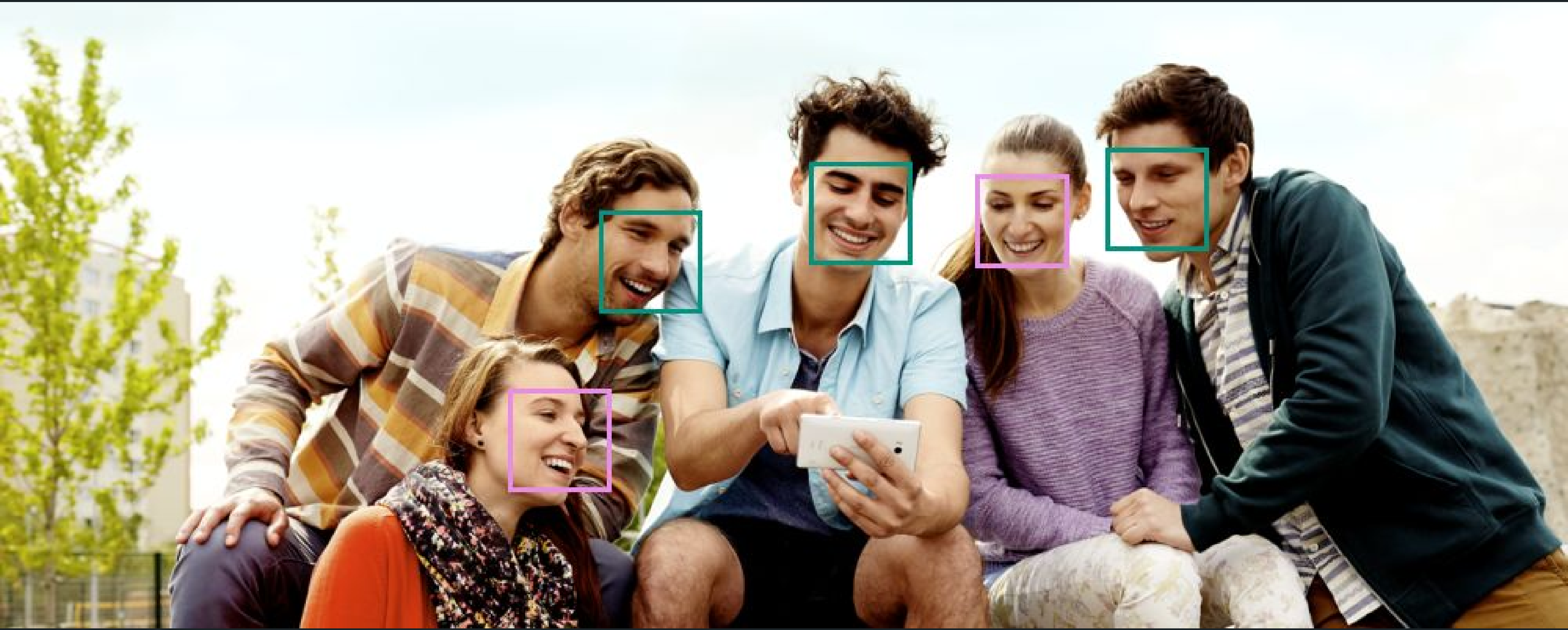
CAMERA
Horsea
CP 197

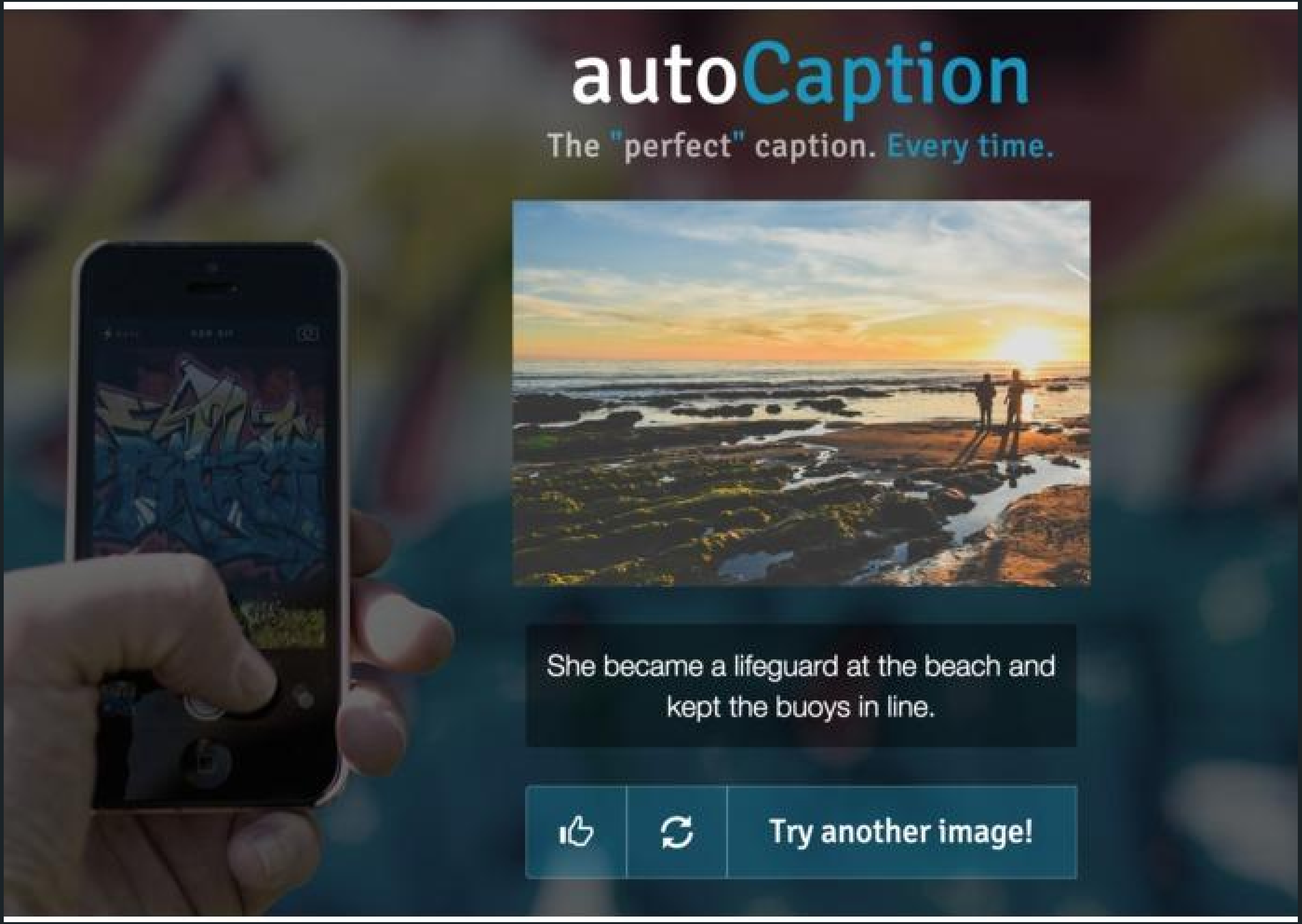

VIU


Signa 1.5T SYS#crmr_oc0
SA
Ex: 30675
Se: 8
Im: 9
OCor P50.5
ET:16
R
104
L
95
FSE-XL/90
TR:7100
TE:99.6/Ef
EC:1/1 20.8kHz
HEAD
FOV:20x20
3.0thk/1.0sp
34/03:26
320X224/2 NEX
St:I/VB/TRF
IP
W = 1018 L = 547


MACH 0.06
TGT DIST
N 41°19'20"
W 34°25'34"
RNG 269 M
ELV 76 M
04000
03000
02000
01000
PUSH
TO OPEN
Santa Monica Ferriswheel

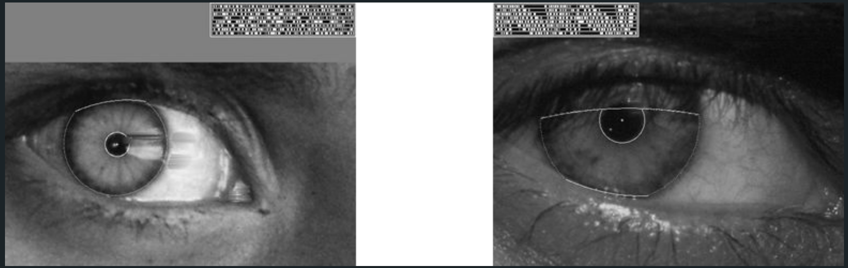

KINECT
Microsoft
재활병원과 함께
장소: 재활병원 3층




autoCaption
The "perfect" caption. Every time.
She became a lifeguard at the beach and kept the buoys in line.
Try another image!

Vision based biometrics

“*How the Afghan Girl was Identified by Her Iris Patterns*” Read the story
wikipedia



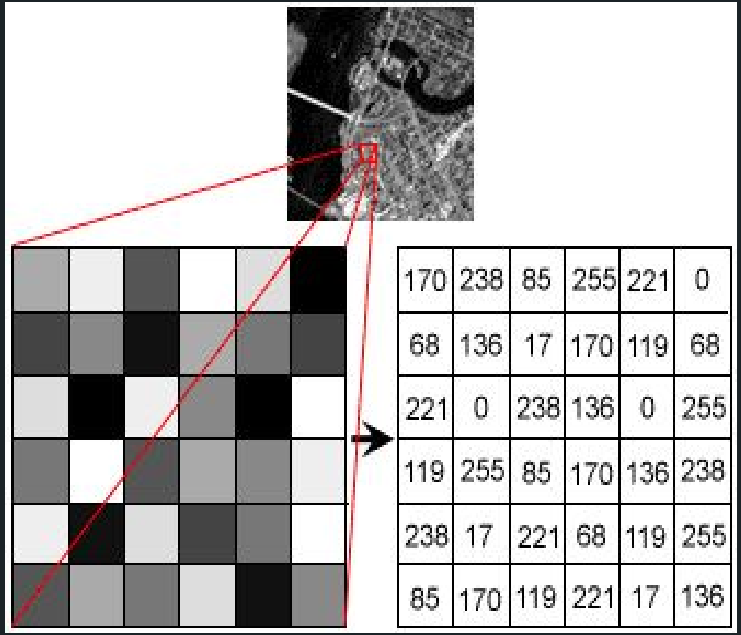
Login without a password…

Fingerprint scanners on
many new laptops,
other devices

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

Composition of an Image

- Images are a matrix of pixels, and pixels are numbers
- Black and white images contain pixels, which hold only one value, while RGB images have pixels that contain values for Red, Green and Blue composition.
- 1080p is actually 1920x1080 pixels (Full HD) (aspect ratio = 16:9, widescreen))



GrayScale

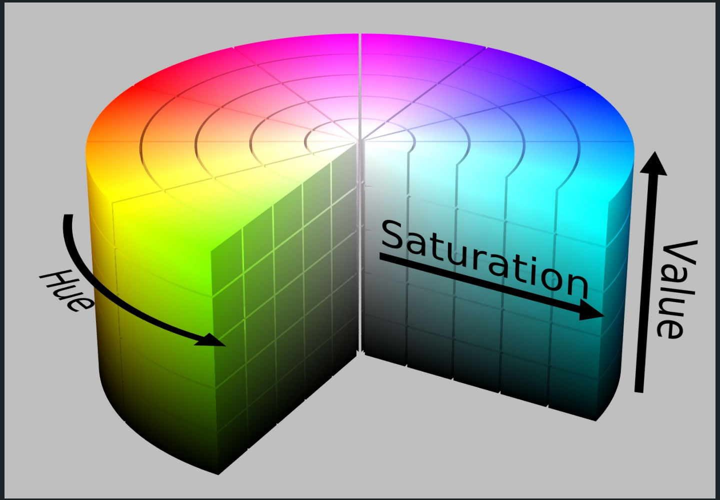
- Each pixel is a 8 bit number
- It can take values from 0-255
- Each value corresponds to a shade between black

  and white(0 -black,255-white)
- Number of channels for a grayscale image is 1
- Depth of a grayscale image is 8(bits)

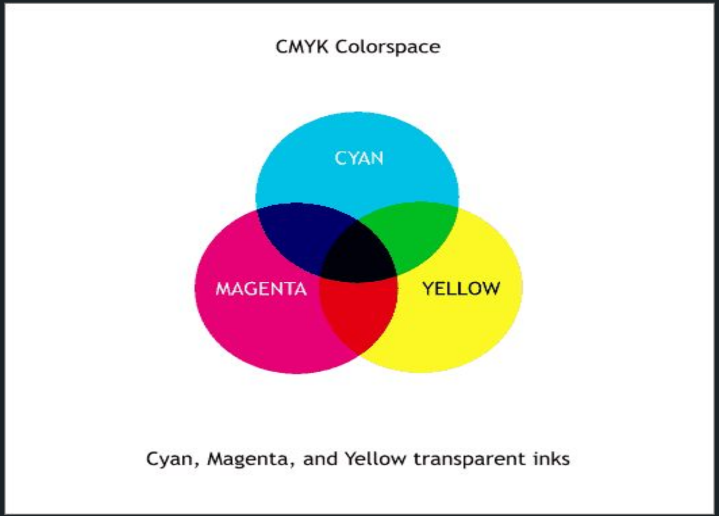
RGB

- Each pixel stores three values:
    1. R : 0-255
    2. G : 0-255
    3. B : 0-255
- Each number between 0-255 corresponds to a shade of corresponding color
- Depth of a RGB image is 8(bits)
- Number of channels for a RGB image is 3

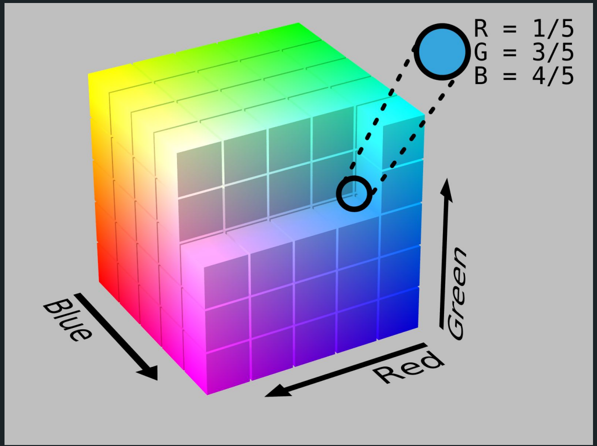
- HSV (Hue, Saturation and value)



- CMYK (Cyan Yellow Magenta Key) is a subtractive model (printing)



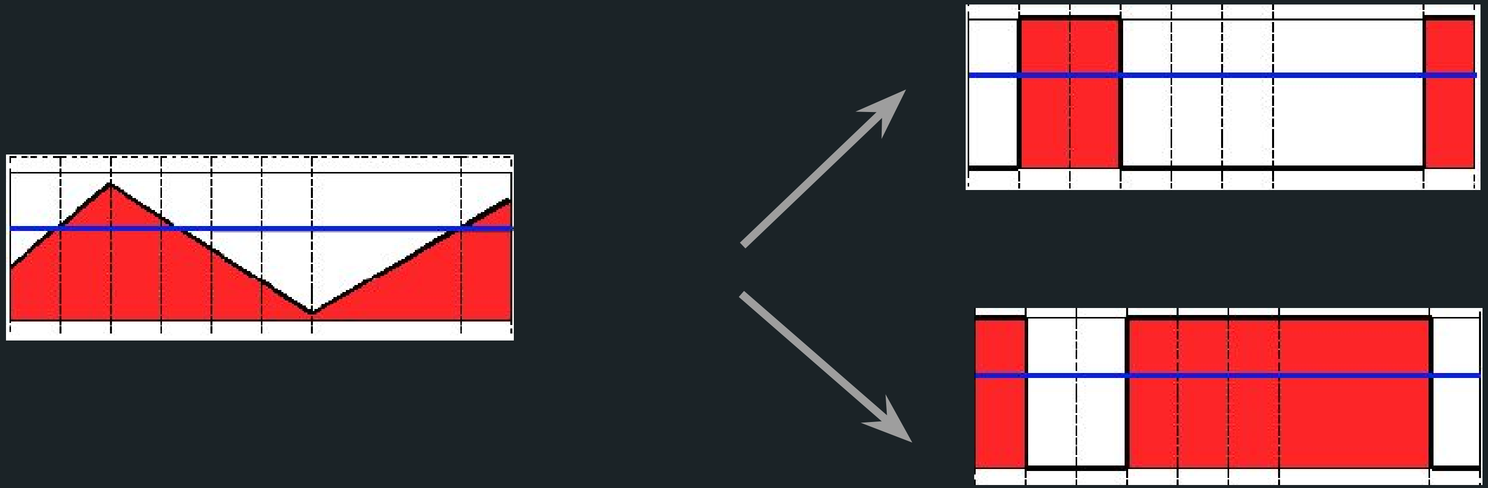
- RGB (Red, Green and Blue), is an additive color model.



THRESHOLDING

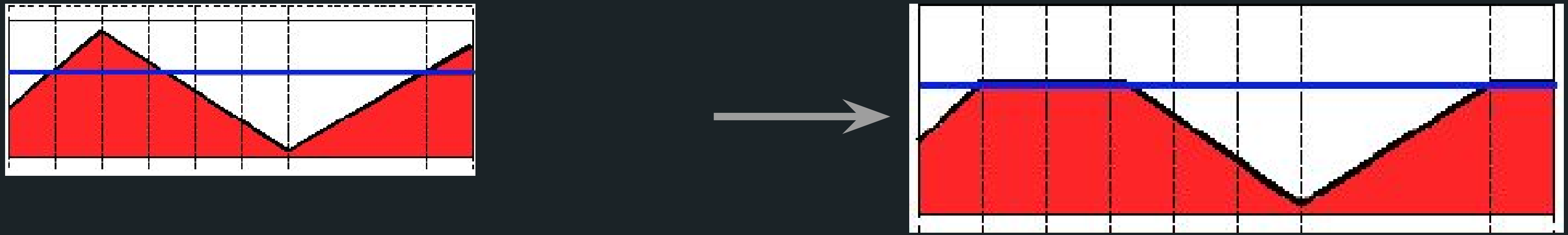
Simple Thresholding

Binary Threshold



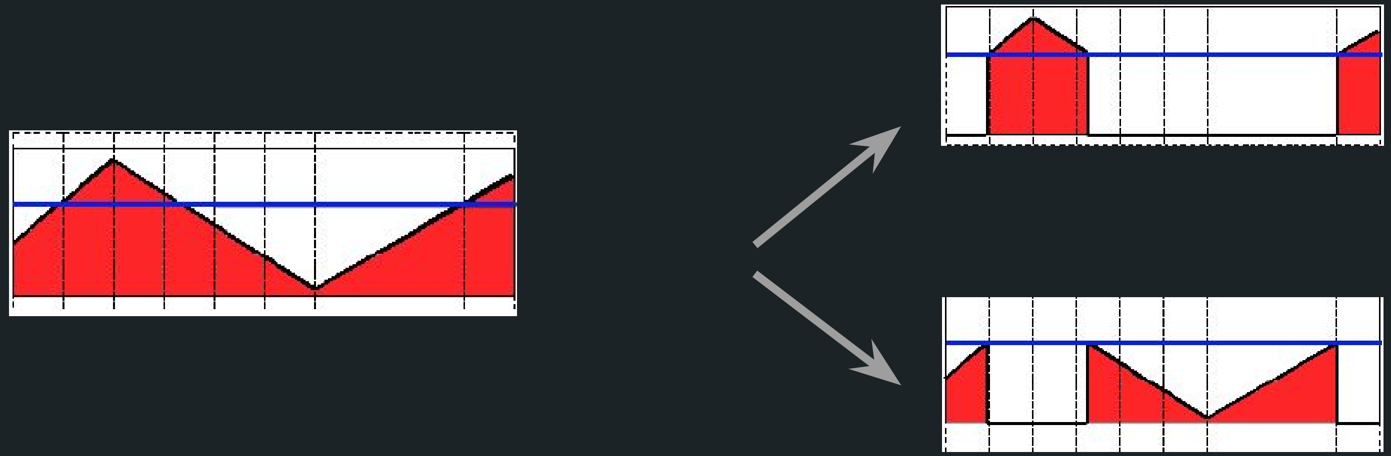
Simple Thresholding

Truncate



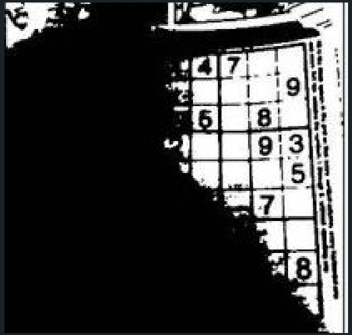
Simple Thresholding

Truncate to zero or invert



Non Uniform Illumination

Global Thresholding for non uniformly illuminated image



Give me My Image back!! :-(

ADAPTIVE THRESHOLDING

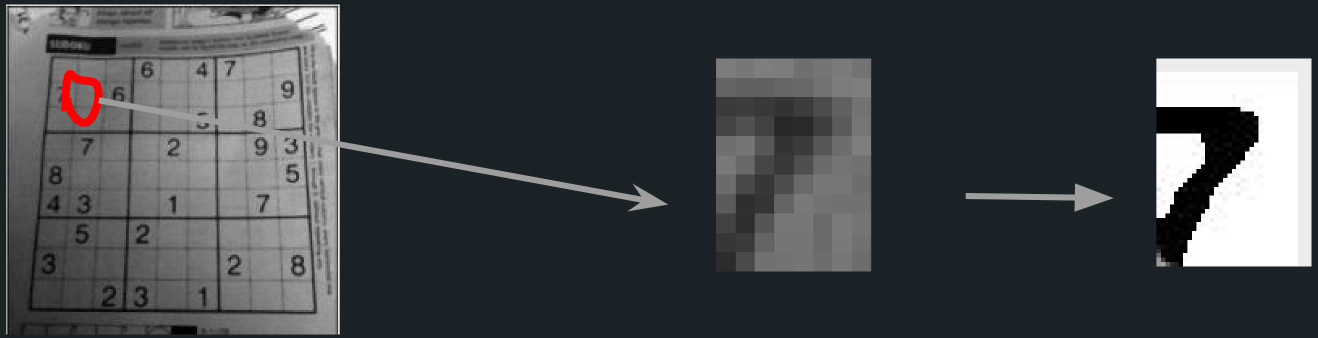
Why Adaptive Thresholding??

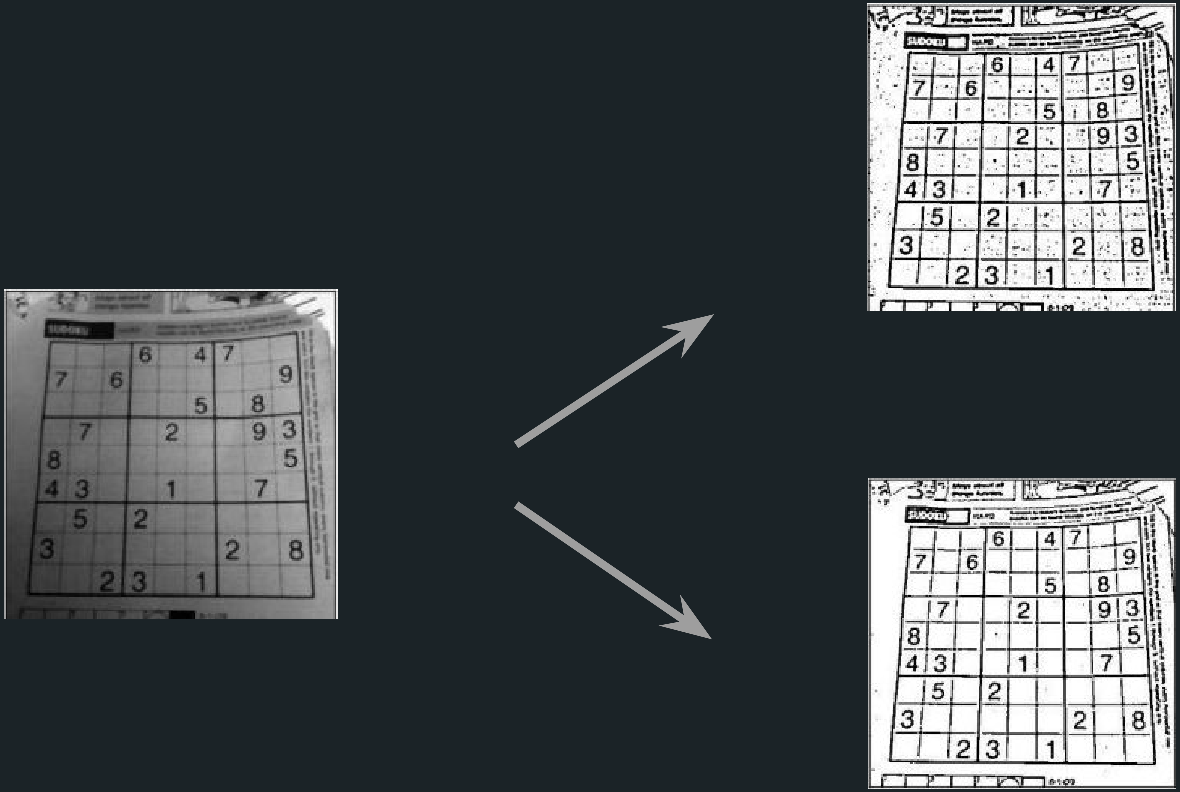
There was a single threshold for all pixels of the image in the previous case.

In this, the algorithm calculate the threshold for a small regions of the image.

In this way we get threshold for a neighbourhood.

Global Thresholding for the rounded part gives almost a clear 7.





Adaptive Mean

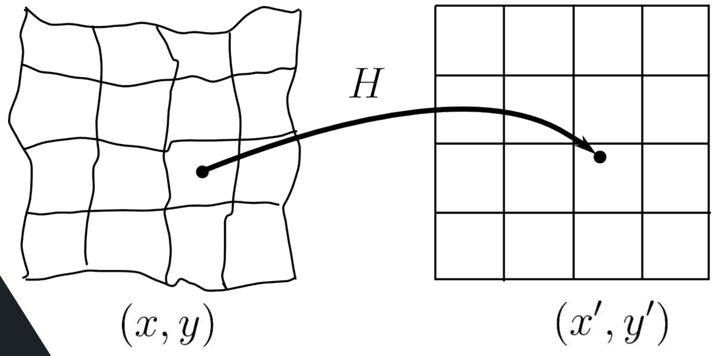
Adaptive Gaussian

In Adaptive mean thresholding, arithmetic mean of the neighbourhood is taken as the threshold for that neighbourhood.

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

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

In Gaussian Thresholding, the mean of the neighbourhood is taken with Gaussian matrix as weight.



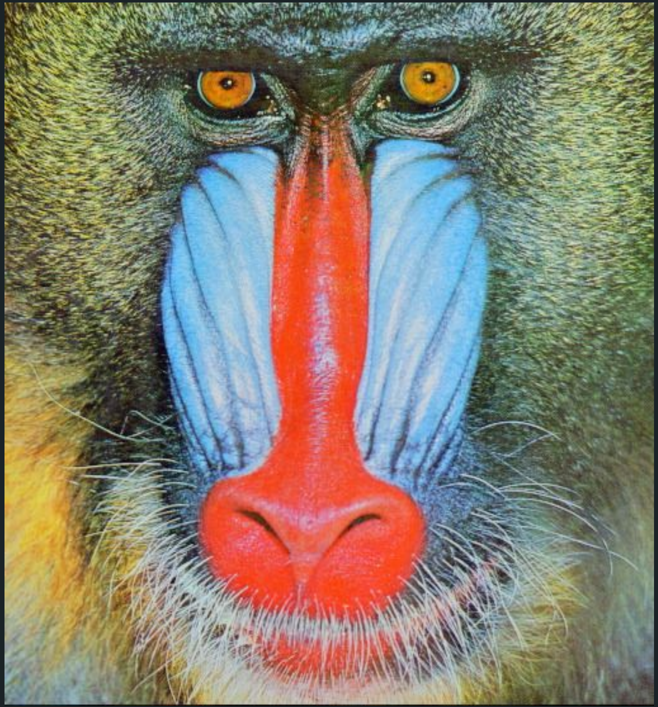
GEOMETRIC TRANSFORMATIONS

Image Scaling

How are images resized??

Original image

Image Translation



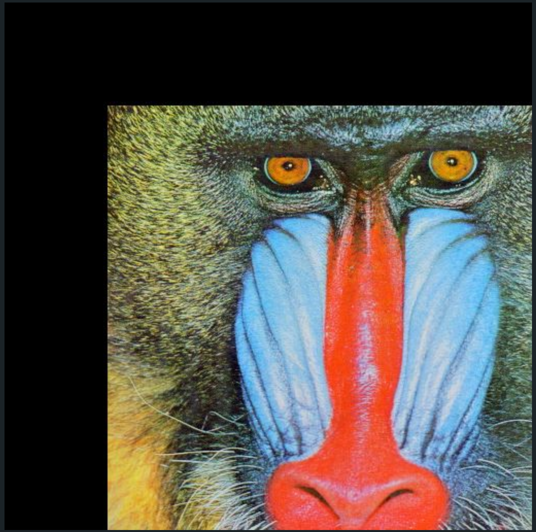
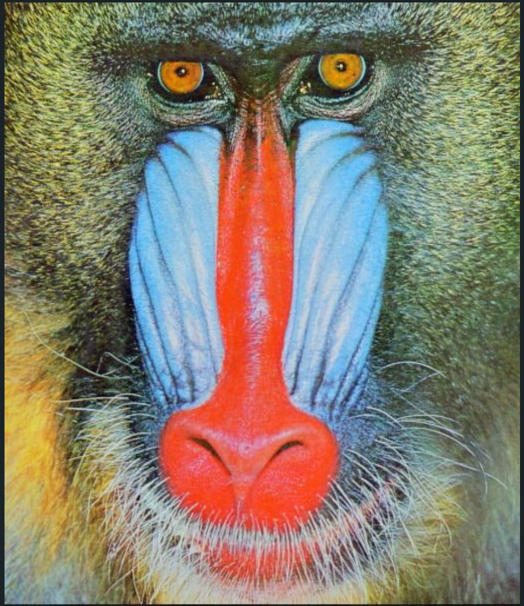


Image Rotation



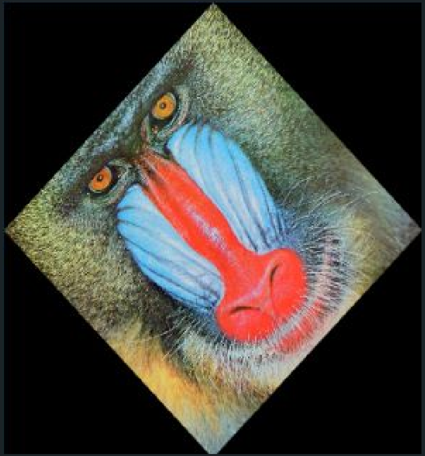
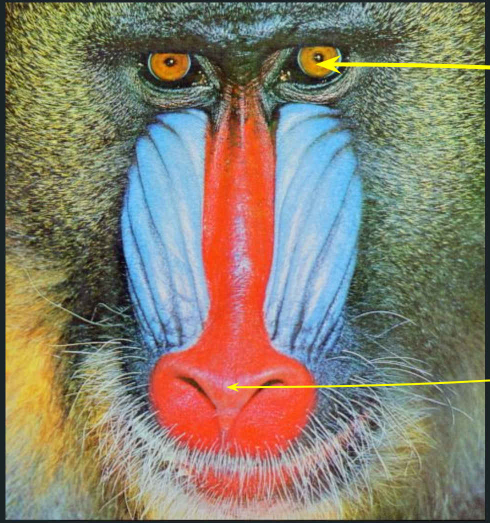
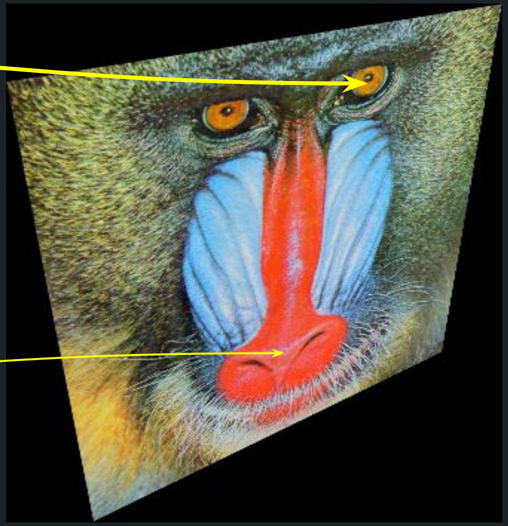


Image Warping

Pure warping means that points are mapped to points without changing the colors.

Example





TRY

How can you match the two images?

CONVOLUTION

Making your own linear filters!

- Effects like the ones you might find in Photoshop or Gimp
- Ex: blurring, sharpening, outlining or embossing

Convolution: Trick of Image Filtering

- Convolution: general purpose filter effect for images.
- A mathematical operation between every part of an image and an operator (kernel)
- The output is a new modified filtered image

The process of image convolution

- Done by adding the weighted values of all its neighbors together
- Kernel:
  - Small matrix of numbers, say a 3x3, or 5x5 matrix (Why small?, Why odd size?)
  - The 2D filter matrix used to apply effects like blurring, sharpening, outlining
  - Example of a kernel:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Let’s Convolve with this Kernel

2	3	-2
4	5	-4
-4	-4	2

$$g(i,j) = \sum_{k,l} f(i+k, j+l)h(k,l)$$

Let's Convolve the Kernel with this

5	4	25	35	20
60	45	40	30	10
11	60	80	15	5
80	22	20	10	1
200	140	50	0	8

Let's Convolve the Kernel with this

2 5	3 4	-2 25	35	20
4 60	5 45	-4 40	30	10
-4 11	-4 60	2 80	15	5
80	22	20	10	1
200	140	50	0	8

45 is replaced with 2*5 + 3*4 + (-2)*25 + 4*60 + 5*45 + (-4)*40 + (-4)*11 + (-5)*60 + 2*80

Ensuring Values are within range

What if values exceed 255 or drop below 0?

Truncation: Output more than 255 is taken as 255, and output with negative value is taken as 0 or as its absolute value

Normalization:

Dividing each element in the kernel by the sum of all elements

Normalized kernel gives output image with same brightness as input image

What to do with edge pixels?

- Use “zeros” as the neighbour pixels that aren’t there, i.e. pad with zeros
- Wrap around the image to the other side
- Extend the nearest border pixels as far as necessary
- Crop, i.e. don’t consider edge pixels. Output image is slightly smaller

Try it Out!

What happens on convolving with this kernel?

0	0	0
1	0	0
0	0	0

Try it Out!

What happens on convolving with this kernel?

0	**0**	**0**
1	**0**	**0**
0	**0**	**0**

Displaces the image

Try it Out!

What happens on convolving with this kernel?

0	-1	0
-1	5	-1
0	-1	0

Try it Out!

What happens on convolving with this kernel?

0	-1	0
-1	5	-1
0	-1	0

Creates a sharpened image. (Convince yourself)

Try it Out!

What happens on convolving with this kernel?

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

Try it Out!

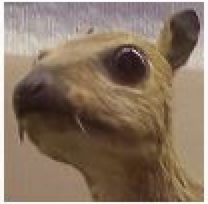
What happens on convolving with this kernel?

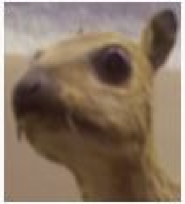
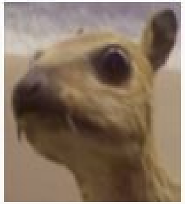
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

This causes Blurring.
The above kernel is normalized Box Filter

Blurring

- Also called *Smoothing*
- Important in image processing
- The different kinds of filters for blurring:
  - Simple Blur (Normalized box filter) : Mean of kernel neighbors
  - Gaussian Blur: Uses Gaussian kernel
  - Median Blur: Median of neighbor pixels in kernel
  - Bilateral Filter : Gaussian function of space and intensity (preserves edges)

Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	

Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$