# Computer Vision

## Image Introduction
## Geometric Operations

Fátima Leal
2020/2021

DEPARTAMENTO CIÊNCIA E TECNOLOGIA



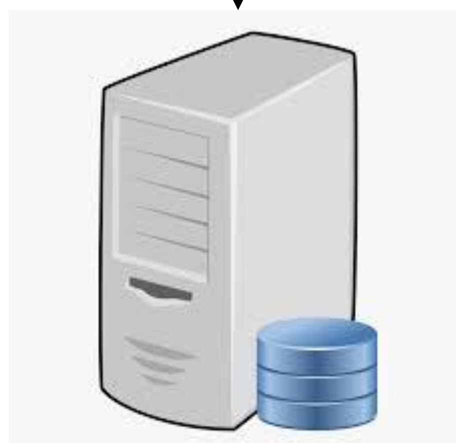
UNIVERSIDADE PORTUCALENSE

# Content

- Computer Vision Introduction
- Image concept
- Geometric operations
- Practical Examples

# Computer Vision Introduction

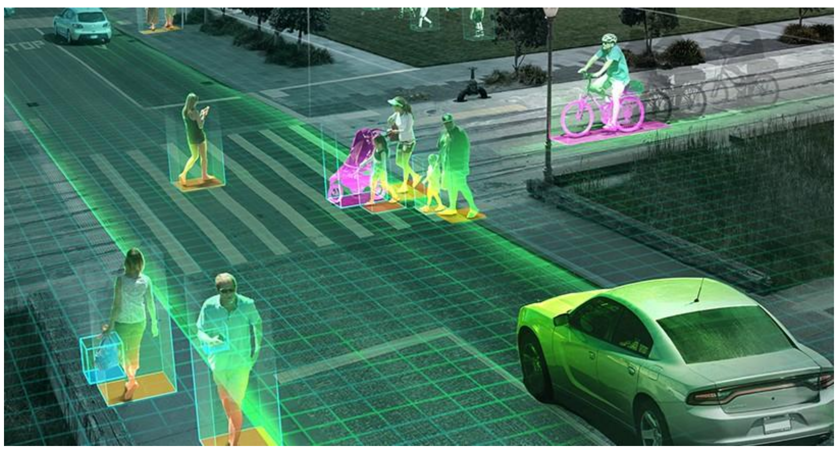

Give vision power to a machine generates an impressive result!!

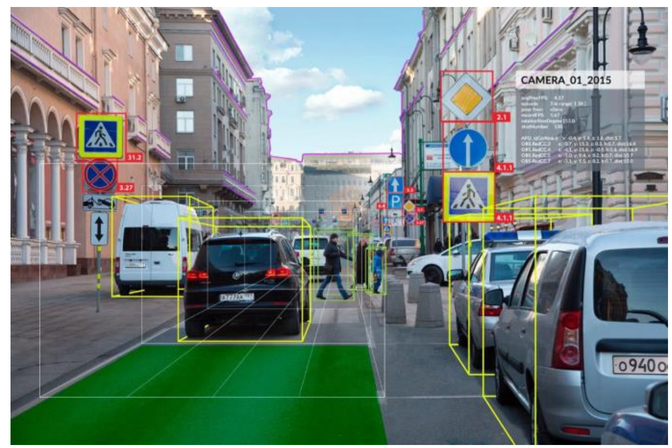


It is important to understand basic image concepts!

# Computer Vision Introduction

- **What is computer vision?**
  - A field of artificial intelligence that trains computers to interpret and understand the visual world
  - Can be applied to images or videos
  - Images are everywhere
  - Aims to extract information from images and interpret their contents





# Image concepts

- **What is an image?**
  - A **visual representation** of an object, a person, or a scene
  - Produced by an optical device such as a mirror, a lens, or a camera
  - **Two-dimensional representation** (2D)
  - One of the infinitely many projections of a real-world - **three-dimensional** (3D) **object or scene**.

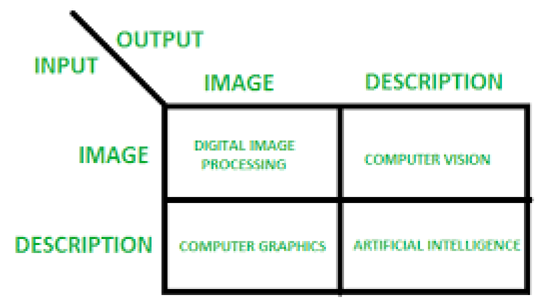
- **What is a digital image**?
  - A representation of a **two-dimensional image**
  - Uses a **finite number of points**: picture elements or **pixels**
  - Each pixel is represented by one or more numerical values:
    - **monochrome** (grayscale) images: a **single value** to represent the pixel intensity (usually in a [0, 255] range);
    - **colour images**: **three values** to represent the amount of red (R), green (G), and blue (B)



DEPARTAMENTO CIÊNCIA E TECNOLOGIA

# Image concepts

- **What is Digital Image Processing?**
  - The study of any algorithm that takes an image as input and returns an image as output
- **What's the goal?**
  - Changing the nature of an image in order to either:
    - improve its pictorial information for human interpretation
    - render it more suitable for autonomous machine perception
- **What does it include?**
  - Image display and printing
  - Image editing and manipulation
  - Image enhancement
  - Feature detection
  - Image compression



# Image concepts

**Images are changed automatically according to an algorithm as opposed to image manipulation using tools like Photoshop.**

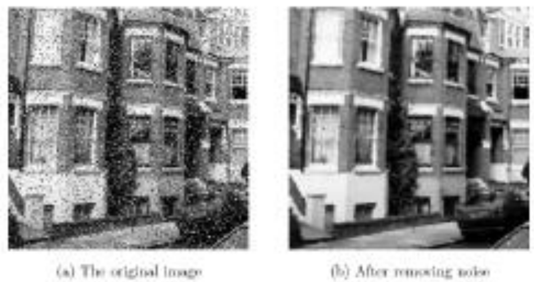
- Image Processing
- Image Analysis
- Computer Vision

# Image concepts

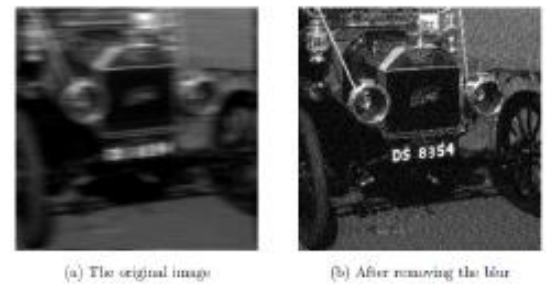
- **Image processing** covers a wide and diverse array of techniques and **algorithms**:
  - **Sharpening**
  - **Noise removal**
  - **Deblurring**
  - **Edge extraction**
  - **Binarization**
  - **Blurring**
  - **Contrast Enhancement**
  - **Object Segmentation and labelling**

# Image concepts


(a) Image sharpening.




(b) Removing noise.




(c) Image deblurring.

# Image concepts

(a) The original image (b) Its edge image

(a) Finding edges.

(a) The original image (b) Blurring to remove detail

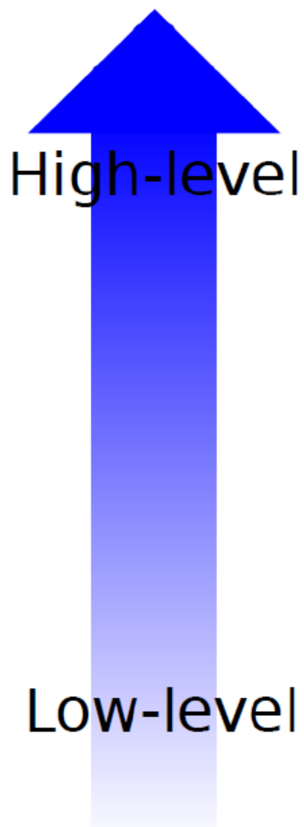
(b) Blurring an image.

# Image concepts

- **Image Analysis** involves extracting meaningful information from an image:
  - Image segmentation
  - Image matching and comparison
  - Medical diagnosis from an image

- **Computer Vision** aims to emulate the human visual system and interpret our 3D world from 2D images or video
  - Object recognition
  - Motion tracking
  - 3D shape from multiple 2D images

# Image concepts




## Computer Vision

Object detection, recognition, shape analysis, tracking
Use of Artificial Intelligence and Machine Learning

## Image Analysis

Segmentation, image registration, matching

## Image Processing

Image enhancement, noise removal, restoration, feature detection, compression



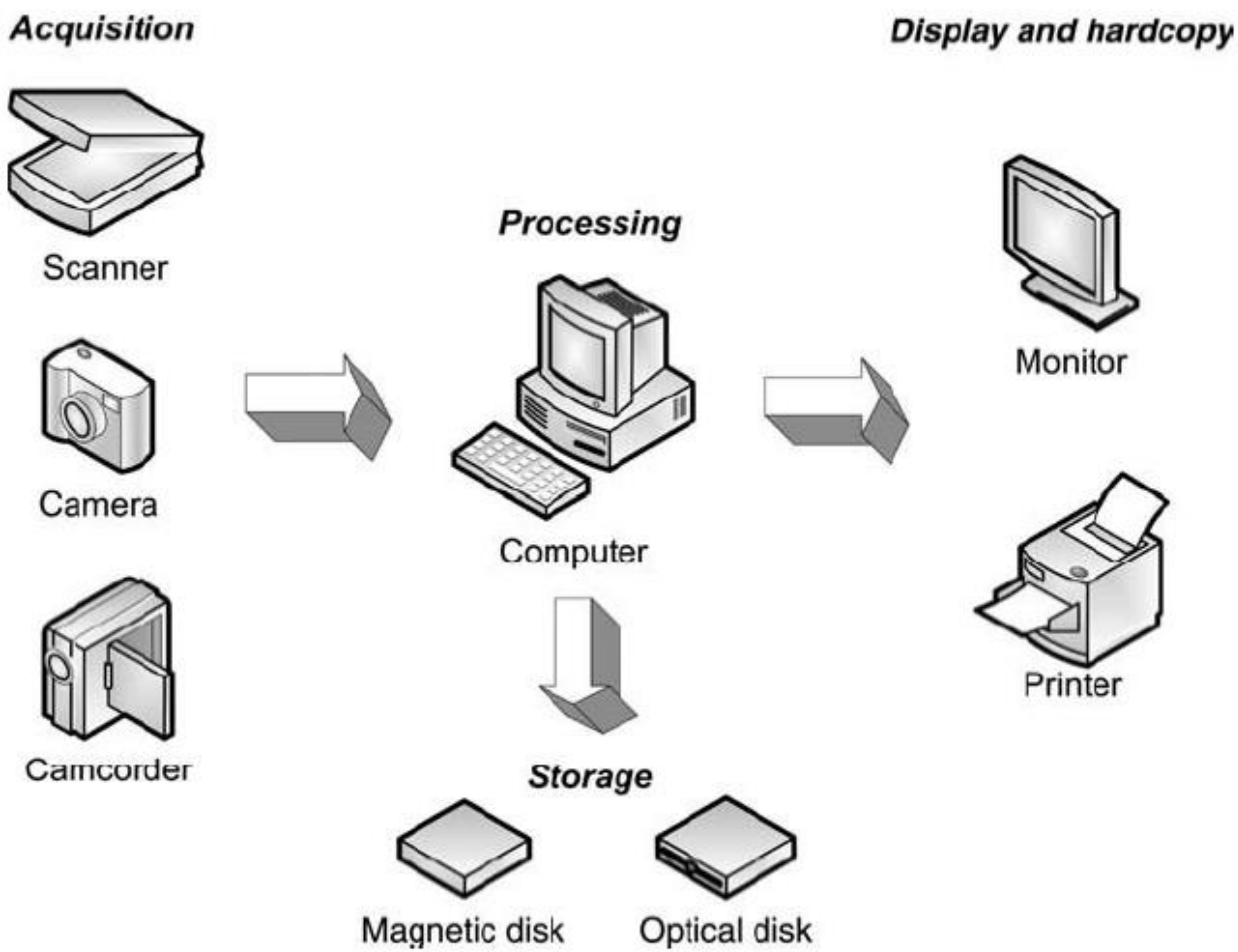
DEPARTAMENTO CIÊNCIA E TECNOLOGIA

# Image concepts

- **Three levels**:
  - **Low Level**: Primitive operations (*e.g.*, noise reduction, contrast enhancement, *etc.*) where both the input and the output are images.
  - **Mid Level**: Extraction of attributes (*e.g.*, edges, contours, regions, *etc.*) from images.
  - **High Level**: Analysis and interpretation of the contents of a scene

| Low Level Process | Mid Level Process | High Level Process |
|---|---|---|
| Input: Image | Input: Image | Input: Attributes |
| Output: Image | Output: Attributes | Output: Understanding |
| | | |
| Examples: Noise removal, image sharpening | Examples: Object recognition, segmentation | Examples: Scene understanding, autonomous navigation |

# Digital Image Processing System



# Digital Image Processing System

- A generic digital image processing system is built around a computer
- Includes hardware and software for image acquisition, storage, and display.
- **Hardware components**:
  - **Acquisition Devices**: Responsible for capturing and digitising images or video sequences.
  - **Processing Equipment**: Responsible for running software that allows the processing and analysis of acquired images.
  - **Display and Hardcopy Devices**: Responsible for showing the image contents for human viewing (monitors and printers) .
  - **Storage Devices**: Magnetic or optical disks responsible for long-term storage
- **Software**:
  - modules which perform specialised tasks.
  - In this course our choice is Python.

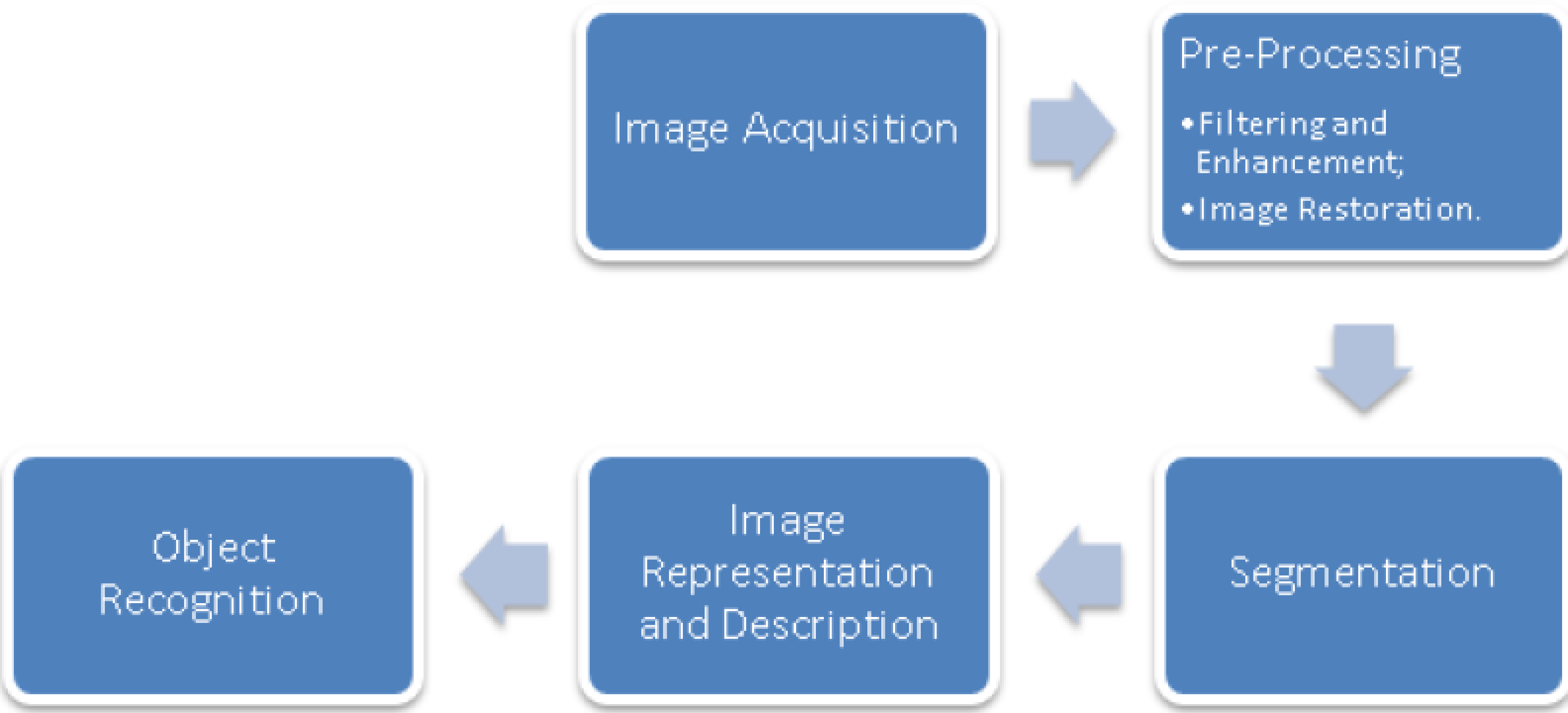
# Machine Vision System

- Let's take as a practical exemple: **recognising license plates at a highway toll booth**

# Machine Vision System

- Most solutions follow a sequential processing scheme:
  - **Image acquisition**: collection of one of more images
  - **Pre-processing**: improve the quality of those images
  - **Segmentation**: partitioning an image into its main components
    - extracting the license plate from the rest of the original image
    - segmenting characters within the plate area
  - **Image representation**: algorithms to encode image contents employing measures of colour (or intensity) distribution, texture, and shape of the most relevant (previously segmented) objects within the image.
  - **Object recognition**: classify (*i.e.*, assign a label to) each individual character, producing a string at the output, containing the license plate contents

# Machine Vision System






DEPARTAMENTO CIÊNCIA E TECNOLOGIA

# Human Visual System

- **Human visual** system and a **machine vision system** have different strengths and limitations
- It is extremely difficult to emulate the performance of the human visual system
- Eye (input sensor) and the brain (information processing unit) connected by the optic nerve (transmission path)
- Image perception consists of capturing the image with the eye, then recognising it, and finally interpreting its contents in the brain

# Human Visual System

This is a cat!!

Of course!!

How have we discovered that this is a cat?

# Human Visual System

| Easy | Hard |
|---|---|
| How wide is this plate? Is it dirty? | Look at a picture of a random kitchen, and find all the dirty plates. |
| Did something change between these two images? | Track an object or person moving through a crowded room of other people. |
| Measure the diameter of a wheel. Check to see if it is bent. | Identify arbitrary parts on pictures of bicycles. |
| What color is this leaf? | What kind of leaf is this? |

# Digital Image Acquisition

- The **formation of an image** is the result of:
  - the **reflection of light** on an object or scene
  - the **sensors** typically used to capture the reflected energy
  - Selection of appropriate number of samples and quantization levels

- The **reflection of light on an object or scene**
  - **Light** is an essential requirement for an image to be created, captured, and perceived
  - Light can be described in terms of **electromagnetic waves** or particles, called photons
  - The **human visual system** is sensitive to **photons of wavelengths** (violet, blue, green, yellow, orange, red).
  - Colours perceived by humans are determined by the nature of the light



UPT DCT DEPARTAMENTO CIÊNCIA E TECNOLOGIA

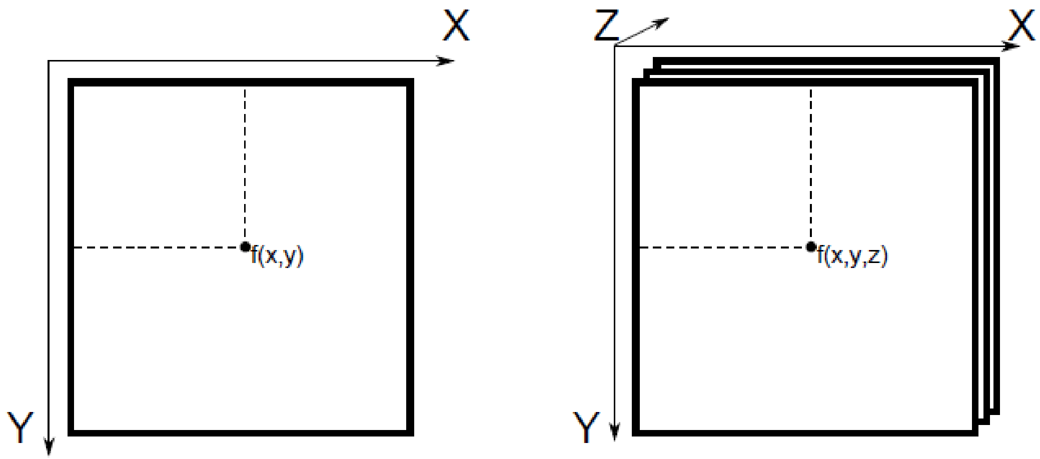
# Digital Image Acquisition

- The simplest way to encode colour in cameras and displays is by using the red (R), green (G), and blue(B) values of each pixel.

Let’s start now with basics of image representation and start coding!

# Image Representation

- Images are represented by a **2/3-D matrix** (gray/coloured image):
  - Y **rows**, X **columns** and Z **channels**;
  - Each **pixel position** is encompassed by **discrete coordinates**
  - The **point (0,0)** corresponds to the upper **left corner** of the image.
  - **Moving down** and to the right, both the **Y and X values increase**.






DEPARTAMENTO CIÊNCIA E TECNOLOGIA

# Image Representation

- **Images** are **set of pixels**
- Thinking of an image **as a grid**, **each square** is single pixel
- Gray images are composed by a 2D matrix

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 2 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 3 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 4 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 5 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 6 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 7 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 8 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 9 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |
| 10 | | 15 | 30 | 45 | 60 | 75 | 90 | 105 | 120 | 135 | 150 | 165 | 180 | 195 | 210 | 225 | 240 | 255 |

# Image Representation

- Example:

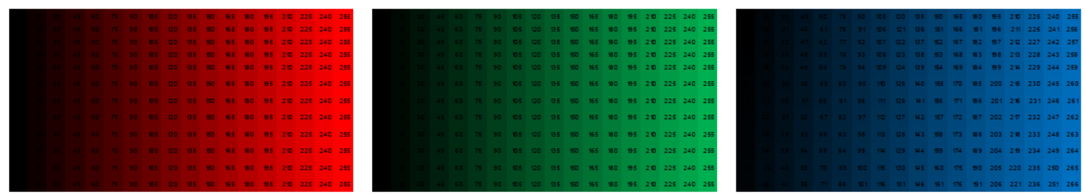
How many pixels has an image with a resolution of 500 X 300?

The image is represented as a grid of pixels with 500 rows and 300 columns.

In total, there are 500 X 300 = 150 000 pixels

# Image Representation

- Colour images are composed by three matrixes one for each colour RGB
- Each pixel is composed of a tuple of 3 int where (0,0,0) is black and (255,255,255) is white
  - Black: (0,0,0)
  - White: (255,255,255)
  - Red: (255,0,0)
  - Green: (0,255,0)
  - Blue: (0,0,255)
  - Yellow: (255,255,0)
  - Purple: (128,0,128)



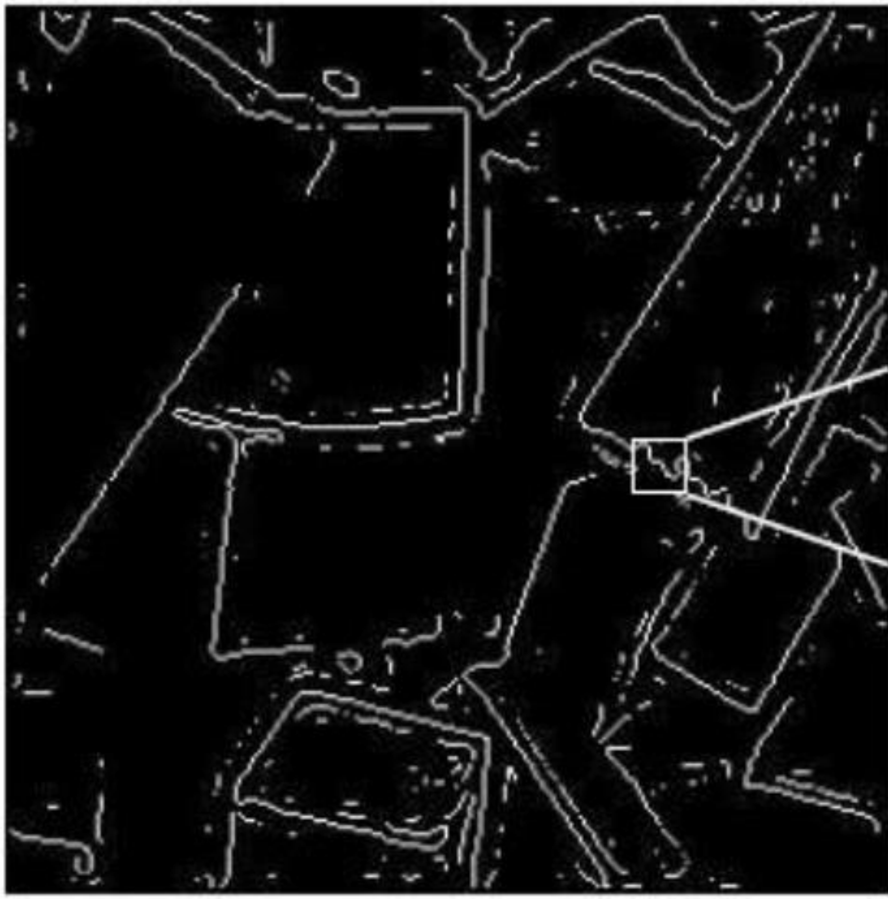
# Image Representation

- The pixel range of a given image format is determined by its bit depth.
- 8-bit image will have a range of [0, 255].
- An image with higher bit depth needs more storage in disk and memory

| Data type | Description | Range |
|---|---|---|
| int8 | 8-bit integer | $-128$ — 127 |
| uint8 | 8-bit unsigned integer | 0 — 255 |
| int16 | 16-bit integer | $-32768$ — 32767 |
| uint16 | 16-bit unsigned integer | 0 — 65535 |
| double | Double precision real number | Machine specific |

- Binary: 1-bit
- Grayscale: 8-bits
- RGB: 24-bits (16.7 million colours since 8 bits represents 256 values, with 3 matrixes we have $256^3$ = 16.7 million values)

# Types of a digital image: binary



| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 |

# Types of a digital image: grayscale

| | | | | | | |
|---|---|---|---|---|---|---|
| 230 | 229 | 232 | 234 | 235 | 232 | 148 |
| 237 | 236 | 236 | 234 | 233 | 234 | 152 |
| 255 | 255 | 255 | 251 | 230 | 236 | 161 |
| 99 | 90 | 67 | 37 | 94 | 247 | 130 |
| 222 | 152 | 255 | 129 | 129 | 246 | 132 |
| 154 | 199 | 255 | 150 | 189 | 241 | 147 |
| 216 | 132 | 162 | 163 | 170 | 239 | 122 |

# Types of a digital image: RGB

| | | | | | |
|---|---|---|---|---|---|
| 49 | 55 | 56 | 57 | 52 | 53 |
| 58 | 60 | 60 | 58 | 55 | 57 |
| 58 | 58 | 54 | 53 | 55 | 56 |
| 83 | 78 | 72 | 69 | 68 | 69 |
| 88 | 91 | 91 | 84 | 83 | 82 |
| 69 | 76 | 83 | 78 | 76 | 75 |
| 61 | 69 | 73 | 78 | 76 | 76 |

Red

| | | | | | |
|---|---|---|---|---|---|
| 64 | 76 | 82 | 79 | 78 | 78 |
| 93 | 93 | 91 | 91 | 86 | 86 |
| 88 | 82 | 88 | 90 | 88 | 89 |
| 125 | 119 | 113 | 108 | 111 | 110 |
| 137 | 136 | 132 | 128 | 126 | 120 |
| 105 | 108 | 114 | 114 | 118 | 113 |
| 96 | 103 | 112 | 108 | 111 | 107 |

Green

| | | | | | |
|---|---|---|---|---|---|
| 66 | 80 | 77 | 80 | 87 | 77 |
| 81 | 93 | 96 | 99 | 86 | 85 |
| 83 | 83 | 91 | 94 | 92 | 88 |
| 135 | 128 | 126 | 112 | 107 | 106 |
| 141 | 129 | 129 | 117 | 115 | 101 |
| 95 | 99 | 109 | 108 | 112 | 109 |
| 84 | 93 | 107 | 101 | 105 | 102 |

Blue

# Images files

- Image file formats use a file header followed by (often compressed) pixel data

- The image file header stores information about the image
  - height and width
  - number of bands
  - number of bits per pixel
  - signature bytes indicating the file type

- More complex file formats, the header may also contain information about:
  - the type of compression used and other parameters that are necessary to decode (i.e., decompress) the image.



DEPARTAMENTO CIÊNCIA E TECNOLOGIA

# Images files

- BIN, PPM (PBM,PGM, PNM) – raw pixel data
- BMP – Microsoft Windows
- JPEG – most popular for photographic quality images
- GIF – uses indexed representation of colours with maximum of 256 colours palette
- TIFF – 24 bit per pixel (true colour) 5 different compression schemes
- PNG – supports both indexed and true colour format, patent free replacement for GIF
- RAW – adopted by camera manufacturers

# Geometric Operations

- **Modify the geometry** of an image by **repositioning pixels** in a constrained way
- **Modify the spatial relationships** between **groups of pixels** representing features or objects of interest within the image
- Are used for different purposes:
  - Correction of geometric distortions introduced during the image acquisition process;
  - Special effects (*e.g.*, twirling, bulging, or squeezing);
- The most **common techniques** include **translation**, **rotation**, **resizing**, **flipping**, and **cropping**

# Geometric Operations

translation

scaling

rotation

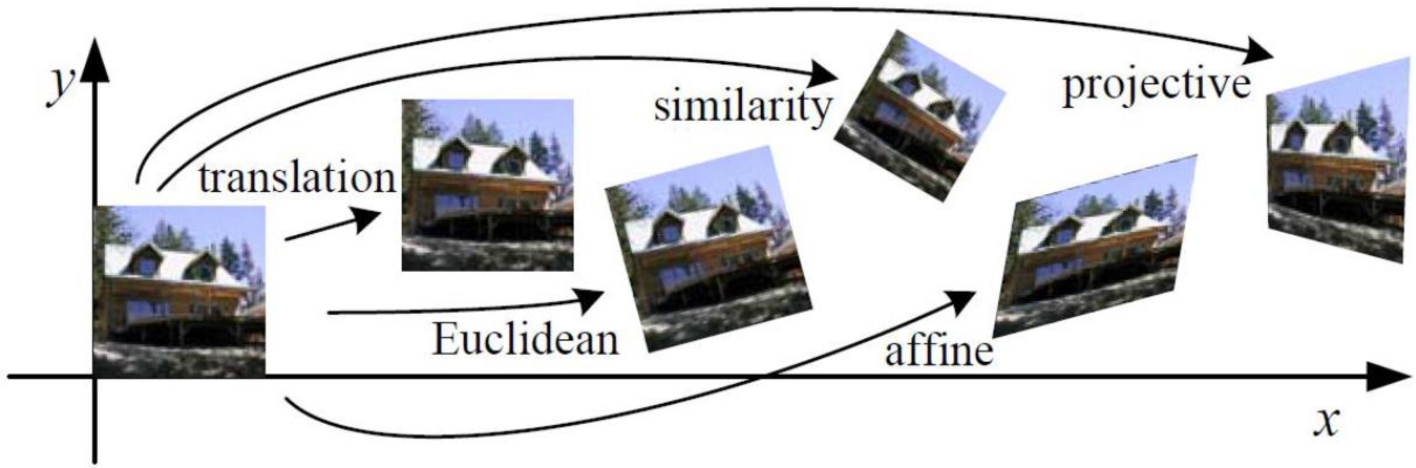
mirror

affine

projective

# Geometric Operations

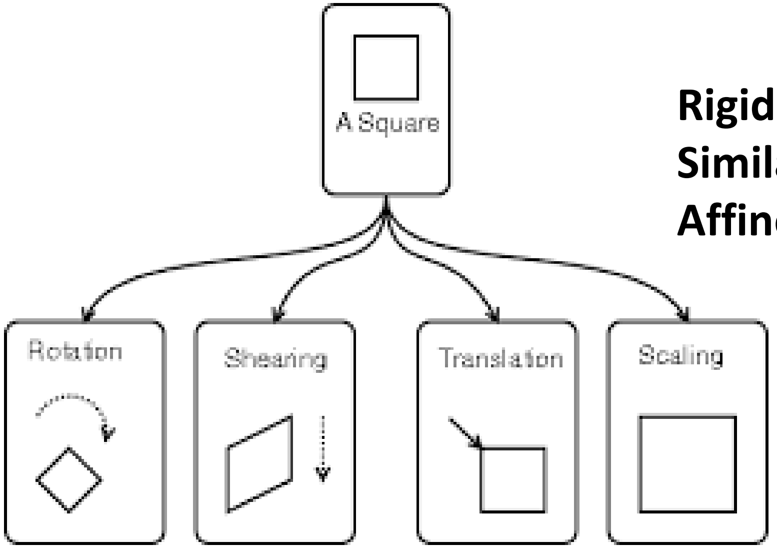
- Composition of transformations:
  - Rigid transformation: Translation + Rotation (distance preserving).
  - Similarity transformation: Translation + Rotation + uniform Scale
  - Affine transformation: Translation + Rotation + Scale + Shear



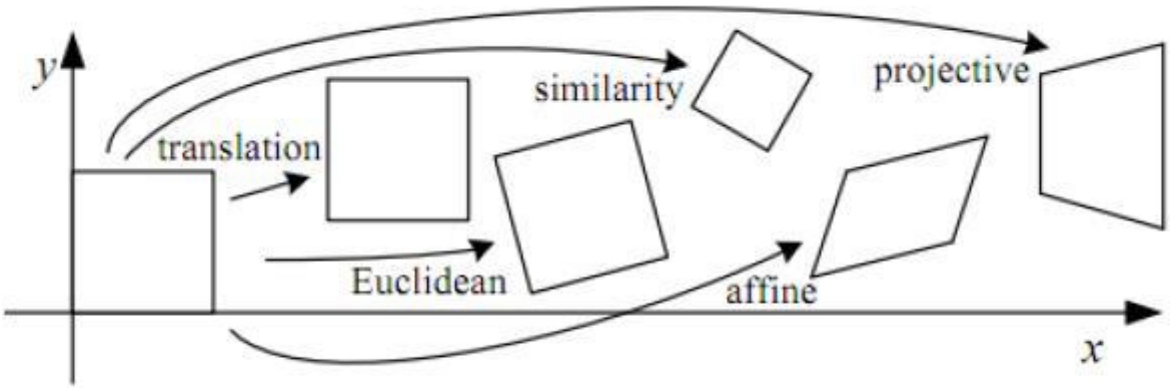



UPT DCT DEPARTAMENTO CIÊNCIA E TECNOLOGIA

# Geometric Operations




**Rigid transformation**: Translation + Rotation (distance preserving)
**Similarity transformation**: Translation + Rotation + uniform scaling
**Affine transformation**: Translation + Rotation + Scaling + Shearing



# Geometric Operations

- A geometric transformation require:
  - a **mapping function** that can be described in terms of matrix operations

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{Translation by } (x_0, y_0)$$

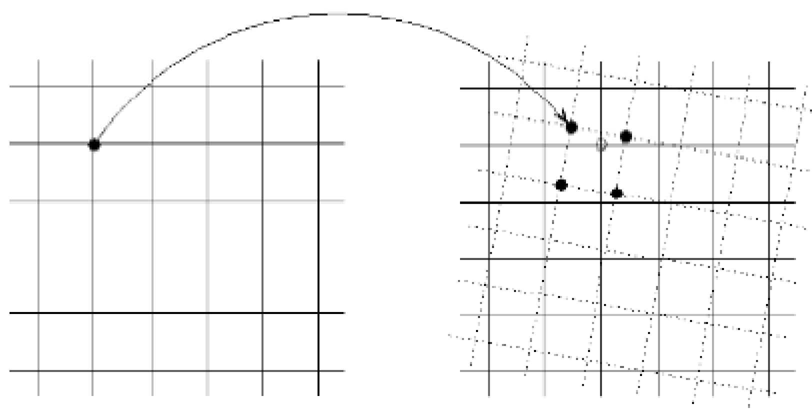
$$\mathbf{T} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{Scale by } s_1 \text{ and } s_2$$

$$\mathbf{T} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{Rotate by } \theta$$

# Geometric Operations

- **Interpolation methods** to find the value of the image at the grid points in the target coordinate system.

- The previous mapping locates the grid points of A in the coordinate system of B, but those grid points are not on the grid of B





DEPARTAMENTO **CIÊNCIA E TECNOLOGIA**

# Geometric Operations

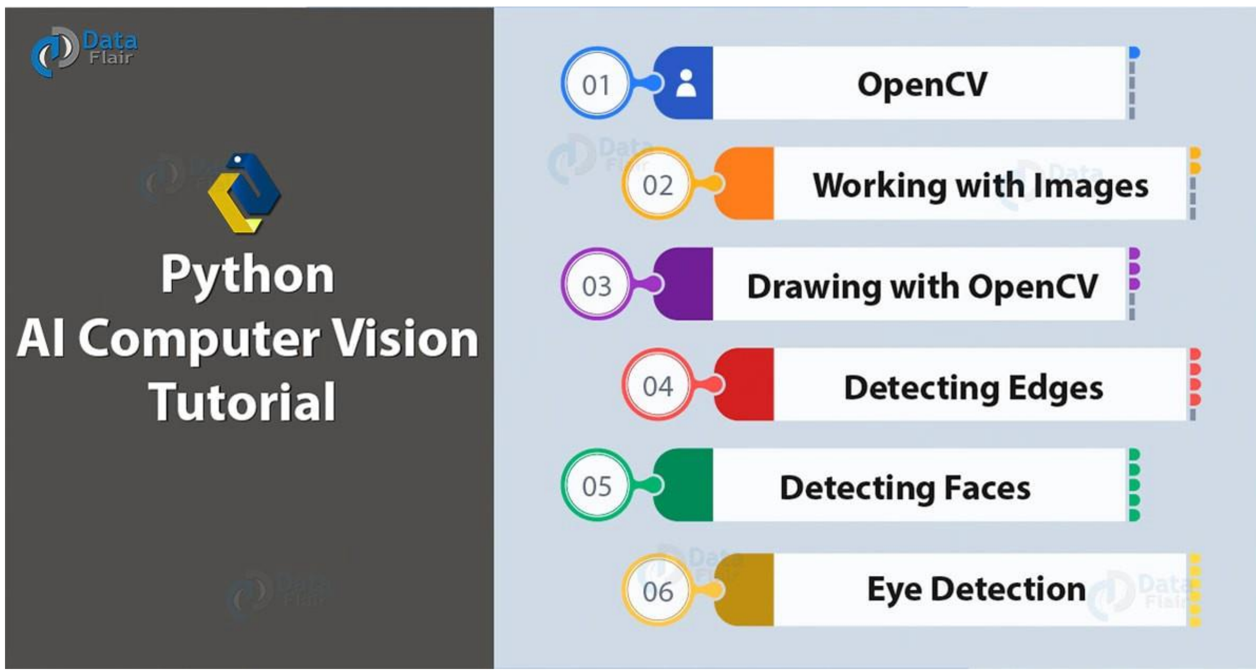
- **Translation** - shifting of an image along the x and y axis (shift up, down, left, or right, along with any combination of the above)
- **Bird’s-eye view** - an elevated view of an object from above, with a perspective as though the observer were a bird
- **Rotation** - rotating an image by some angle
- **Resizing** - changing the size of an image
- **Flipping** - moving an image by a mirror-reversal of an axis.
- **Cropping** - remove/select parts of an image.



DEPARTAMENTO CIÊNCIA E TECNOLOGIA

# Python: OpenCV Module

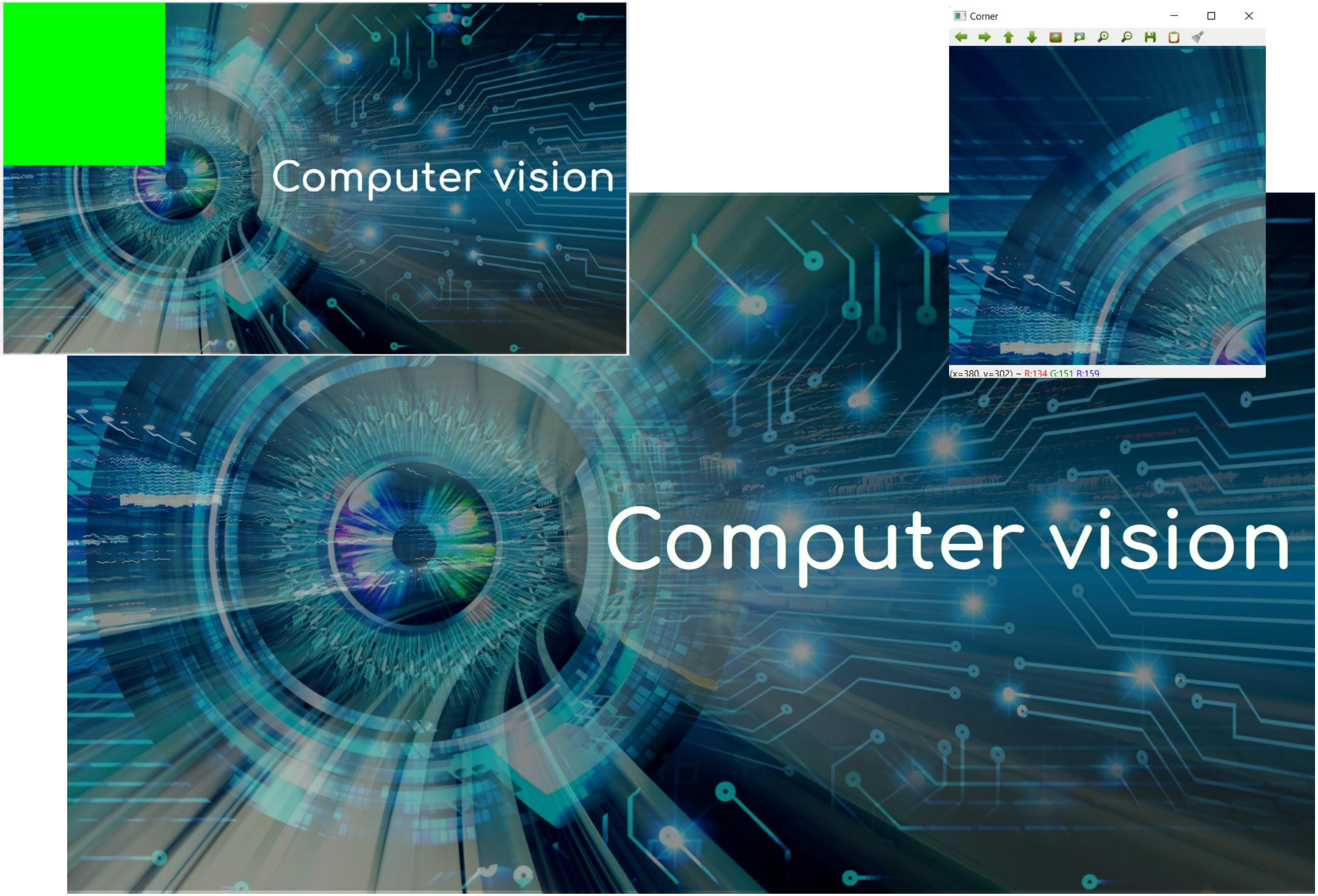
- OpenCV Python is a library designed to solve computer vision problems
- During the semester we will learn to use it.



# Example

- Download CV.jpg from Moodle
- Do:
  1. Image read and show in Python
  2. Access the pixel 500,500
  3. Change the colour of pixel 500,500 to green
  4. Select the pixels from row 0 till 500 and column 0 to 500
  5. Do image show of this part of the image
  6. Put to green the pixels from row 0 till 100 and column 0 to 100

```
import cv2
#1
image = cv2.imread("CV.jpg")
cv2.imshow("Welcome to Computer Vision!", image)
cv2.waitKey(0)
#2
(b,g,r) = image[500,500]
print("Pixel at 0,0 - red {}, green {}, blue {}".format(r,g,b))
#3
image[0,0]=(0,255,0)
#4
corner=image[0:500,0:500]
#5
cv2.imshow(“Corner!", corner)
cv2.waitKey(0)
#6
image[0:500,0:500]=(0,255,0)
cv2.imshow("Image Changed!", image)
cv2.waitKey(0)
```


Computer vision
Corner
(x=380, y=302) ~ R:134 G:151 B:159
Computer vision

# Let’s play with images!

UNIVERSIDADE PORTUCALENSE

Do conhecimento à prática.