



Bài giảng môn học:
Thị giác máy tính (7080518)

CHƯƠNG 1:

TỔNG QUAN VỀ THỊ GIÁC MÁY TÍNH

(Phần 1)

Đặng Văn Nam
dangvannam@hmg.edu.vn

Nội dung chương 1 – Phần 1

1. Tổng quan về Thị giác máy tính
2. Tâm quan trọng của Thị giác máy tính
3. Các ứng dụng của thị giác máy tính
4. Giới thiệu ngôn ngữ Python và Thư viện OpenCV
5. Làm quen với công cụ môi trường học tập
6. Đọc và hiển thị ảnh với OpenCV



Chương 1: Tổng quan về thị giác máy tính – Phần 1

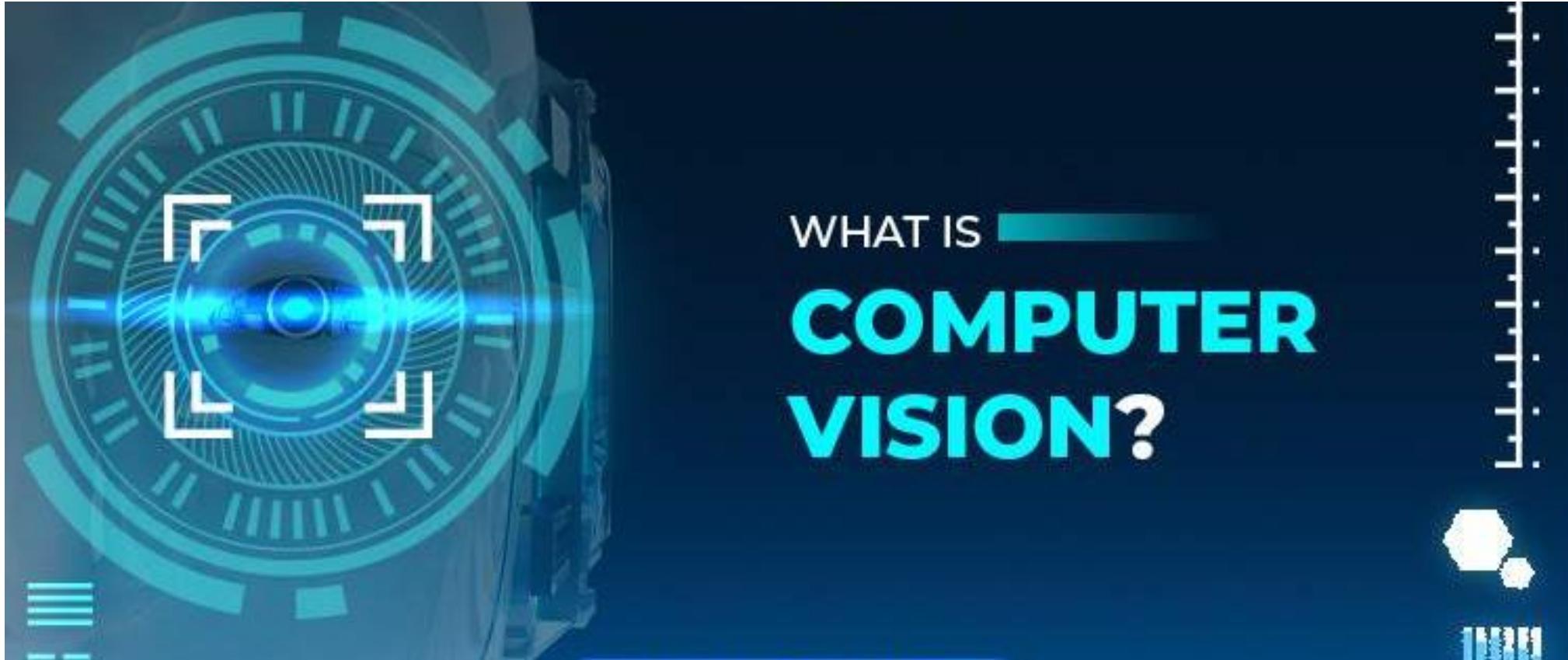


- **Học xong module này người học có thể:**
 - Trả lời được câu hỏi: Thị giác máy tính là gì?
 - Phân biệt được xử lý ảnh với thị giác máy tính
 - Tầm quan trọng của thị giác máy tính hiện nay
 - Các ứng dụng hàng ngày của thị giác máy tính
 - Sơ lược về ngôn ngữ Python và thư viện OpenCV
 - Chuẩn bị công cụ và môi trường thực hành.
 - Đọc và hiển thị ảnh khác nhau với OpenCV

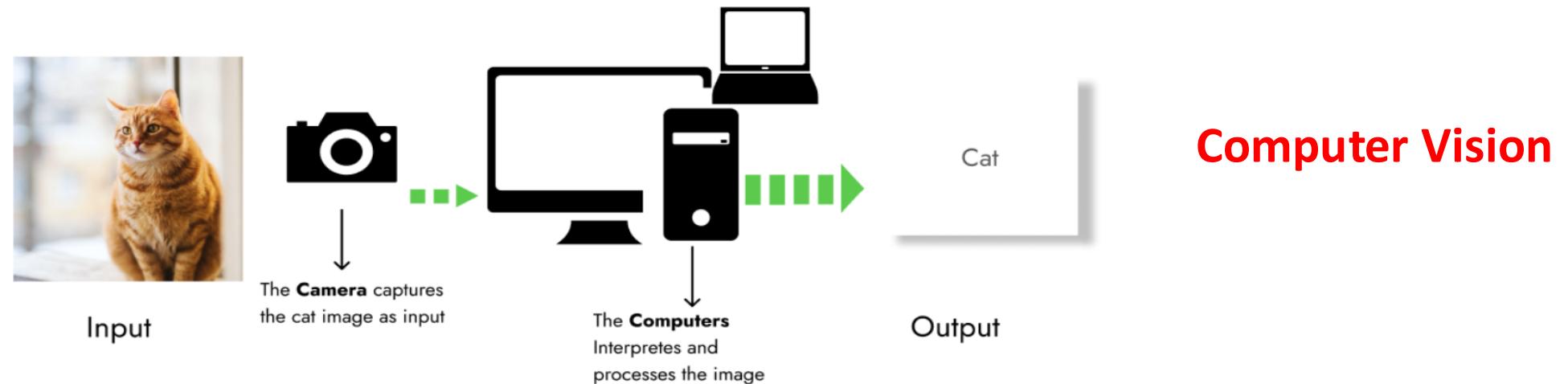
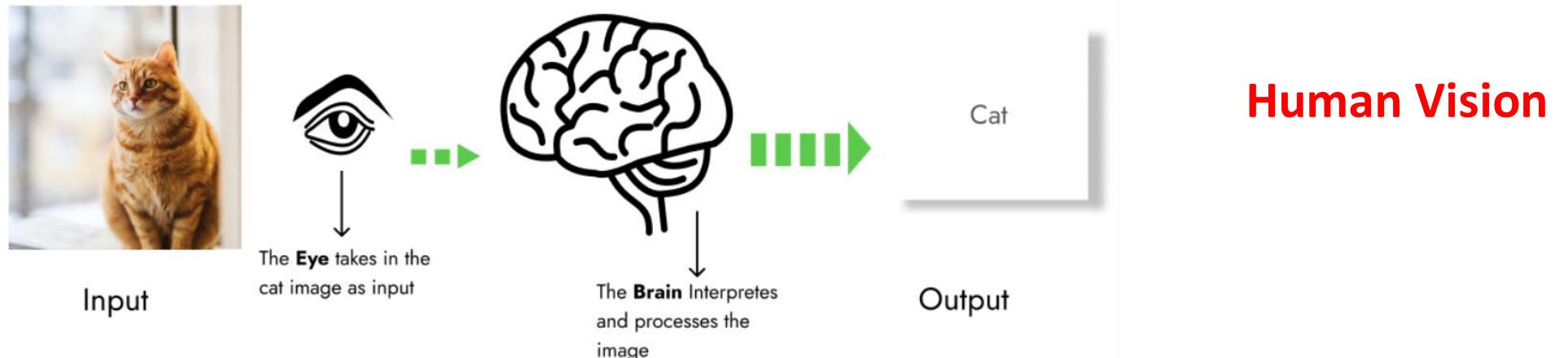


1. Tổng quan về thị giác máy tính

Thị giác máy tính là gì?



Thị giác máy tính là gì?



Thị giác máy tính là gì?

- Thị giác máy tính là lĩnh vực nghiên cứu tập trung vào vấn đề “helping computers to see”. *Computers are able to recognize and process the objects from the video/images as a human can able to recognize.*



Thị giác máy tính đang là một trong những lĩnh vực **hot** nhất của khoa học máy tính và nghiên cứu trí tuệ nhân tạo.

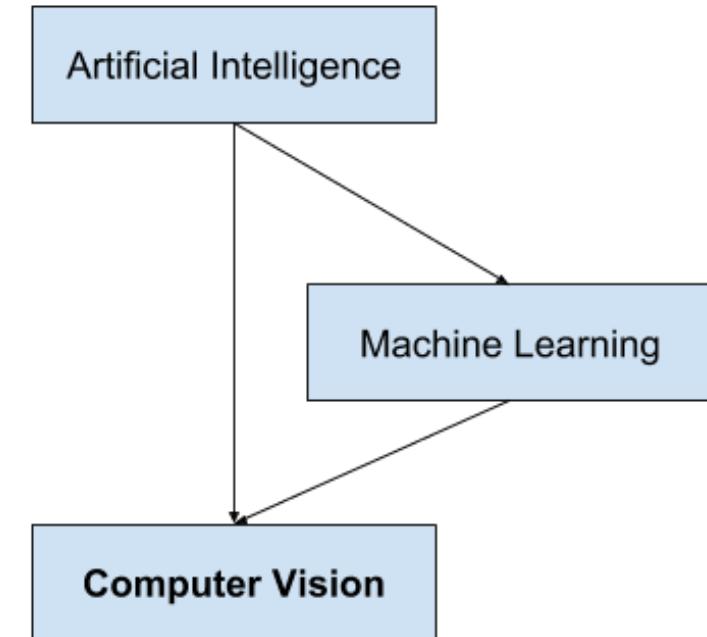
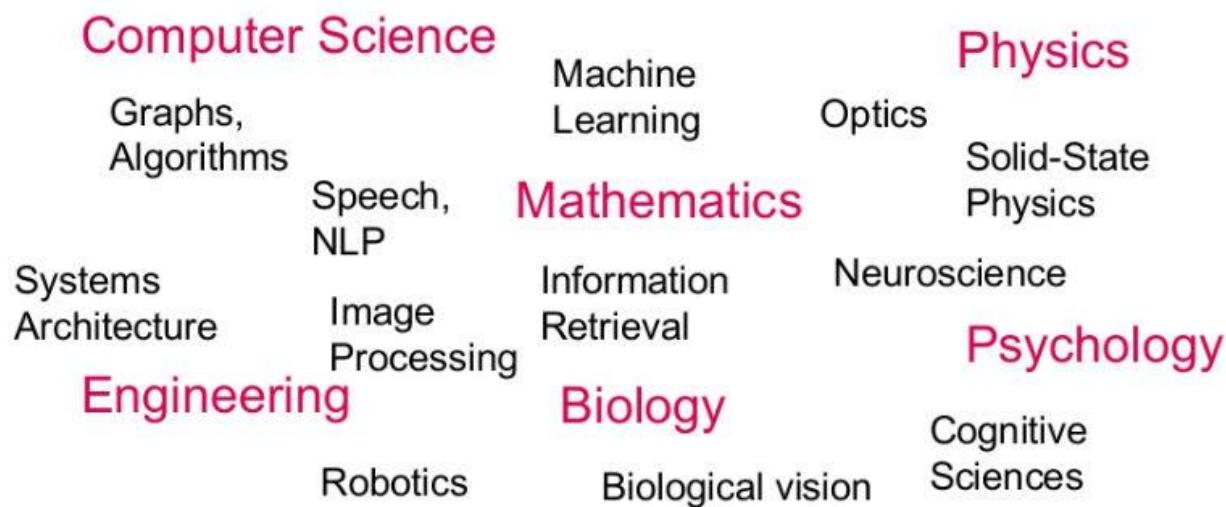
Về mặt tổng thể chúng vẫn chưa thể cạnh tranh với sức mạnh thị giác của mắt người, đã có rất nhiều ứng dụng hữu ích được tạo ra để khai thác tiềm năng của chúng.

Trong một số lĩnh vực hẹp, nhiều hệ thống học sâu (DL) dựa trên thị giác máy tính đã có độ chính xác bằng hoặc vượt con người.

Thị giác máy tính là gì?

- Đây là một **lĩnh vực đa ngành - đa lĩnh vực**, sử dụng các phương pháp chuyên biệt và các thuật toán phức tạp trong xử lý.

What is computer vision
Interdisciplinary field

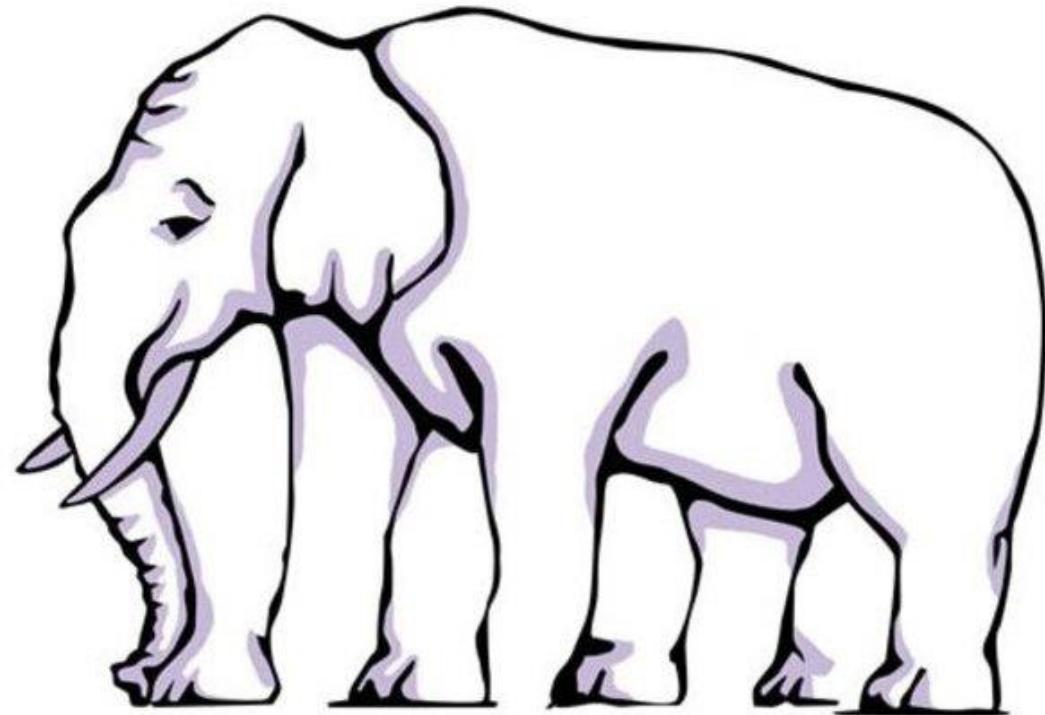


Thị giác máy tính là một lĩnh vực khó, ngay cả với con người!

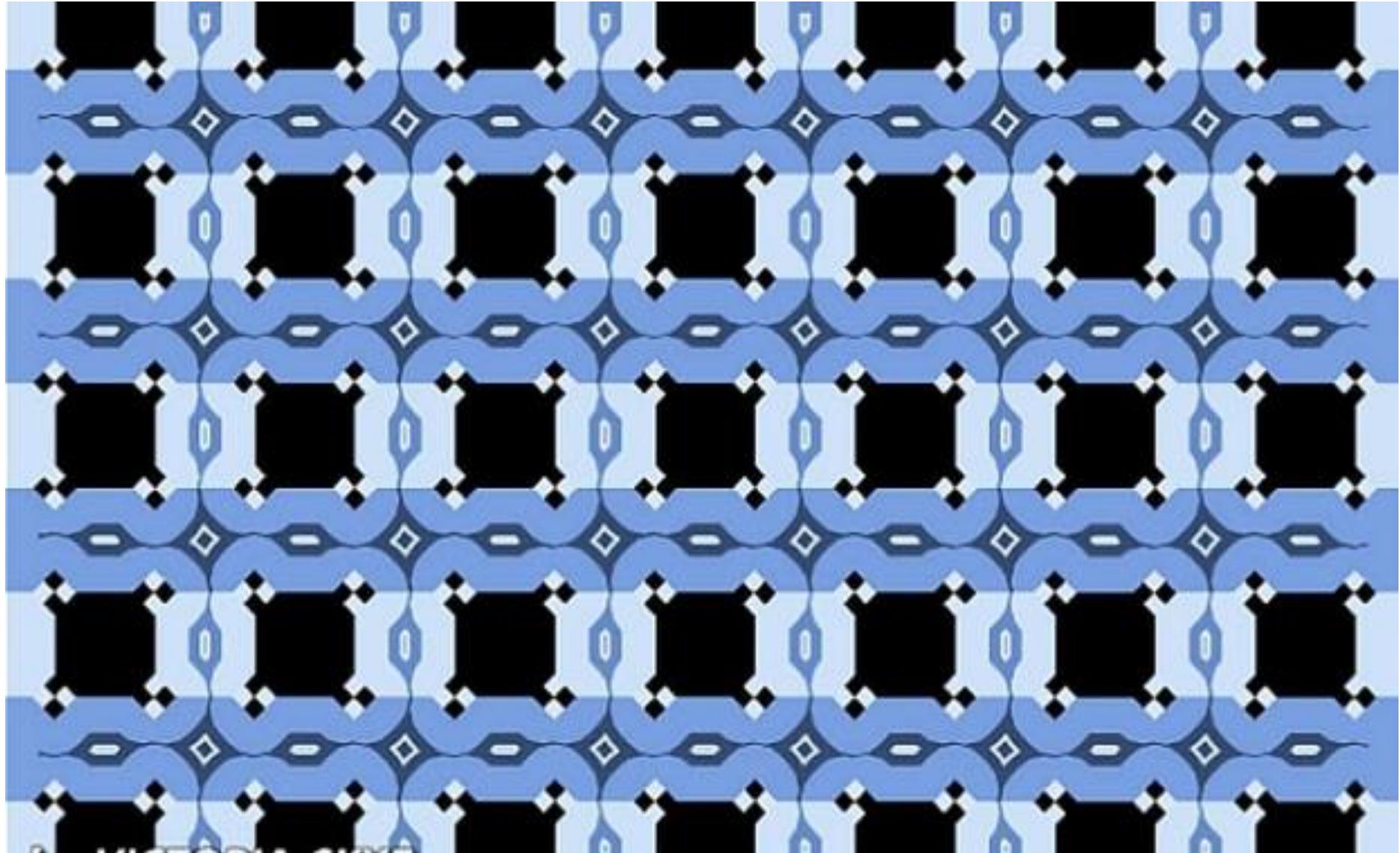
➤ Thách thức thị giác!



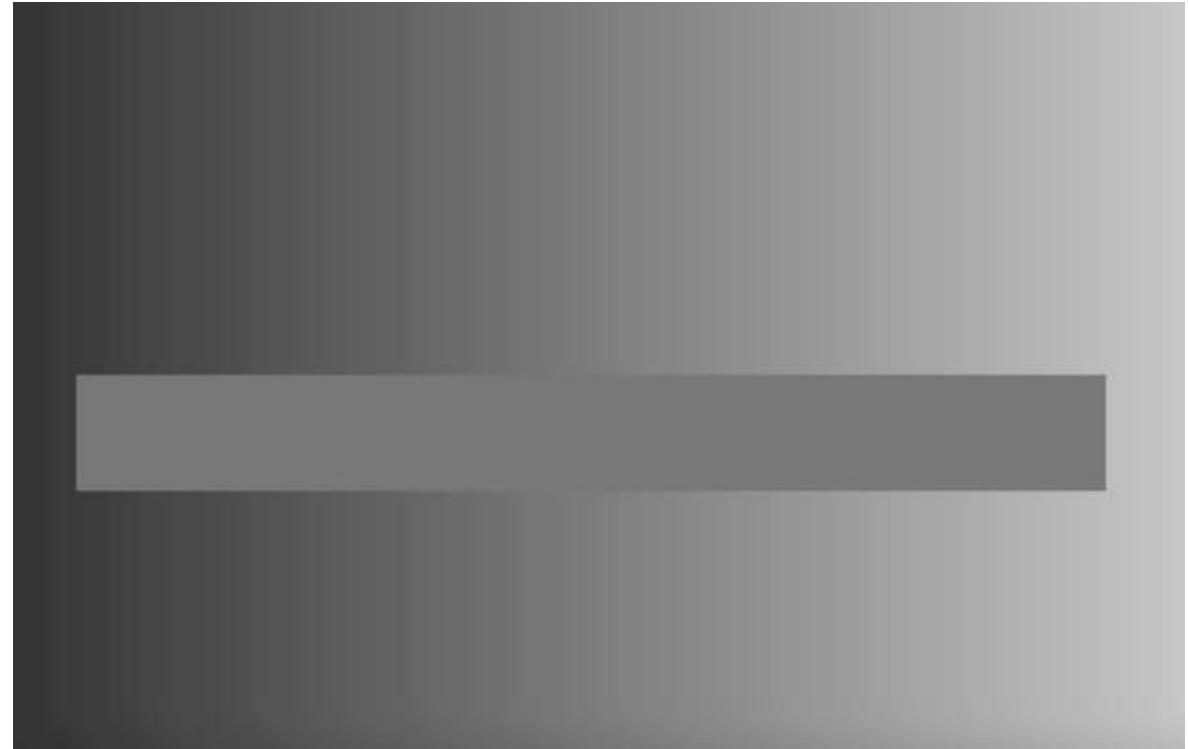
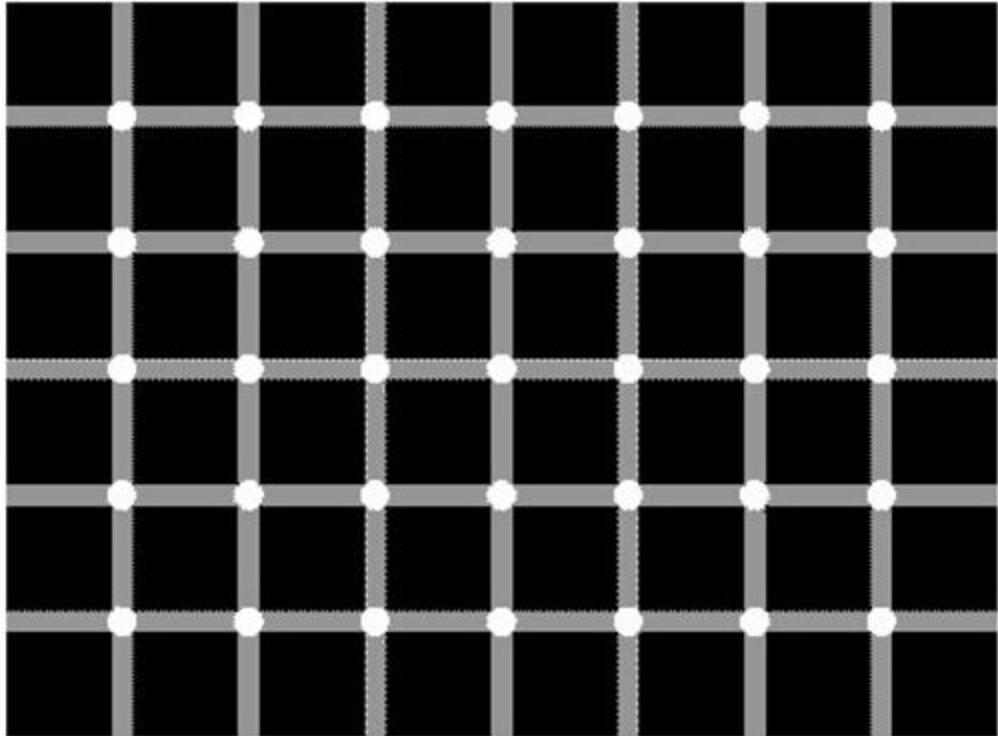
➤ Thách thức thị giác!



➤ Thách thức thị giác!



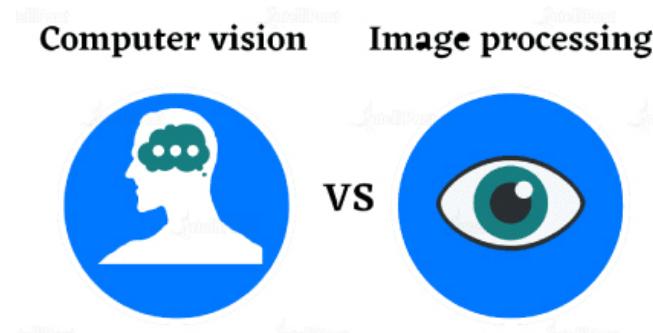
➤ Thách thức thị giác!



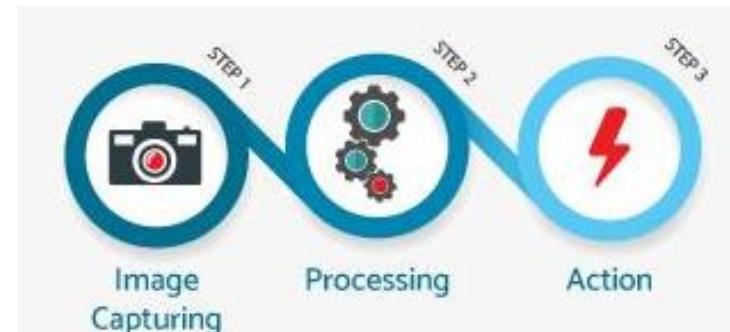
➤ Thách thức thị giác!



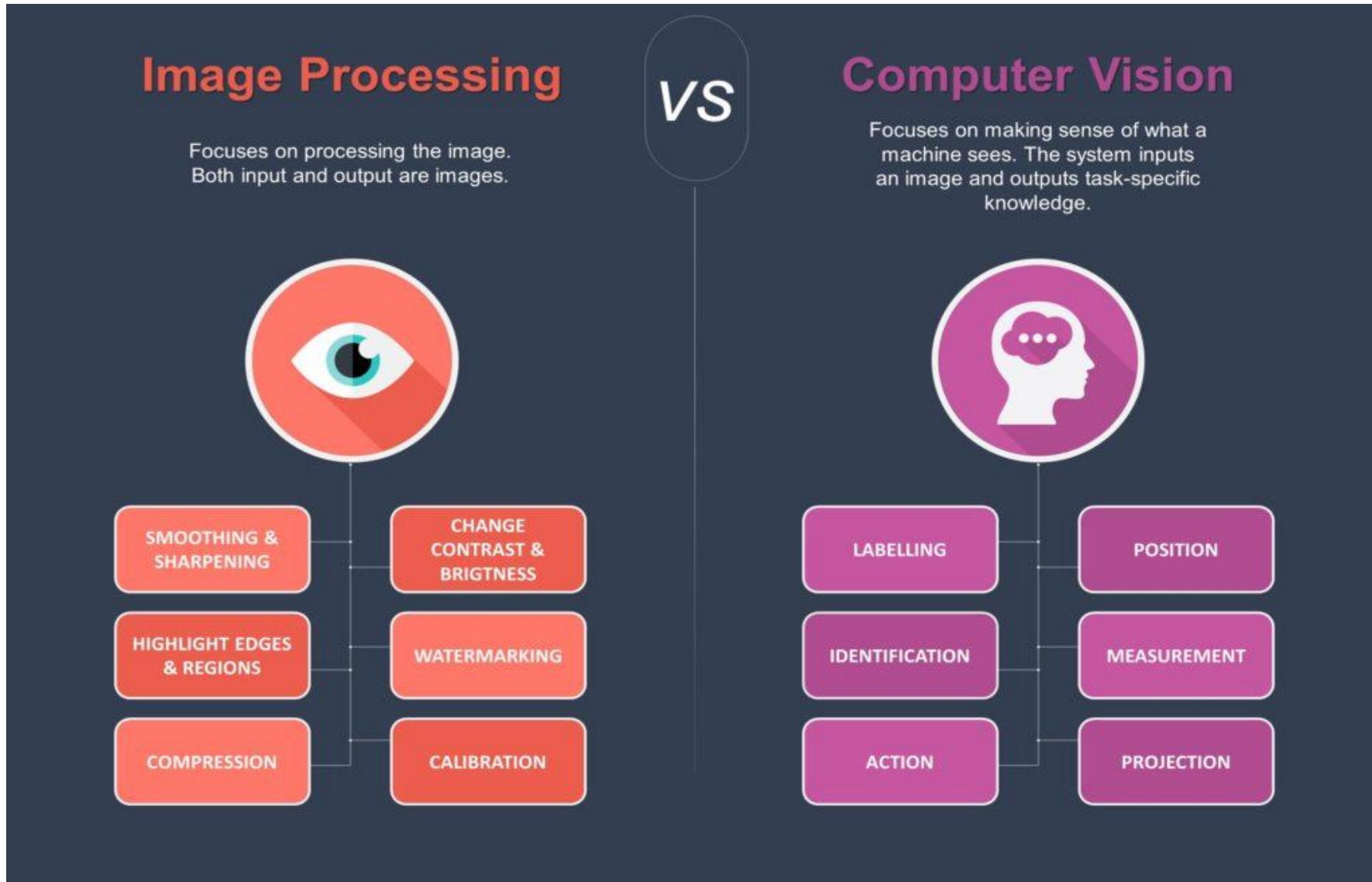
Phân biệt giữa Xử lý ảnh và Thị giác máy tính

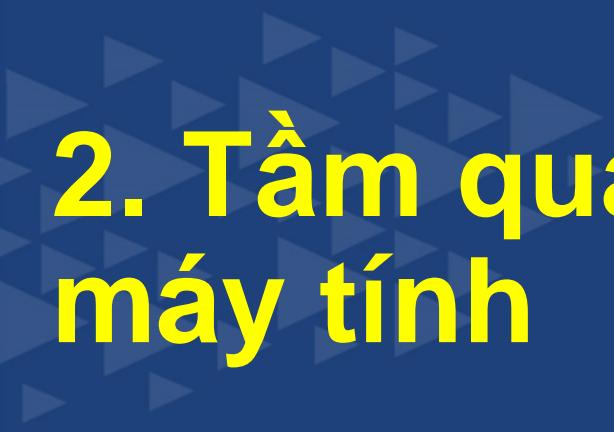


- **Xử lý ảnh (Image processing):** là quá trình tạo ra ảnh mới từ ảnh có sẵn, VD: tăng độ sáng, cắt ảnh, khử nhiễu,... không tập trung vào việc hiểu nội dung của bức ảnh.
- **Thị giác máy tính (Computer vision):** tập trung vào việc hiểu những gì máy móc nhìn thấy, sử dụng xử lý ảnh để xử lý dữ liệu thông tin (tiền xử lý dữ liệu ảnh).



Phân biệt giữa Xử lý ảnh và Thị giác máy tính





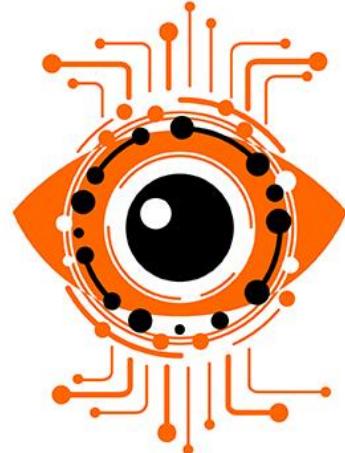
2. Tầm quan trọng của thị giác máy tính

Tâm quan trọng của Thị giác máy tính

44%

of enterprises are actively exploring computer vision.

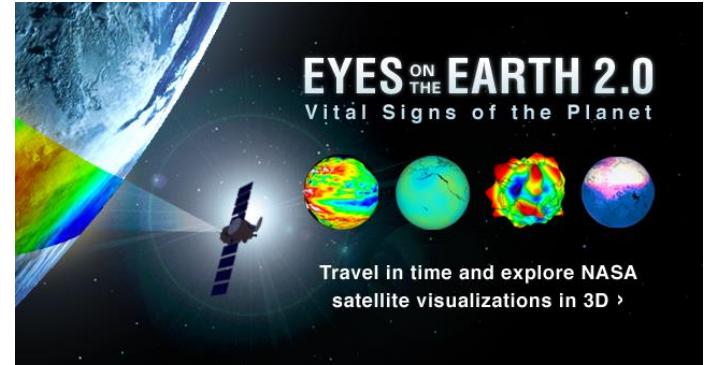
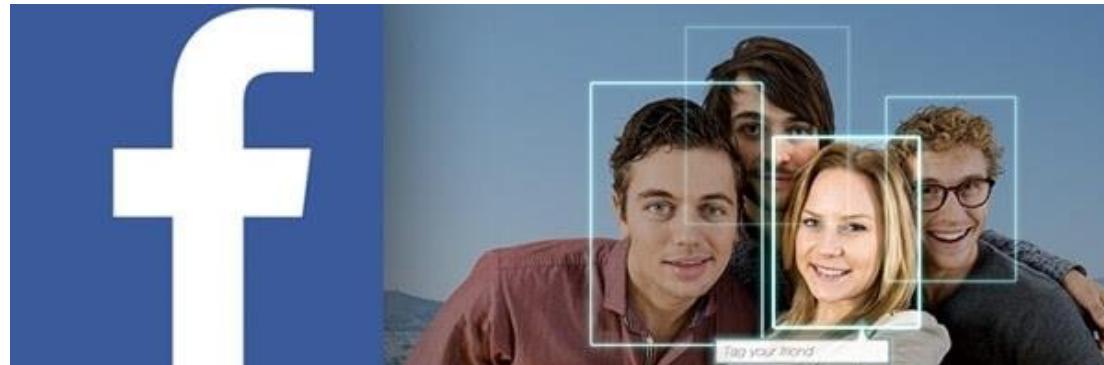
(Source: IDG)



Computer vision market



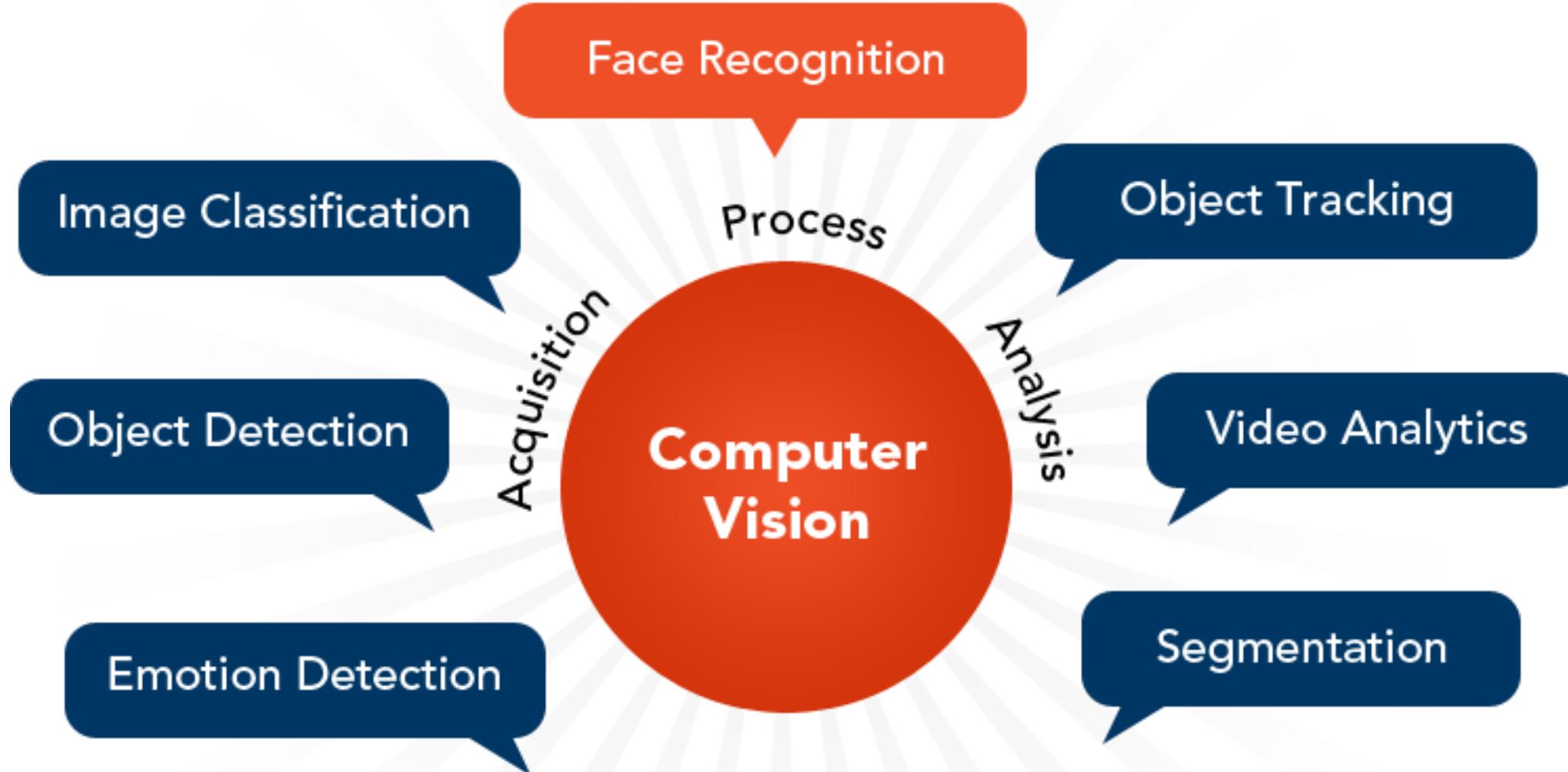
- Bloomberg estimates machine vision market size to be \$18.24 billion by 2025.



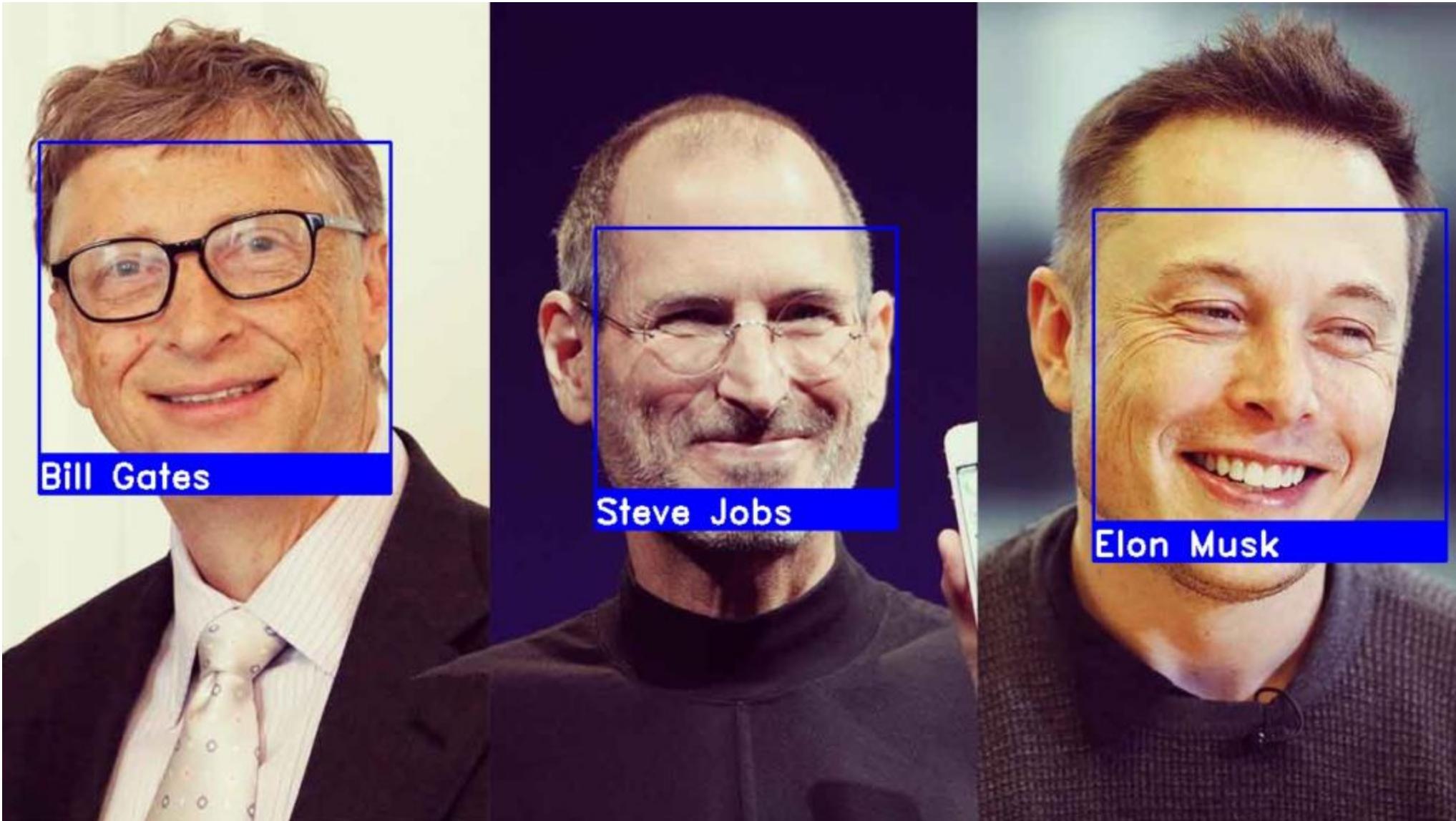


3. Ứng dụng của thị giác máy tính

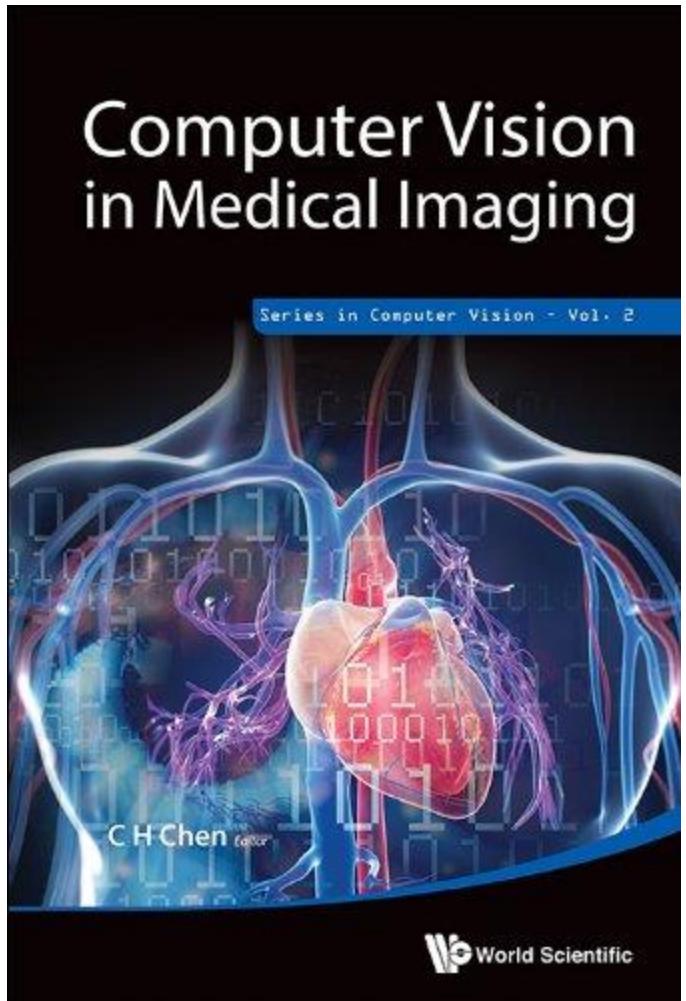
Ứng dụng của thị giác máy tính



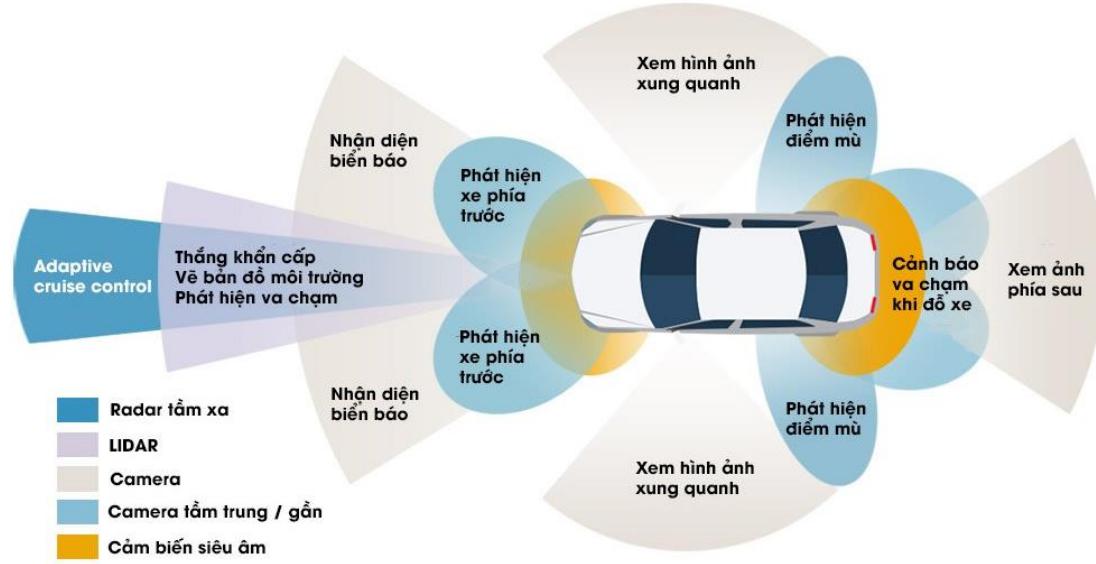
Phát hiện và nhận dạng khuôn mặt



Thị giác máy tính trong y học



Hệ thống xe tự hành, lái xe an toàn, robotics



<https://www.youtube.com/watch?v=51GasqVz1l0>

Sinh trắc học

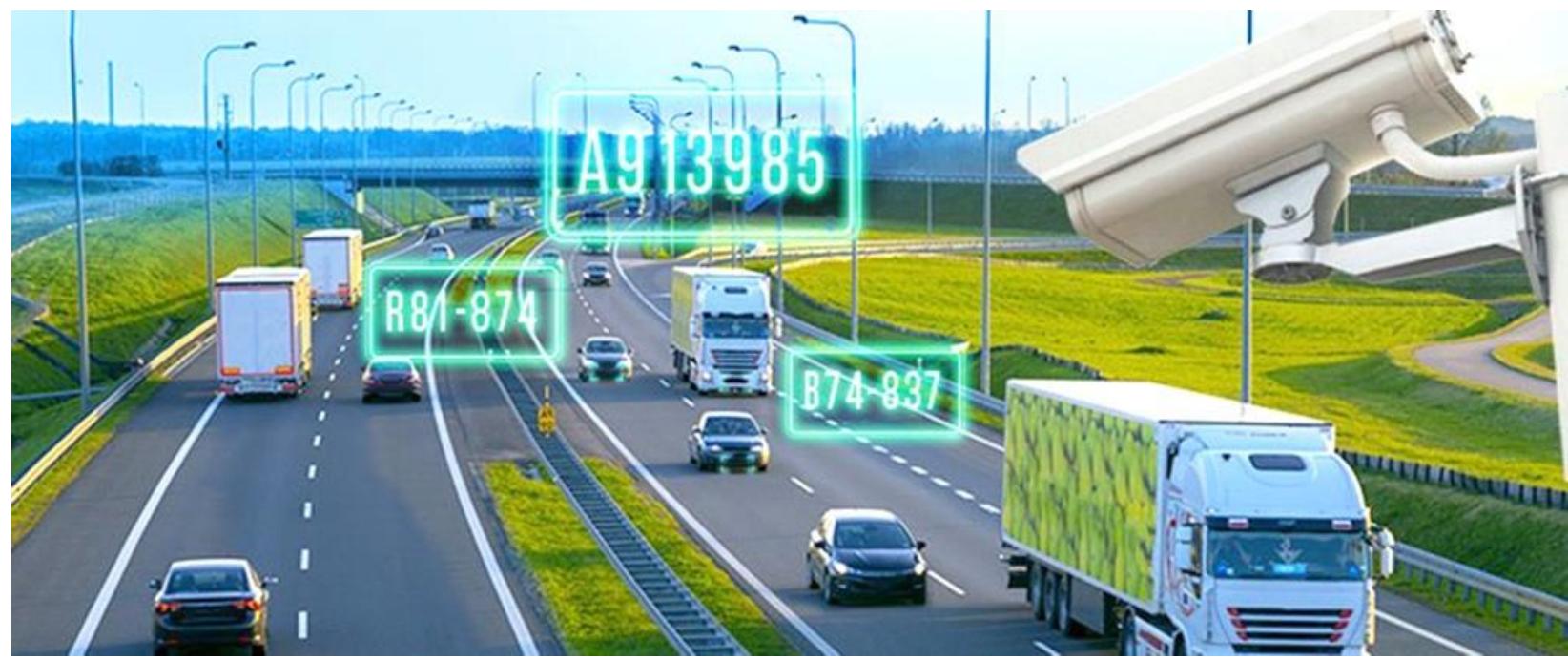
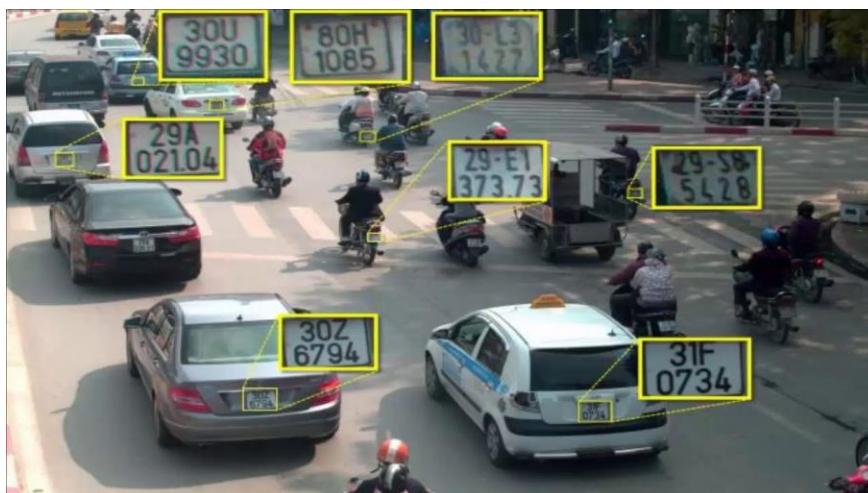
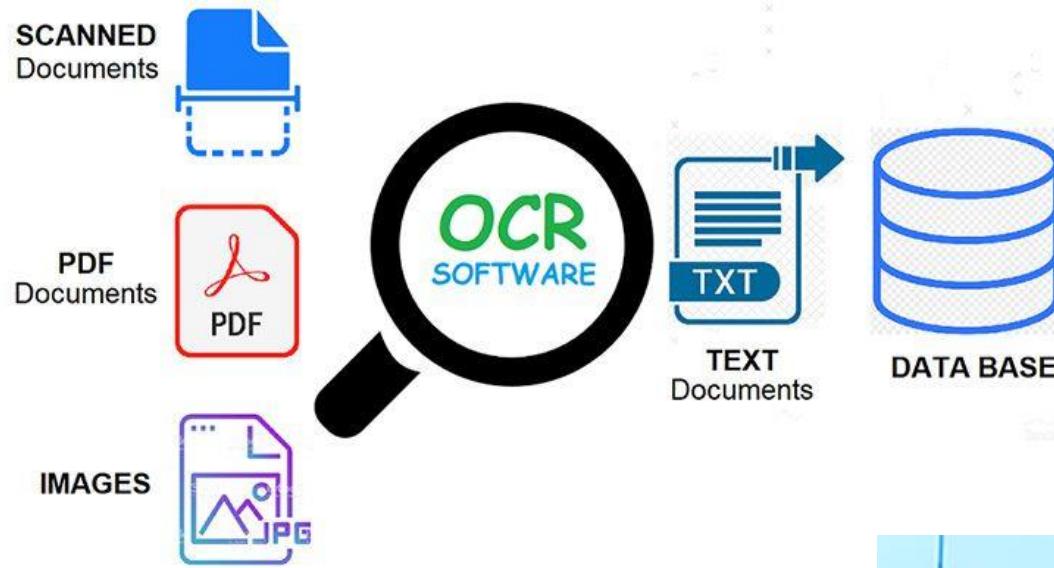


Fingerprint scanners on
many new laptops,
other devices



Face recognition systems now beginning
to appear more widely
iphone X just introduced face recognition

Nhận dạng các ký tự quang học



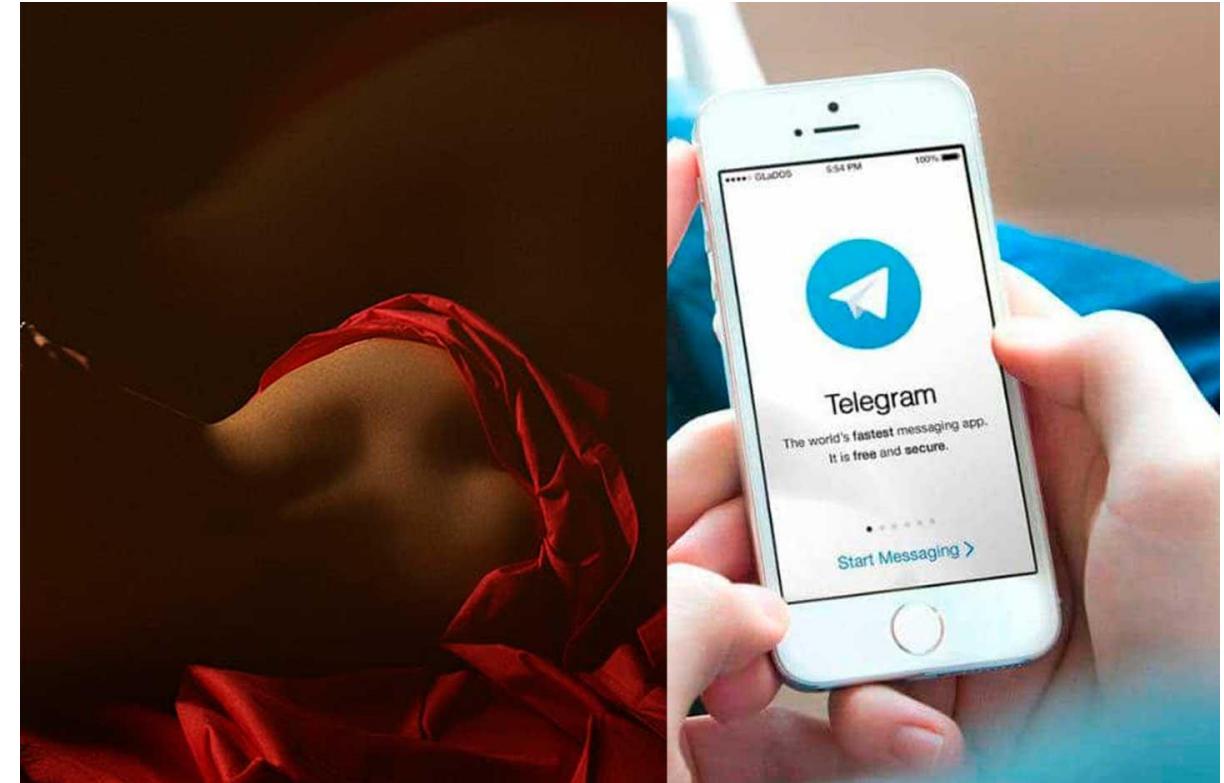
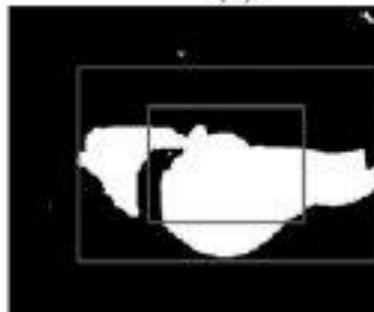
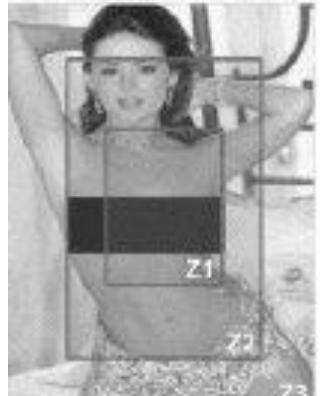
Hiệu ứng đặc biệt: copy hình dạng và chuyển động



Mô hình hóa 3D



Phát hiện ảnh nhạy cảm



Tương tác với người



Microsoft's Kinect



Sony EyeToy



Assistive technologies

Source: S. Seitz

Thực tế ảo tăng cường (AR)

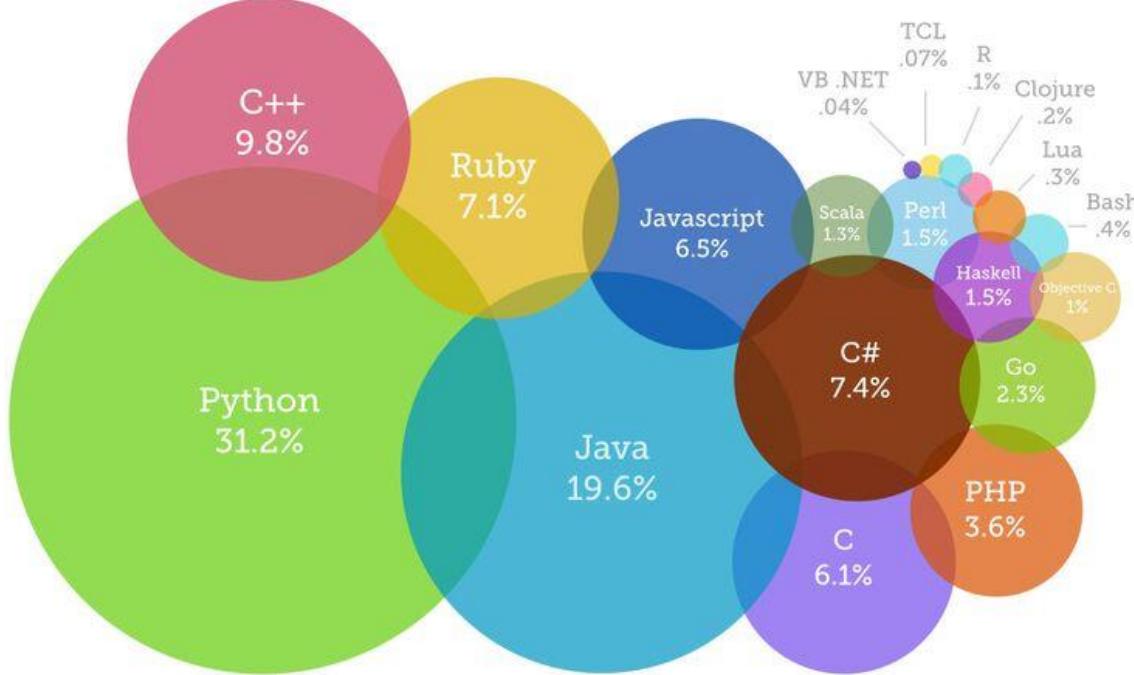


4. Ngôn ngữ Python và Thư viện OpenCV

Ngôn ngữ lập trình Python



Rank	Language	Type	Score
1	Python	🌐💻⚙️	100.0
2	Java	🌐💻⚙️	95.3
3	C	💻⚙️	94.6
4	C++	💻⚙️	87.0
5	JavaScript	🌐	79.5
6	R	💻	78.6
7	Arduino	⚙️	73.2
8	Go	🌐💻	73.1
9	Swift	💻⚙️	70.5
10	Matlab	💻	68.4



Python là một ngôn ngữ lập trình để xây dựng các phần mềm, website... Là một ngôn ngữ đa dụng để tạo ra nhiều chương trình, sản phẩm khác nhau mà không bị giới hạn trong một phạm vi hay vấn đề cụ thể nào.

Thư viện OpenCV

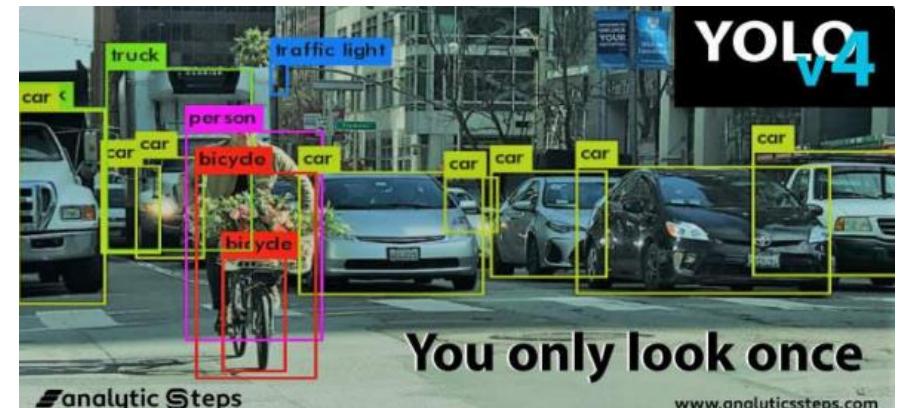
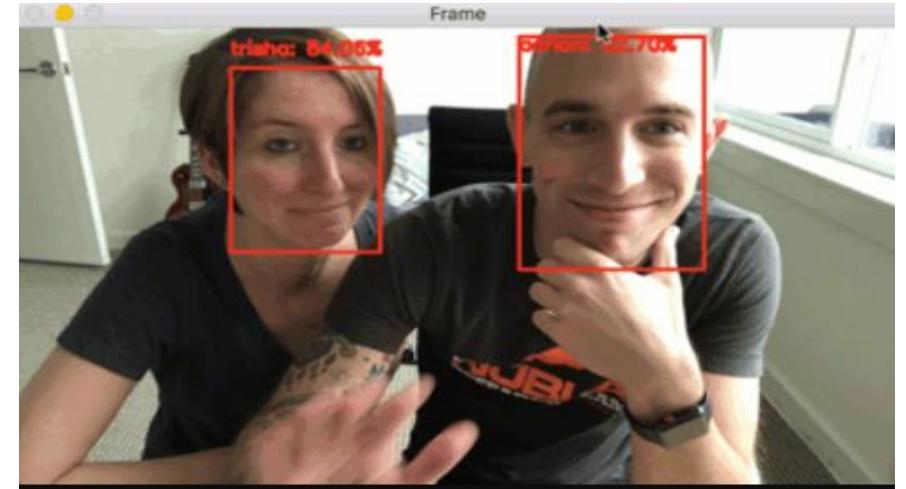
- **OpenCV (Open Source Computer Vision)** là một trong những thư viện mã nguồn mở hàng đầu cho thị giác máy tính, machine learning, xử lý ảnh.
- Chính thức được ra mắt đầu tiên vào năm 1999, OpenCV là thư viện mã nguồn mở miễn phí cho cả học thuật và thương mại.
- Hỗ trợ đa nền tảng gồm Windows, Linux, Mac OS, iOS và Android.
- Được viết bằng C/C++ nên tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng thời gian thực.
- Hỗ trợ ngôn ngữ lập trình C/ C++, Python và Java.



Thư viện OpenCV

Thư viện OpenCV bao gồm một số tính năng nổi bật như:

- Bộ công cụ hỗ trợ 2D và 3D
- Nhận diện khuôn mặt
- Nhận diện cử chỉ
- Nhận dạng chuyển động, đối tượng, hành vi,
- Tương tác giữa con người và máy tính
- Điều khiển Robot
- Hỗ trợ thực tế tăng cường





5. Cài đặt và làm quen với Môi trường học tập

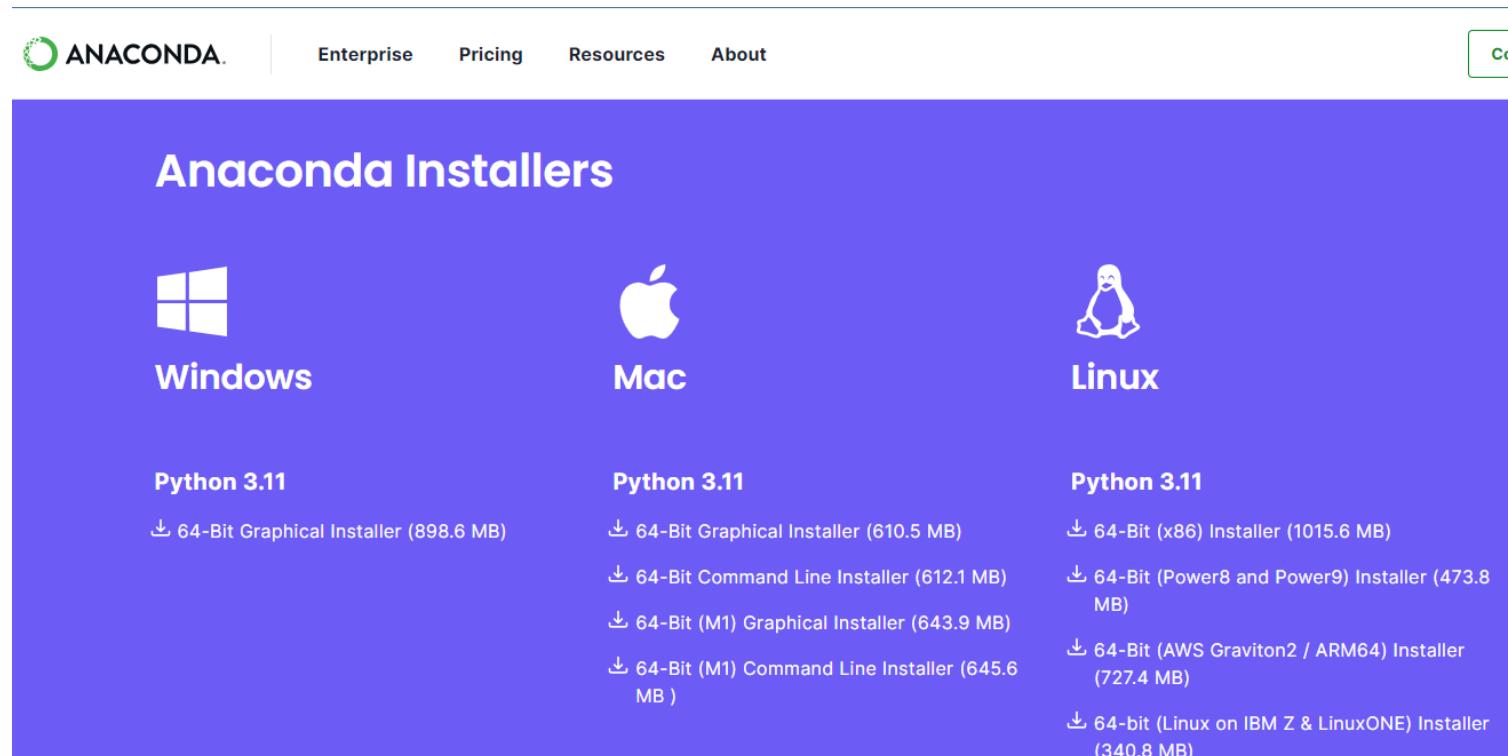
Chuẩn bị môi trường thực hành



Chuẩn bị môi trường thực hành

B1: Tải bộ cài Anaconda

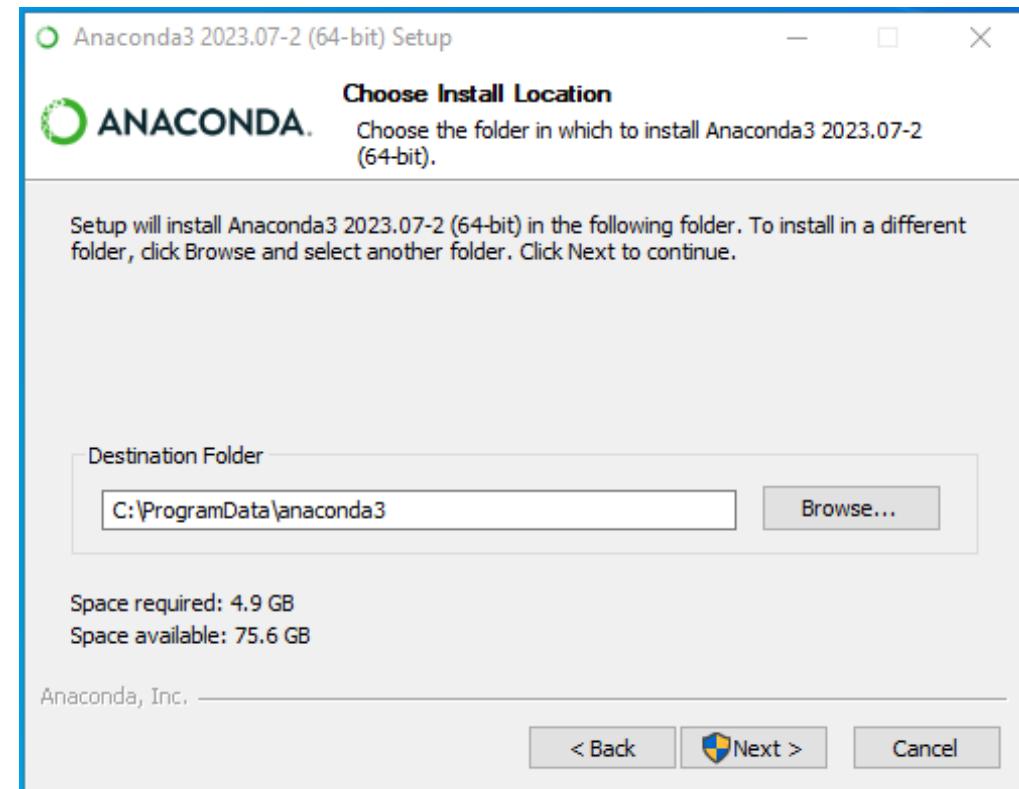
- Truy cập link: <https://www.anaconda.com/download>
- Lựa chọn phiên bản phù hợp với máy:



Chuẩn bị môi trường thực hành

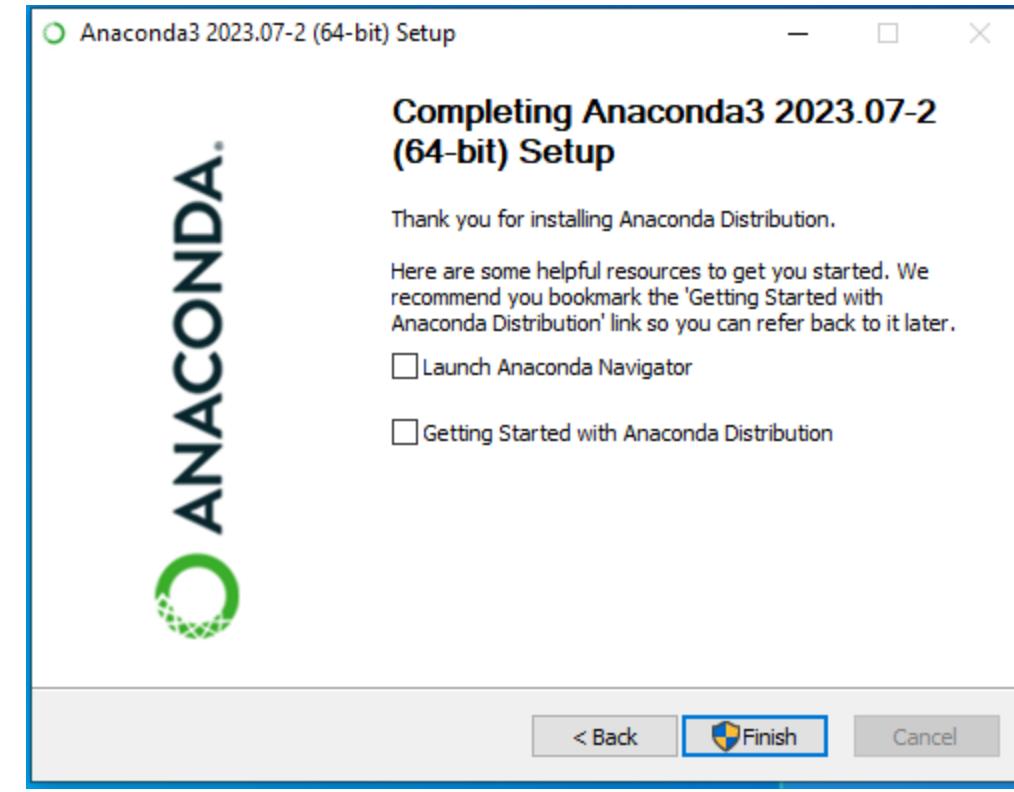
B2: Cài đặt Anaconda

- Chạy bộ cài đã tải về máy (Kích chuột phải vào bộ cài, chọn Run as Administrator):



Chuẩn bị môi trường thực hành

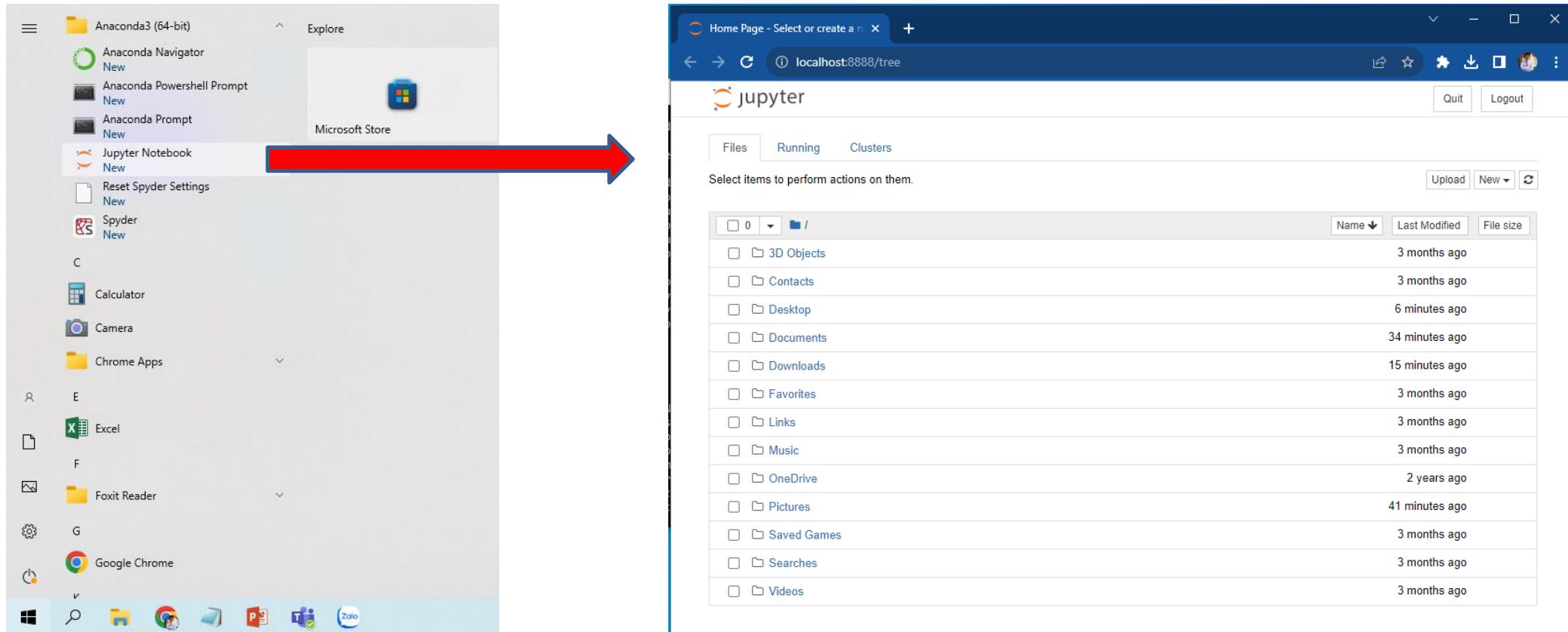
- Next → và hoàn tất cài đặt!



Chuẩn bị môi trường thực hành

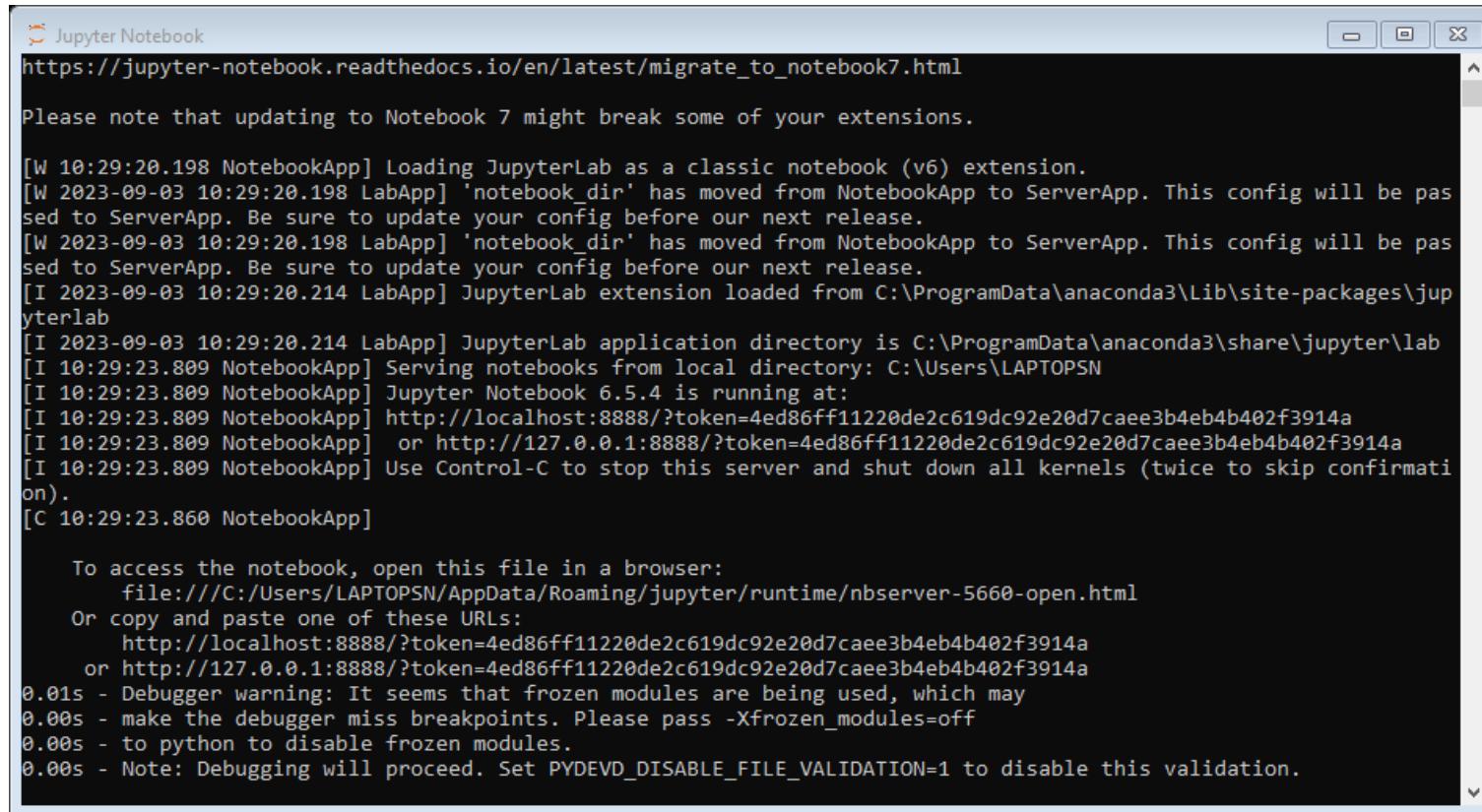
B3: Chạy Jupyter notebook

- Mặc định thư mục gốc của Jupyter notebook là thư mục C:/



Chuẩn bị môi trường thực hành

- Lưu ý khi chạy Jupyter notebook, không tắt cửa sổ biên dịch cmd của ứng dụng; cửa sổ được mở trong suốt quá trình sử dụng công cụ này!



The screenshot shows a terminal window titled "Jupyter Notebook" with the URL https://jupyter-notebook.readthedocs.io/en/latest/migrate_to_notebook7.html. The window displays a warning message: "Please note that updating to Notebook 7 might break some of your extensions." followed by several log entries from the JupyterLab application. The log includes messages about loading JupyterLab as a classic notebook (v6) extension, moved 'notebook_dir' from NotebookApp to ServerApp, and successful loading of the JupyterLab extension. It also shows the application directory as C:\ProgramData\anaconda3\share\jupyter\lab, serving notebooks from local directory C:\Users\LAPTOPSN, and the Jupyter Notebook 6.5.4 is running at http://localhost:8888/?token=4ed86ff11220de2c619dc92e20d7caee3b4eb4b402f3914a or http://127.0.0.1:8888/?token=4ed86ff11220de2c619dc92e20d7caee3b4eb4b402f3914a. It advises using Control-C to stop the server and shut down all kernels (twice to skip confirmation). The terminal ends with a note to access the notebook in a browser at file:///C:/Users/LAPTOPSN/AppData/Roaming/jupyter/runtime/nbserver-5660-open.html and provides URLs for copying and pasting.

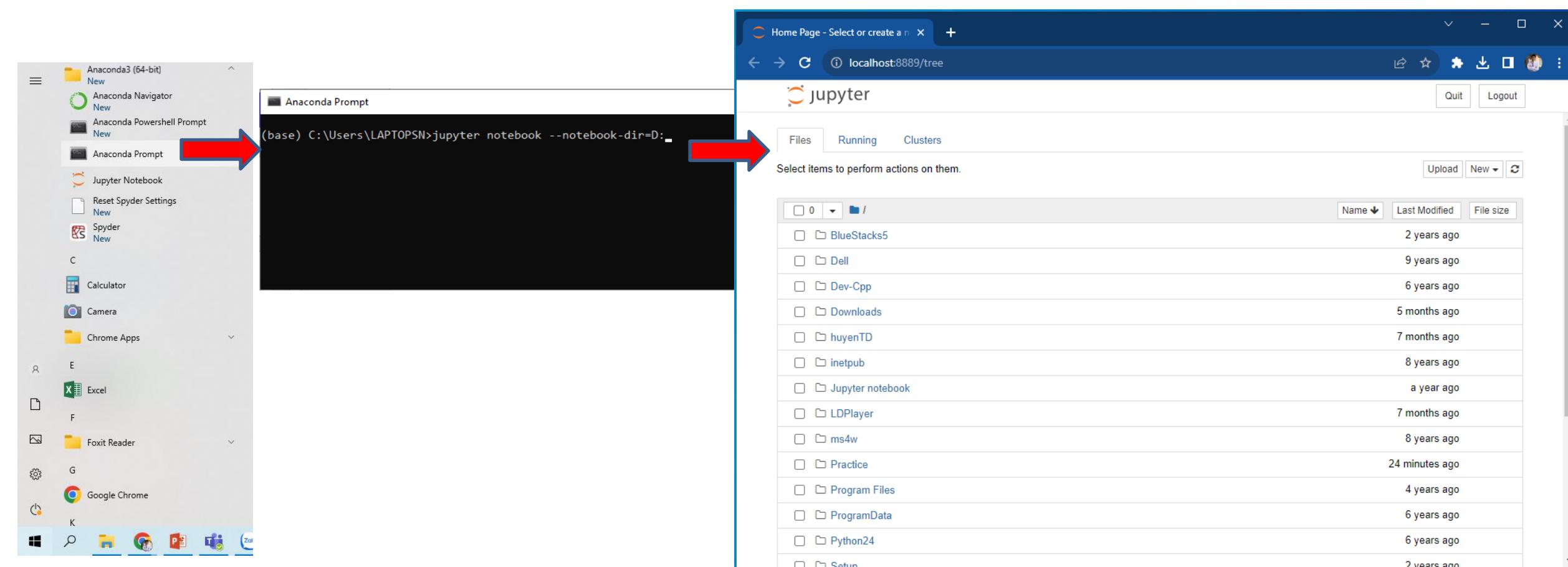
```
[W 10:29:20.198 NotebookApp] Loading JupyterLab as a classic notebook (v6) extension.
[W 2023-09-03 10:29:20.198 LabApp] 'notebook_dir' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2023-09-03 10:29:20.198 LabApp] 'notebook_dir' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2023-09-03 10:29:20.214 LabApp] JupyterLab extension loaded from C:\ProgramData\anaconda3\Lib\site-packages\jupyterlab
[I 2023-09-03 10:29:20.214 LabApp] JupyterLab application directory is C:\ProgramData\anaconda3\share\jupyter\lab
[I 10:29:23.809 NotebookApp] Serving notebooks from local directory: C:\Users\LAPTOPSN
[I 10:29:23.809 NotebookApp] Jupyter Notebook 6.5.4 is running at:
[I 10:29:23.809 NotebookApp] http://localhost:8888/?token=4ed86ff11220de2c619dc92e20d7caee3b4eb4b402f3914a
[I 10:29:23.809 NotebookApp] or http://127.0.0.1:8888/?token=4ed86ff11220de2c619dc92e20d7caee3b4eb4b402f3914a
[I 10:29:23.809 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 10:29:23.860 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/LAPTOPSN/AppData/Roaming/jupyter/runtime/nbserver-5660-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=4ed86ff11220de2c619dc92e20d7caee3b4eb4b402f3914a
or http://127.0.0.1:8888/?token=4ed86ff11220de2c619dc92e20d7caee3b4eb4b402f3914a
0.01s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
```



Chuẩn bị môi trường thực hành

- Trong trường hợp tài nguyên để ở ổ khác (Ví dụ ổ D:/), để khởi động Jupyter Notebook mặc định truy cập đến ổ này. Thực hiện như sau:
 - Chạy Anaconda Prompt - Nhập lệnh: **jupyter notebook --notebook-dir=D:**



Chuẩn bị môi trường thực hành

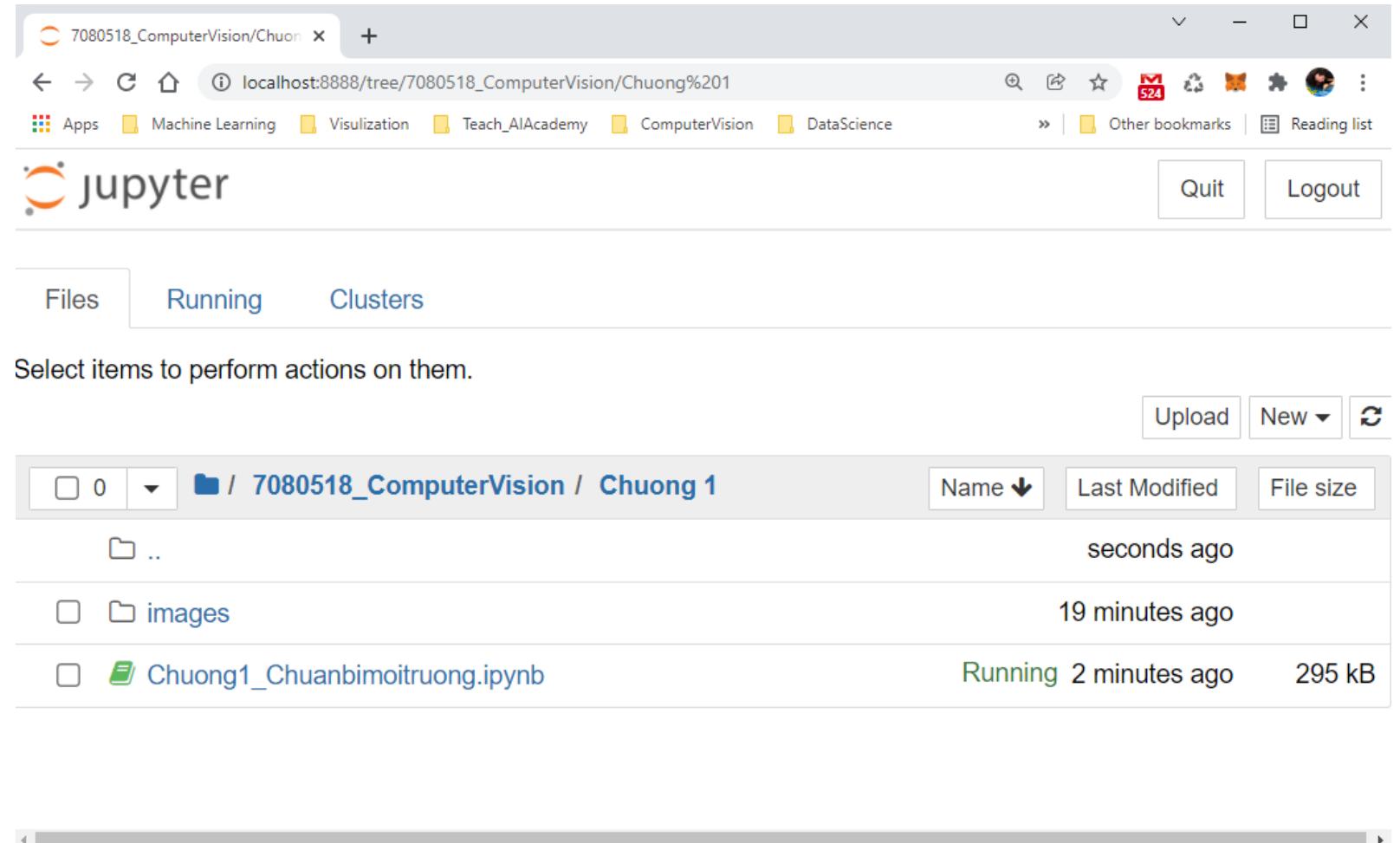
Sinh viên cài đặt Anaconda, sử dụng Jupyter notebook như tài liệu hướng dẫn.

1. Tạo một thư mục

7080518_NhomMH

“Chứa toàn bộ bài học, bài thực hành môn học”.

2. Tải file ảnh và file jupyter của từng chương về thư mục vừa tạo.



Cài đặt và kiểm tra các thư viện sử dụng

Trong khóa học sẽ sử dụng các thư viện sau:

1. OpenCV
2. Numpy
3. Matplotlib

In []: 1 #Cài đặt thư viện OpenCV:
2 #Cài đặt mở rộng của OpenCV
3 !pip3 install --user opencv-contrib-python

1

In []: 1 #Cài đặt thư viện OpenCV với ngôn ngữ Python
2 !pip3 install --user opencv-python

2

In [1]: 1 #Thư viện numpy
2 import numpy as np
3 print('Phiên bản numpy:', np.__version__)

Phiên bản numpy: 1.22.4

In [2]: 1 #Thư viện matplotlib
2 import matplotlib as mpl
3 print('Phiên bản matplotlib:', np.__version__)

Phiên bản matplotlib: 1.22.4

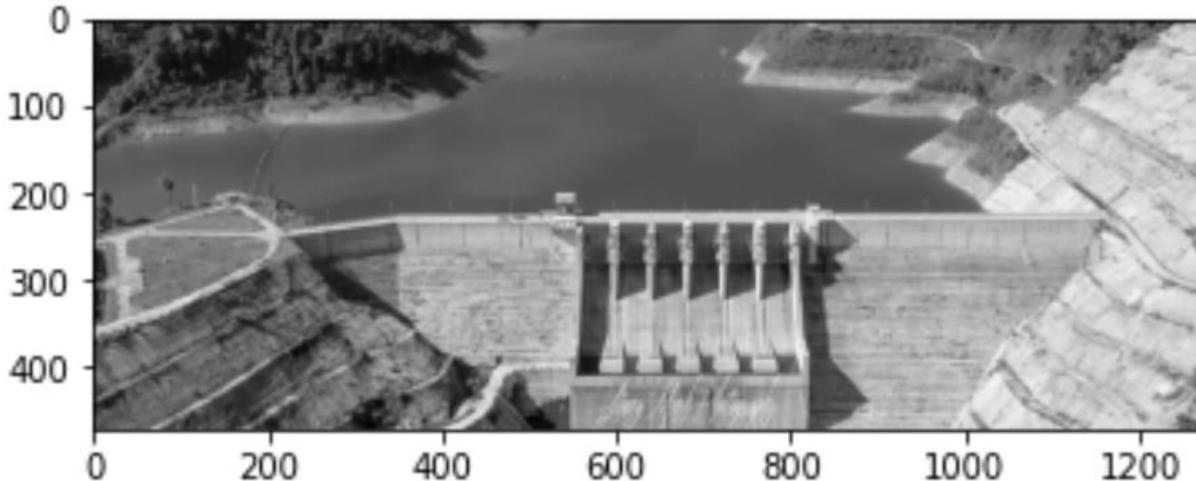
In [3]: 1 #Thư viện opencv
2 import cv2
3 print('Phiên bản opencv:', cv2.__version__)

Phiên bản opencv: 4.5.5

Cài đặt và kiểm tra các thư viện sử dụng

Kiểm tra: Đọc và hiển thị ảnh

```
1 #Đọc ảnh:  
2 img1 = cv2.imread('images/pic1.jpg',0)  
3 #Hiển thị ảnh:  
4 import matplotlib.pyplot as plt  
5 plt.imshow(img,cmap='gray')  
6 plt.show()
```

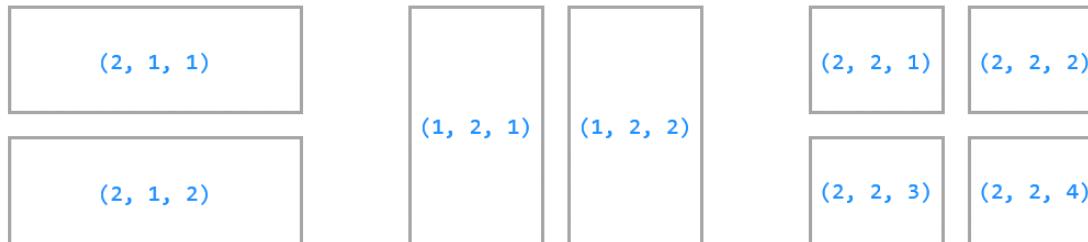


```
1 #Đọc ảnh:  
2 img3 = cv2.imread('images/Pic3.jpg',1)  
3 img3_color = cv2.cvtColor(img3,cv2.COLOR_BGR2RGB)  
4  
5 #Hiển thị ảnh:  
6 import matplotlib.pyplot as plt  
7 plt.figure(figsize=(5,10))  
8 plt.imshow(img3_color,cmap='gray')  
9 plt.show()
```



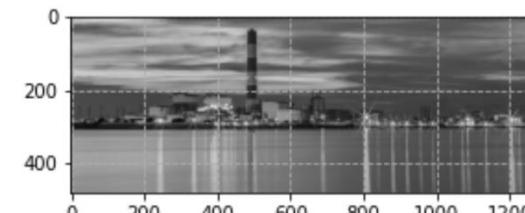
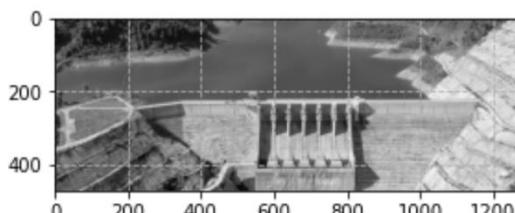
Cài đặt và kiểm tra các thư viện sử dụng

Kiểm tra: Đọc và hiển thị ảnh

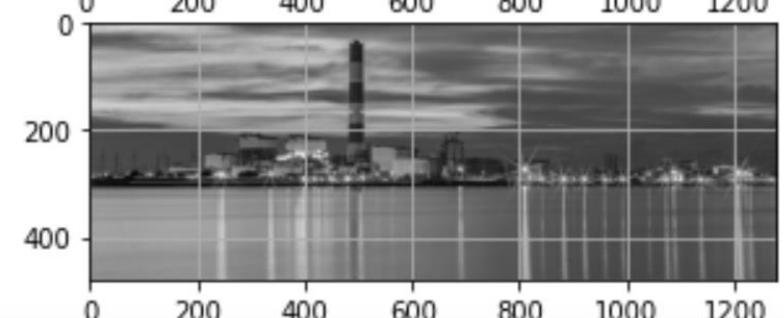
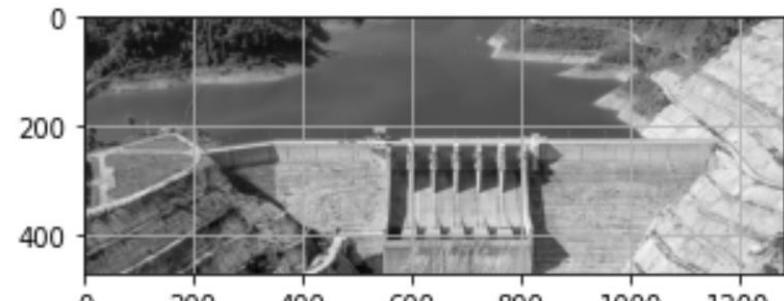


`plt.subplot(row, column, index)`

```
1 plt.figure(figsize=(10,5))
2 #Hiển thị nhiều ảnh:
3 plt.subplot(1,2,1)
4 plt.imshow(img1,cmap='gray')
5 plt.grid(ls='--')
6
7 plt.subplot(1,2,2)
8 plt.imshow(img2,cmap='gray')
9 plt.grid(ls='--')
10
11 plt.show()
```



```
1 #Hiển thị nhiều ảnh:
2 plt.subplot(2,1,1)
3 plt.imshow(img1,cmap='gray')
4 plt.grid()
5
6 plt.subplot(2,1,2)
7 plt.imshow(img2,cmap='gray')
8 plt.grid()
9 plt.show()
```





6. Đọc và hiển thị ảnh với OpenCV



Đọc ảnh với các tùy chọn khác nhau

Hàm imread():

cv2.imread(image_path, color_flag)

Color_flag:

- **cv2.IMREAD_COLOR**: Đọc ảnh màu, bỏ qua kênh alpha (kênh transparency – độ trong suốt). Là giá trị mặc định.
- **cv2.IMREAD_GRAYSCALE**: Đọc ảnh vào thành ảnh mức xám.
- **cv2.IMREAD_UNCHANGED**: Đọc ảnh màu, bao gồm cả kênh alpha nếu có.

Thay vì đưa 3 cò, có thể đưa vào các giá trị lần lượt là 1, 0, -1:

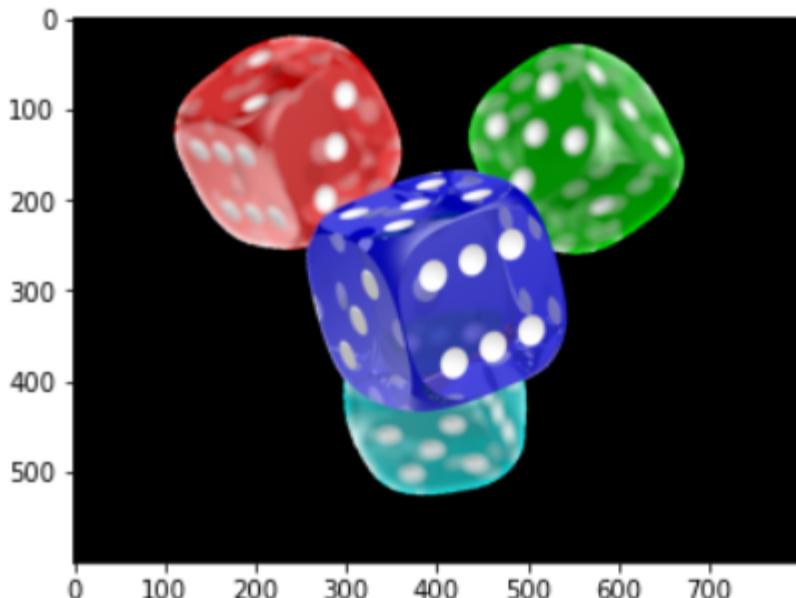
➤ `image = cv2.imread("images/pic_2.png.jpg", cv2.IMREAD_COLOR)`

Tương đương:

➤ `image = cv2.imread("images/pic_2.png", 1)`
➤ `image = cv.imread("images/pic_2.png")`

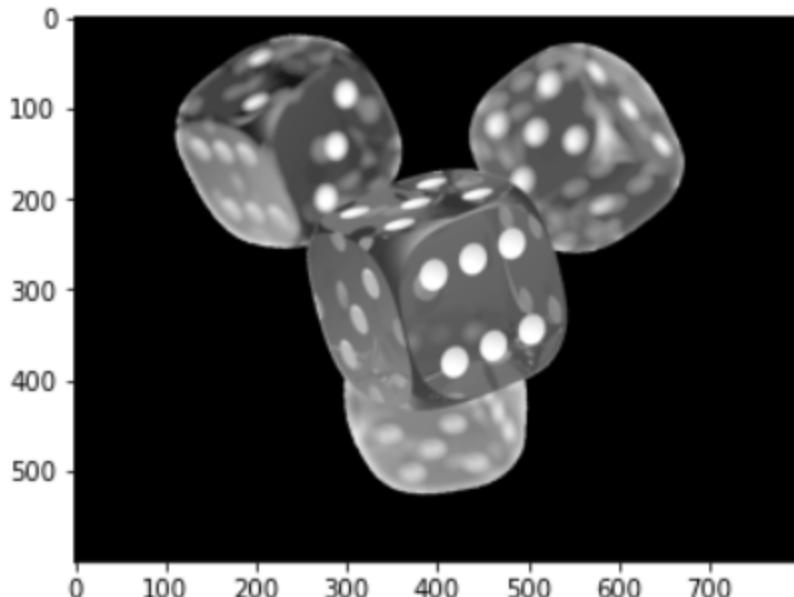
Đọc ảnh với các tùy chọn khác nhau

```
1 #1. Đọc ảnh với chế độ mặc định: Đọc ảnh màu, bỏ qua kênh alpha (kênh transparency - độ trong suốt)
2 img1 = cv2.imread('images/pic_2.png')
3
4 #hoặc:
5 #img1 = cv2.imread('images/pic_2.png',1)
6
7 #hoặc:
8 #img1 = cv2.imread('images/pic_2.png',cv2.IMREAD_COLOR)
9
10 #Hiển thị ảnh:
11 plt.imshow(img1)
12 plt.show()
```

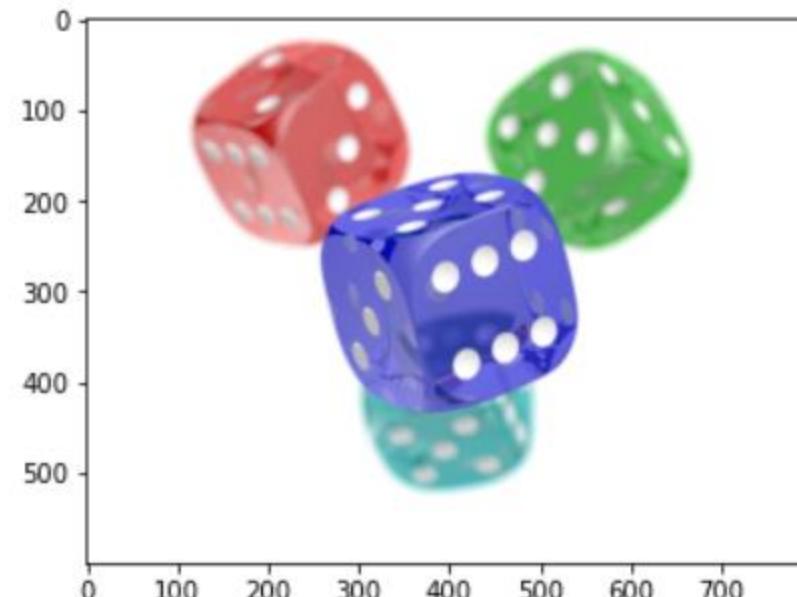


Đọc ảnh với các tùy chọn khác nhau

```
1 #2. Đọc ảnh vào thành ảnh mức xám.  
2 img2 = cv2.imread('images/pic_2.png',0)  
3  
4 #hoặc:  
5 #img2 = cv2.imread('images/pic_2.png',cv2.IMREAD_GRAYSCALE)  
6  
7 #Hiển thị ảnh xám với thư viện Matplotlib  
8 plt.imshow(img2, cmap='gray')  
9 plt.show()
```



```
1 #3. Đọc ảnh màu, bao gồm cả kênh alpha nếu có.  
2 img3 = cv2.imread('images/pic_2.png',-1)  
3  
4 #hoặc:  
5 #img3 = cv2.imread('images/pic_2.png',cv2.IMREAD_UNCHANGED)  
6  
7 #Hiển thị ảnh:  
8 plt.imshow(img3)  
9 plt.show()
```

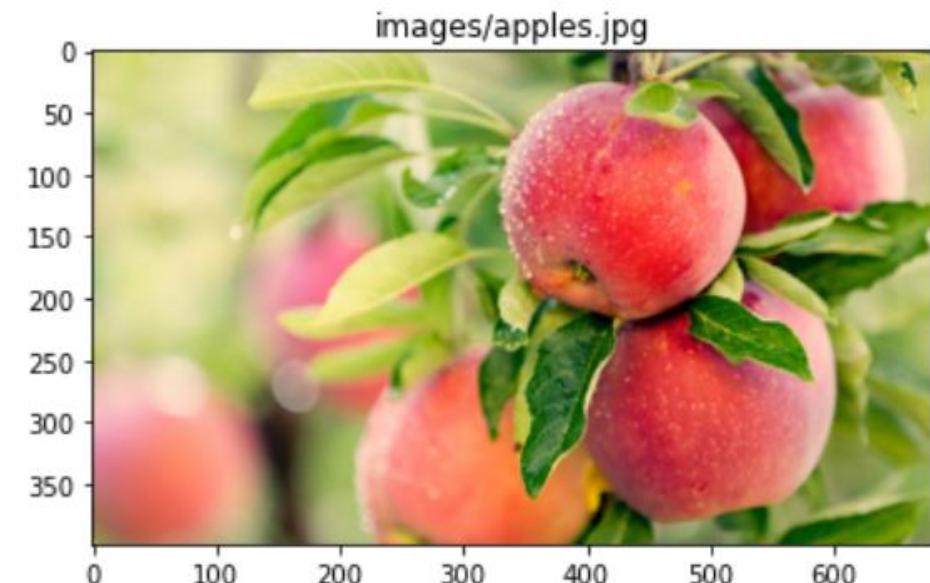


Thực hành số 1.1

Thực hành 1.1

Sinh viên sử dụng ảnh `images/apples.jpg`:

- **Đọc file ảnh và cho biết kích thước ảnh**
- **Hiển thị ảnh với OpenCV**
- **Hiển thị ảnh với Matplotlib**

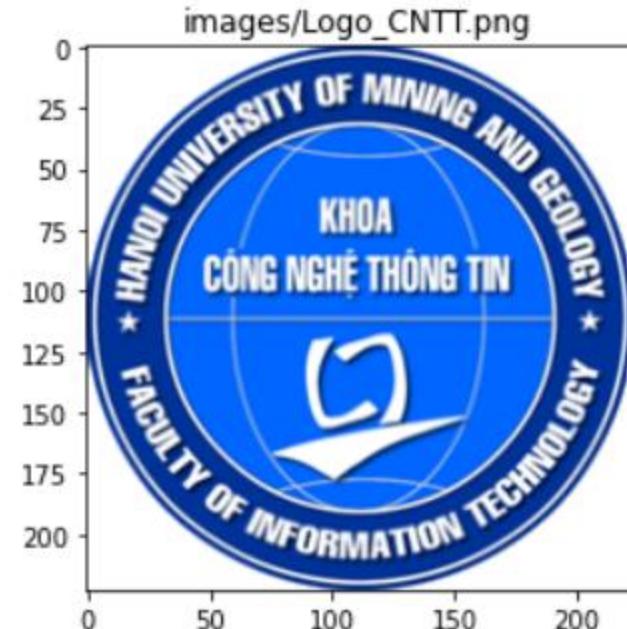
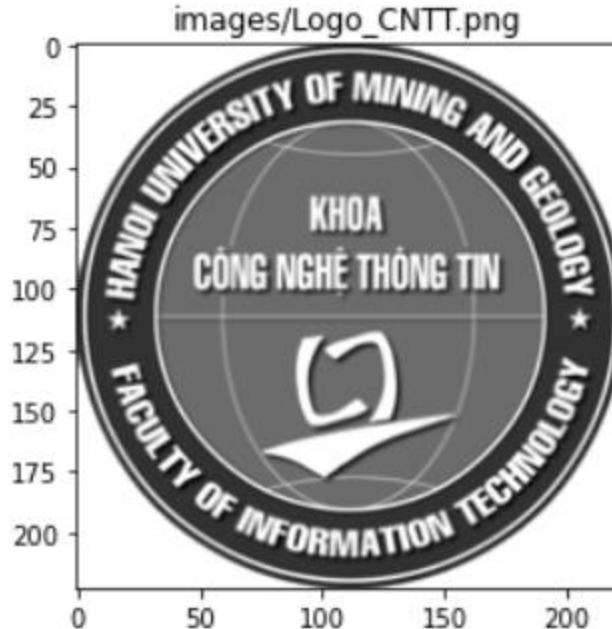
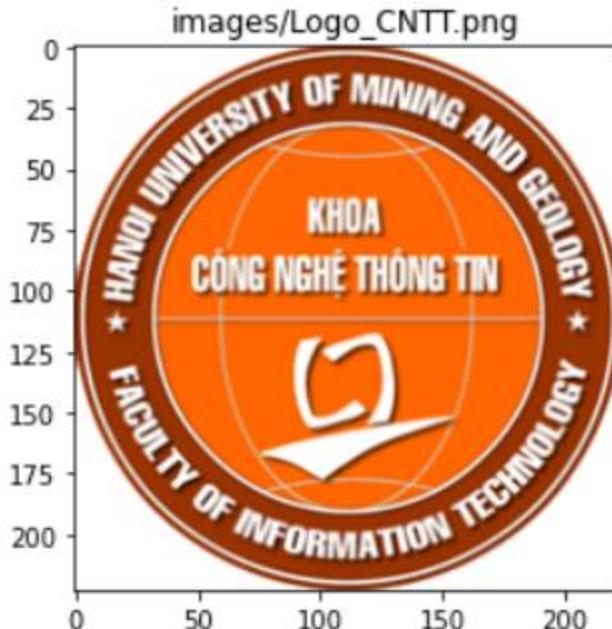


Thực hành số 1.2

Thực hành 1.2

Sinh viên sử dụng ảnh `images/Logo_CNTT.png`:

- Đọc file ảnh và hiển thị ảnh với các chế độ đọc khác nhau





7. Ghi ảnh với OpenCV

Ghi ảnh sử dụng OpenCV

Ghi ảnh:

cv2.imwrite(image_path, img)

**SAVE/WRITE
IMAGE**

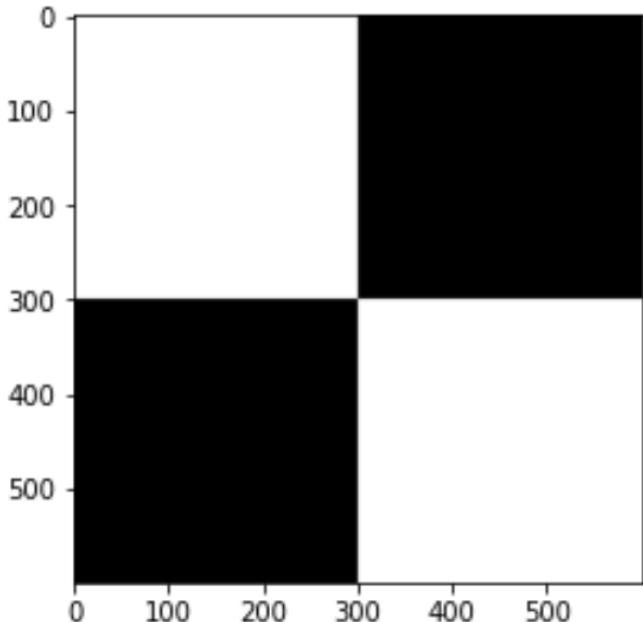


Lưu ý khi ghi ảnh:

- Định dạng của ảnh được ghi theo phần extension trong tên file, tên file không có extension sẽ báo lỗi.
- Hàm imwrite sẽ ghi đè lên ảnh đã tồn tại mà không báo lỗi hay gửi yêu cầu xác nhận ghi đè.
- Trả về true nếu thành công, false nếu ghi thất bại

Ghi ảnh sử dụng OpenCV

```
1 #Tạo ảnh:  
2 anh = np.zeros((600,600))  
3 #Chỉnh sửa ảnh:  
4 anh[0:300,0:300] = 255 #Chuyển góc phần tư thứ nhất thành màu trắng  
5 anh[300:,300:] = 255 #Chuyển góc phần tư thứ 2 thành màu trắng  
6 #Hiển thị ảnh:  
7 plt.imshow(anh,cmap='gray')  
8 plt.show()
```



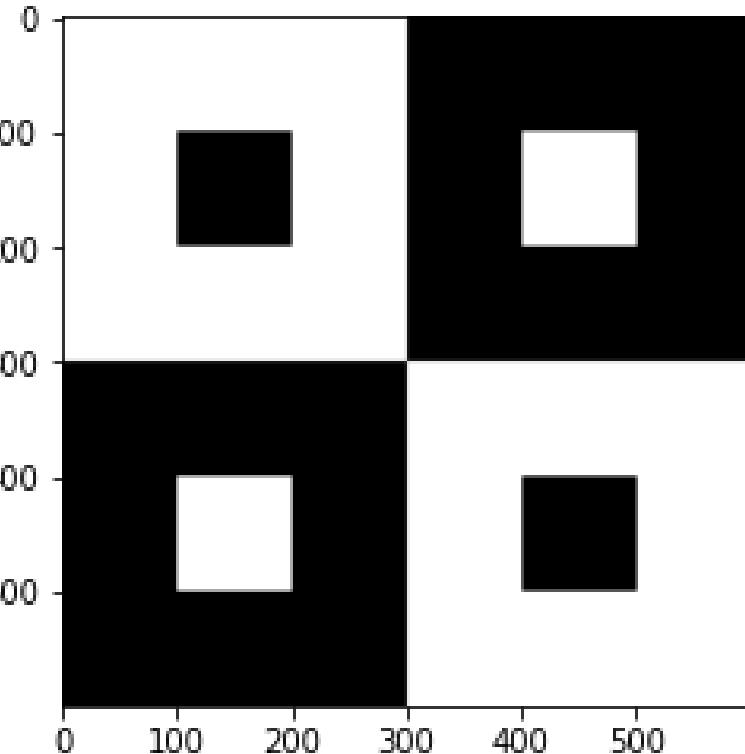
```
1 #Lưu ảnh:  
2 cv2.imwrite('images/anhtao.jpg',anh) #Ghi biên ảnh vào file có tên anhtao.jpg
```

True

Thực hành số 1.3

Thực hành 1.3

Sinh viên sử dụng thư viện numpy tạo một ảnh đen trắng như sau:



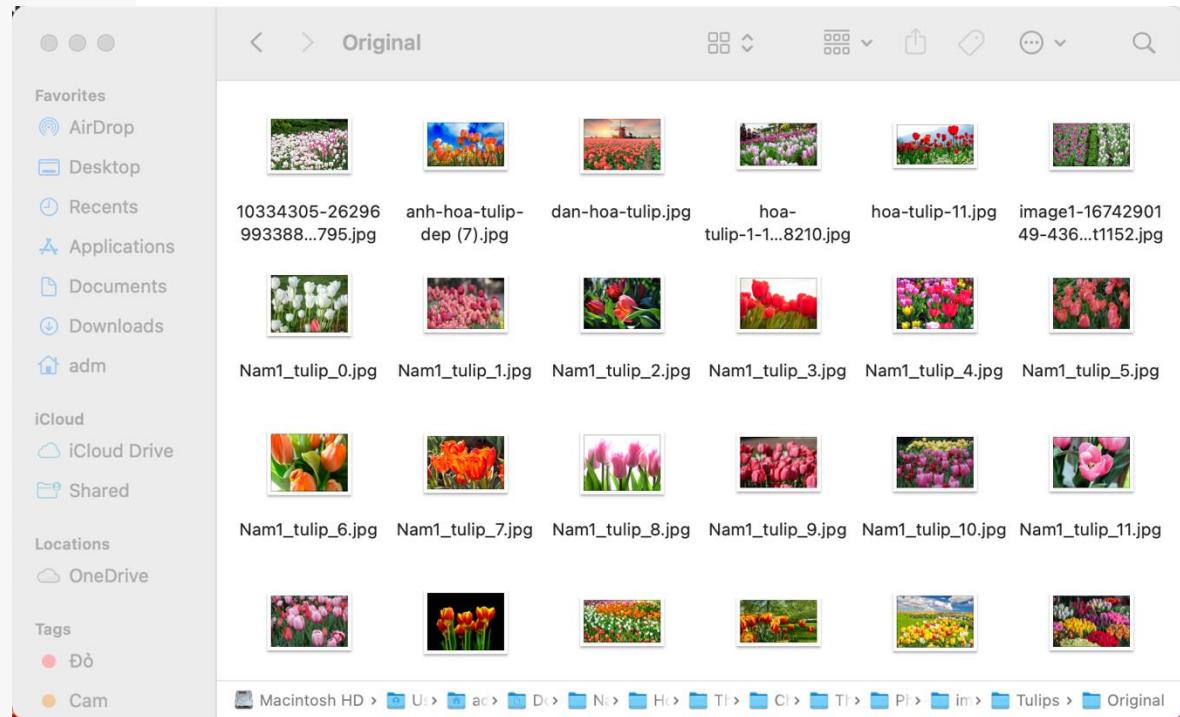
- **Ghi ảnh ra file:**
images/Saves/MSV_TenSV_Chuong1.jpg



8. Đọc và hiển thị nhiều ảnh

Đổi tên file và hiển thị nhiều ảnh trong một thư mục

```
1 #Viết hàm renamefile() đổi tên các file ảnh trong cùng một thư mục
2 import os
3 import shutil
4
5 #Hàm xóa các file trong một thư mục:
6 def delete_files_in_directory(directory_path):
7     files = os.listdir(directory_path)
8     for file in files:
9         file_path = os.path.join(directory_path, file)
10        if os.path.isfile(file_path):
11            os.remove(file_path)
12
13 # Hàm sao chép và đổi tên file:
14 def copy_renamefile(path_old, path_new, name_new):
15     try:
16         #xóa tất cả các file cũ của thư mục:
17         delete_files_in_directory(path_new)
18         #Thiết lập chỉ số cho tên file mới
19         for count, filename in enumerate(os.listdir(path_old)):
20             #Lấy tên file cũ:
21             src = path_old + filename
22             #thiết lập tên file mong muốn đổi:
23             dst = path_new + name_new + str(count) + ".jpg"
24             # Sao chép và Thay thế file cũ bằng file mới
25             dest = shutil.copyfile(src, dst)
26             print('Sao chép và đổi tên file thành công!')
27     except OSError:
28         print('Lỗi xử lý')
```



Đổi tên file và hiển thị nhiều ảnh trong một thư mục

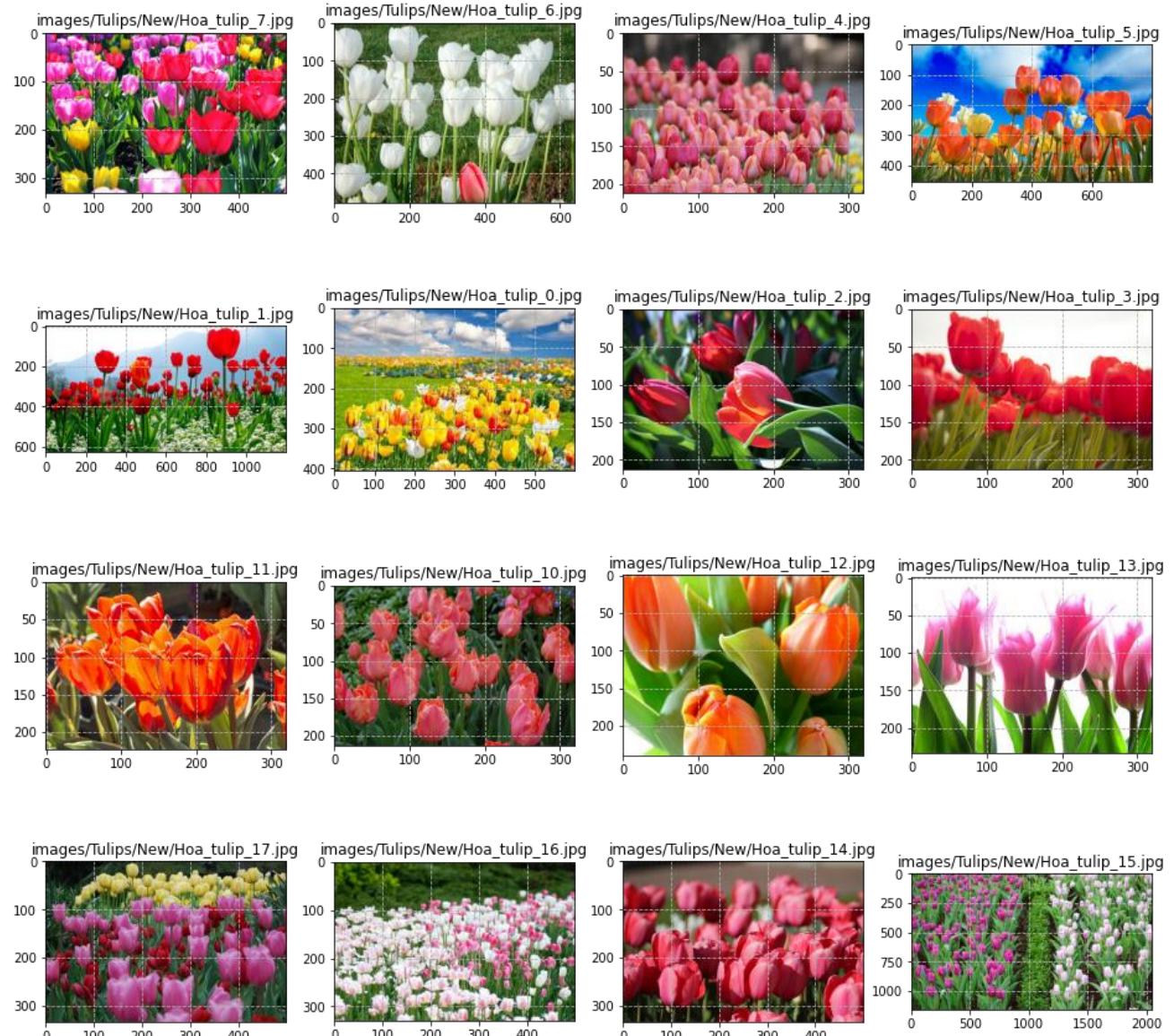
```
1 #Thiết lập đường dẫn và tên file mới:  
2 path_new = 'images/Tulips/New/'  
3 name_new = 'Hoa_tulip_'
```

```
1 #Gọi hàm đổi tên file:  
2 copy_renamefile(path_original,path_new,name_new)  
3  
4 #Hiển thị danh sách file sau khi đã đổi tên.  
5 for count, filename in enumerate(os.listdir(path_new)):  
6     print(count+1,',',filename)
```

Sao chép và đổi tên file thành công!

```
1 ) Hoa_tulip_7.jpg
```

```
1 #HIỂN THỊ ẢNH  
2 import pathlib  
3 #Lấy danh sách tên file có trong thư mục  
4 pic_tulips = list(pathlib.Path(path_new).glob('*'))  
5  
6 #Thiết lập kiểu hiển thị  
7 cot = 4  
8 hang = len(pic_tulips)//cot + 1  
9  
10 #Hiển thị nhiều ảnh sử dụng Matplotlib:  
11 plt.figure(figsize=(4*cot,4*hang))  
12 for count, filename in enumerate(pic_tulips):  
13     #Đọc từng ảnh theo chế độ mặc định  
14     img = cv2.imread(str(filename))  
15     img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
16  
17     #Thiết lập khung hiển thị gồm 5 hàng và 4 cột  
18     ax = plt.subplot(hang,cot,count+1)  
19     #Hiển thị ảnh  
20     plt.imshow(img)  
21     #lấy tên file hiển thị  
22     plt.title(str(filename).split("\\")[-1])  
23     plt.grid(ls='--')  
24     plt.axis('on')  
25  
26 plt.show()
```



Thực hành số 1.4

Thực hành 1.4

Yêu cầu:

1. Sinh viên download từ 20 ảnh khác nhau về một chủ đề bất kỳ (*.jpg) và lưu vào trong thư mục images/Root

2. Sử dụng hàm sao chép và đổi tên các file ảnh đó theo cấu trúc NhómMH_MãSV_i.jpg, i là số tăng dần.

3. Hiển thị toàn bộ ảnh ra một khung đồ thị như ví dụ:





QUESTIONS

Q & A

ANSWERS