

THỊ GIÁC MÁY TÍNH Fall 2023- D20PT		
CVFL23B20DCPT053HW0001	Bài tập chuẩn bị môi trường - vào ra cơ bản với ảnh và video	
Họ và tên: Nguyễn Quốc Đạt	Mã sinh viên: B20DCPT053	Nhóm lớp: 01
Mức độ hoàn thành: 95%		
Đánh giá cụ thể		
Hạng mục	Mức độ thực hiện	
Cài đặt OpenCV cho Python	Cài đặt thành công OpenCV cho Python	
Làm việc với các hàm cơ bản với đối tượng ảnh và video	Sử dụng và hiểu các hàm	
Thực thi một vài ví dụ		

I. Cài đặt OpenCV cho Python:

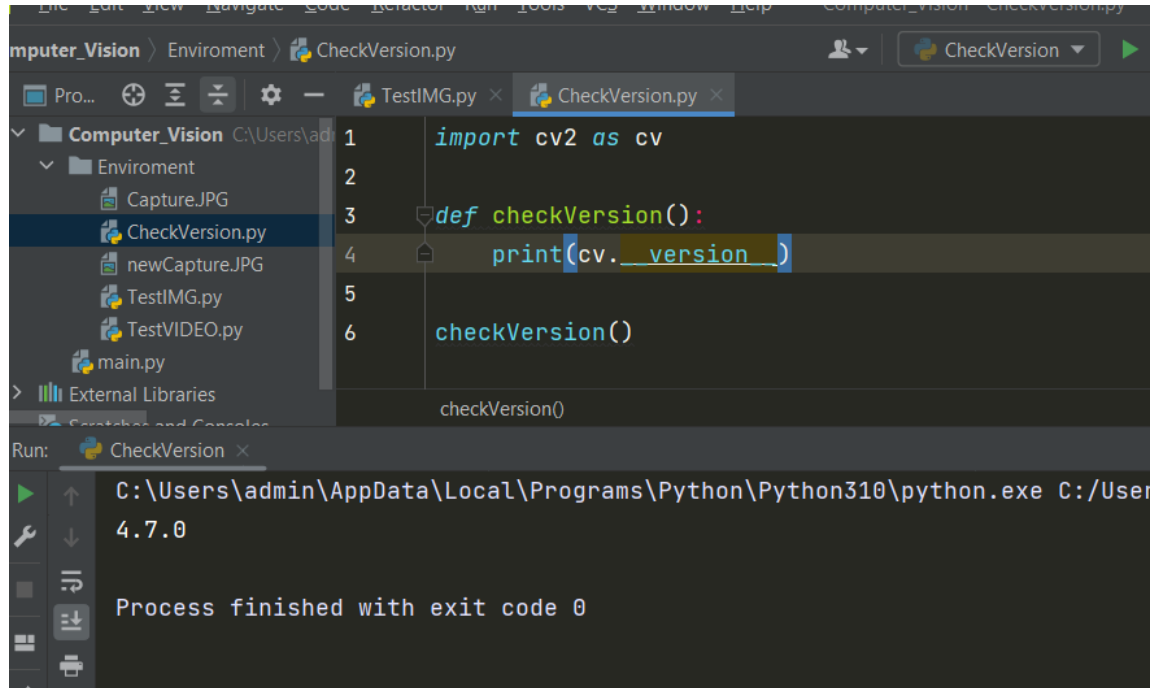
- Mở CMD lên và gõ pip install opencv-python rồi enter.
- Kết quả thu được như hình vẽ.

```
C:\Users\admin>pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\admin\appdata\l
Requirement already satisfied: numpy>=1.21.2 in c:\users\admin\appdata\l

[notice] A new release of pip is available: 23.1.2 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\admin>
```

- Ngoài ra tùy vào khả năng sử dụng của người dùng thì ta có thể Cài Visual Studio 2012 và CMake và OpenCV cho ngôn ngữ C++.
- Import thư viện vào với cú pháp: import cv2 as cv.
- Kiểm tra phiên bản OpenCV với Python:



II. Vào ra với Ảnh

1. Đọc file ảnh ra:

- Thực hiện đọc file ảnh từ các mẫu OpenCV. Để làm như vậy, một lệnh gọi đến hàm **cv::imread** tải hình ảnh bằng cách sử dụng đường dẫn tệp được chỉ định bởi đối số đầu tiên. Đối số thứ hai là tùy chọn và chỉ định định dạng mà chúng ta muốn hình ảnh. Đây có thể là:
 - +) **IMREAD_COLOR** tải hình ảnh ở định dạng BGR 8-bit. Đây là **mặc định**.
 - +) **IMREAD_UNCHANGED** tải hình ảnh nguyên trạng (bao gồm cả kênh alpha nếu có).
 - +) **IMREAD_GRAYSCALE** tải hình ảnh dưới dạng cường độ.

- Cú pháp: `img = cv.imread(cv.samples.findFile("Capture.JPG"))`
- Kiểm tra xem file đọc ra có tồn tại hay hợp lệ hay không:

```
if img is None:
```

```
    sys.exit("Find Not Found and Could not read the image!!")
```

2. Hiện thị file ảnh:

- Sau đó, hình ảnh được hiển thị bằng cách sử dụng lệnh gọi đến hàm **cv::imshow**. Đối số đầu tiên là tiêu đề của cửa sổ và đối số thứ hai là **đối tượng cv::Mat** sẽ được hiển thị.
- Bởi vì chúng tôi muốn cửa sổ của chúng tôi được hiển thị cho đến khi người dùng nhấn một phím (nếu không chương trình sẽ kết thúc quá nhanh), chúng tôi sử dụng **cv::hàm waitKey** có tham số duy nhất chỉ là nó nên đợi bao lâu cho đầu vào của người dùng (được đo bằng mili giây). Số 0 có nghĩa là chờ đợi mãi mãi. Giá trị trả về là phím đã được nhấn.

- Cú pháp:

```
cv.imshow("Display Window", img)
k = cv.waitKey(0)
```

3. Ghi file ảnh:

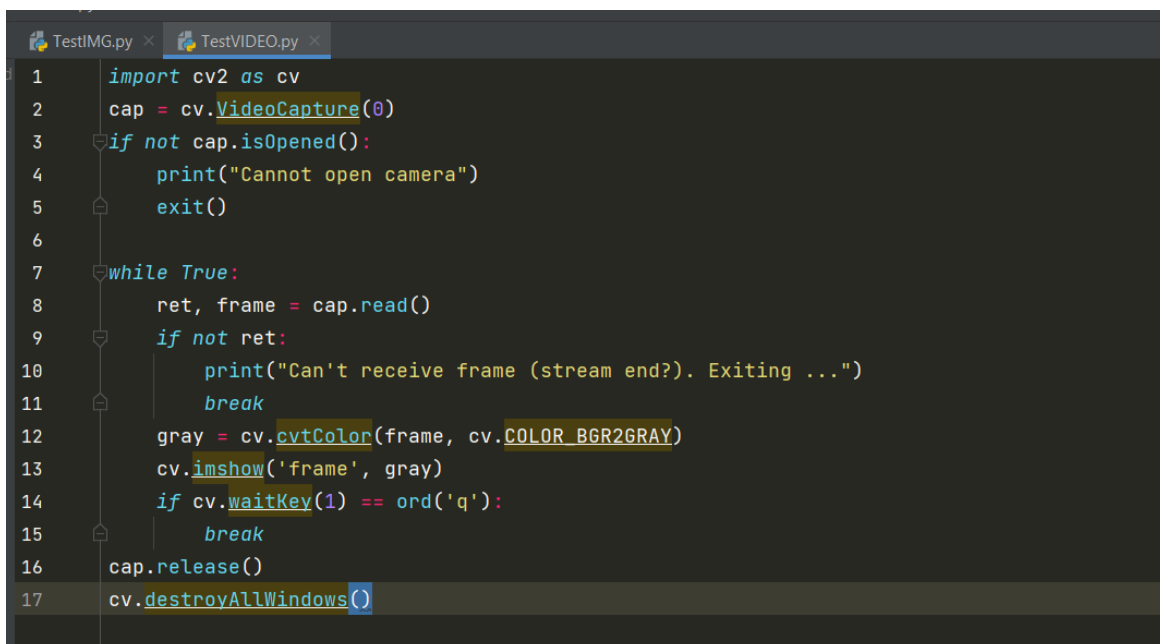
- Cuối cùng, hình ảnh được ghi vào một tệp nếu phím được nhấn là phím "s". Đối với điều này, hàm `cv::imwrite` được gọi là có đường dẫn tệp và đối tượng `cv::Mat` làm đối số.

- Cú pháp:

```
if k == ord('s'):
    cv.imwrite("newCapture.JPG", img)
```

III. Vào ra với Video

1. Quay video từ camera tích hợp:
 - Thông thường, chúng ta phải quay luồng trực tiếp bằng máy ảnh. OpenCV cung cấp một giao diện rất đơn giản để làm điều này. Hãy quay video từ máy ảnh (tôi đang sử dụng webcam tích hợp trên máy tính xách tay của mình), chuyển đổi nó thành video thang độ xám và hiển thị nó. Chỉ là một nhiệm vụ đơn giản để bắt đầu.
 - Để quay video, bạn cần tạo một **Đối tượng VideoCapture**. Đối số của nó có thể là chỉ mục thiết bị hoặc tên của tệp video. Chỉ mục thiết bị chỉ là số để chỉ định máy ảnh nào. Thông thường một camera sẽ được kết nối (như trong trường hợp của tôi). Vì vậy, tôi chỉ cần vượt qua 0 (hoặc -1). Bạn có thể chọn camera thứ hai bằng cách truyền 1, v.v. Sau đó, bạn có thể chụp từng khung hình. Nhưng cuối cùng, đừng quên phát hành bản chụp.
 - Cú pháp:



```
1 import cv2 as cv
2 cap = cv.VideoCapture(0)
3 if not cap.isOpened():
4     print("Cannot open camera")
5     exit()
6
7 while True:
8     ret, frame = cap.read()
9     if not ret:
10        print("Can't receive frame (stream end?). Exiting ...")
11        break
12    gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
13    cv.imshow('frame', gray)
14    if cv.waitKey(1) == ord('q'):
15        break
16 cap.release()
17 cv.destroyAllWindows()
```

- `cap.read()` Trả về bool (/). Nếu khung được đọc chính xác, nó sẽ là . Vì vậy, bạn có thể kiểm tra phần cuối của video bằng cách kiểm tra giá trị trả về này. `TrueFalseTrue`

- Đôi khi, cap có thể không khởi tạo chụp. Trong trường hợp đó, mã này hiển thị lỗi. Bạn có thể kiểm tra xem nó có được khởi tạo hay không bằng phương thức **cap.isOpened()**. Nếu nó là , OK. Nếu không, hãy mở nó bằng **cap.open().True**
- Ta cũng có thể truy cập một số tính năng của video này bằng phương thức **cap.get(propId)** trong đó propId là một số từ 0 đến 18. Mỗi số biểu thị một thuộc tính của video (nếu có thể áp dụng cho video đó). Chi tiết đầy đủ có thể được xem ở đây: **cv::VideoCapture::get()**. Một số giá trị này có thể được sửa đổi bằng cách sử dụng **cap.set (propId, value)**. Giá trị là giá trị mới mà bạn muốn.
- Ví dụ: tôi có thể kiểm tra chiều rộng khung hình và chiều cao bằng và . Nó cung cấp cho tôi 640x480 theo mặc định. Nhưng tôi muốn sửa đổi nó thành 320x240. Chỉ cần sử dụng và

```
cap.get(cv.CAP_PROP_FRAME_WIDTH)cap.get(cv.CAP_PROP_FRAME_HEIGHT)ret
= cap.set(cv.CAP_PROP_FRAME_WIDTH,320)ret
= cap.set(cv.CAP_PROP_FRAME_HEIGHT,240)
```

2. Phát video từ file:

- Phát video từ tệp cũng giống như chụp video từ máy ảnh, chỉ cần thay đổi chỉ mục máy ảnh thành tên tệp video. Ngoài ra trong khi hiển thị khung, hãy sử dụng thời gian thích hợp cho . Nếu nó quá ít, video sẽ rất nhanh và nếu quá cao, video sẽ chậm (Chà, đó là cách bạn có thể hiển thị video ở chế độ chuyển động chậm). 25 mili giây sẽ ổn trong trường hợp bình thường. **cv.waitKey()**
- Cú pháp:

```
def TestPlayVideo():
    cap = cv.VideoCapture('vtest.avi')
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            print("Can't receive frame (stream end?). Exiting ...")
            break
        gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
        cv.imshow('frame', gray)
        if cv.waitKey(1) == ord('q'):
            break
    cap.release()
    cv.destroyAllWindows()
```

3. Lưu video:

- Lần này chúng ta tạo một đối tượng **VideoWriter**. Chúng ta nên chỉ định tên tệp đầu ra (ví dụ: đầu ra.avi). Sau đó, chúng ta nên chỉ định mã **FourCC** (chi tiết trong đoạn tiếp theo). Sau đó, số khung hình mỗi giây (fps) và kích thước khung hình sẽ được thông qua. Và cái cuối

cùng là cờ **isColor**. Nếu nó là , bộ mã hóa mong đợi khung màu, nếu không nó hoạt động với khung thang độ xám. **True**

- **FourCC** là mã 4 byte được sử dụng để chỉ định codec video. Danh sách các mã có sẵn có thể được tìm thấy trong fourcc.org. Nó phụ thuộc vào nền tảng. Các codec sau đây hoạt động tốt đối với tôi.

+) Trong Fedora: DIVX, XVID, MJPG, X264, WMV1, WMV2. (XVID thích hợp hơn. MJPG dẫn đến video kích thước cao. X264 cho video kích thước rất nhỏ)

+) Trong Windows: DIVX (Nhiều hơn để được kiểm tra và thêm vào)

+) Trong OSX: MJPG (.mp4), DIVX (.avi), X264 (.mkv).

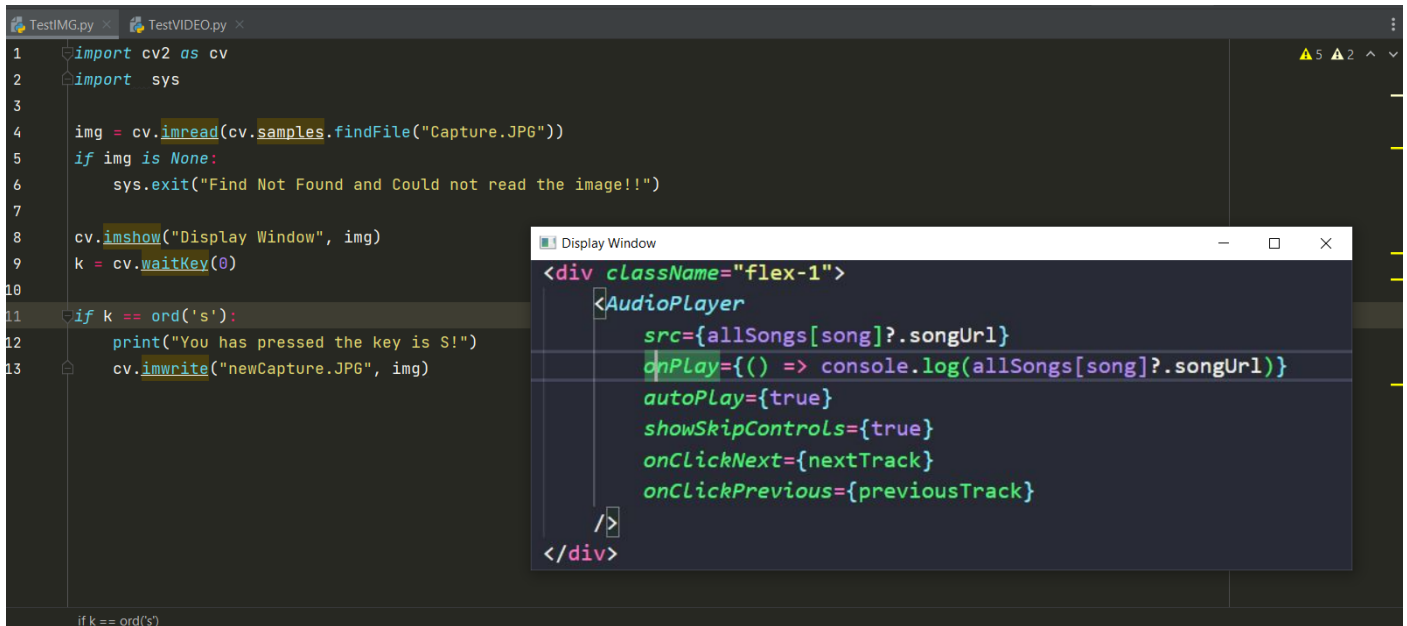
- Mã FourCC được truyền là :

'cv.VideoWriter_fourcc('M','J','P','G')cv.VideoWriter_fourcc(*'MJPG')' cho MJPG

- Cú pháp:

```
def TestSaveVideo():
    cap = cv.VideoCapture(0)
    fourcc = cv.VideoWriter_fourcc(*'XVID')
    out = cv.VideoWriter('output.avi', fourcc, 20.0, (640, 480))
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            print("Can't receive frame (stream end?). Exiting ...")
            break
        frame = cv.flip(frame, 0)
        out.write(frame)
        cv.imshow('frame', frame)
        if cv.waitKey(1) == ord('q'):
            break
    cap.release()
    out.release()
    cv.destroyAllWindows()
```

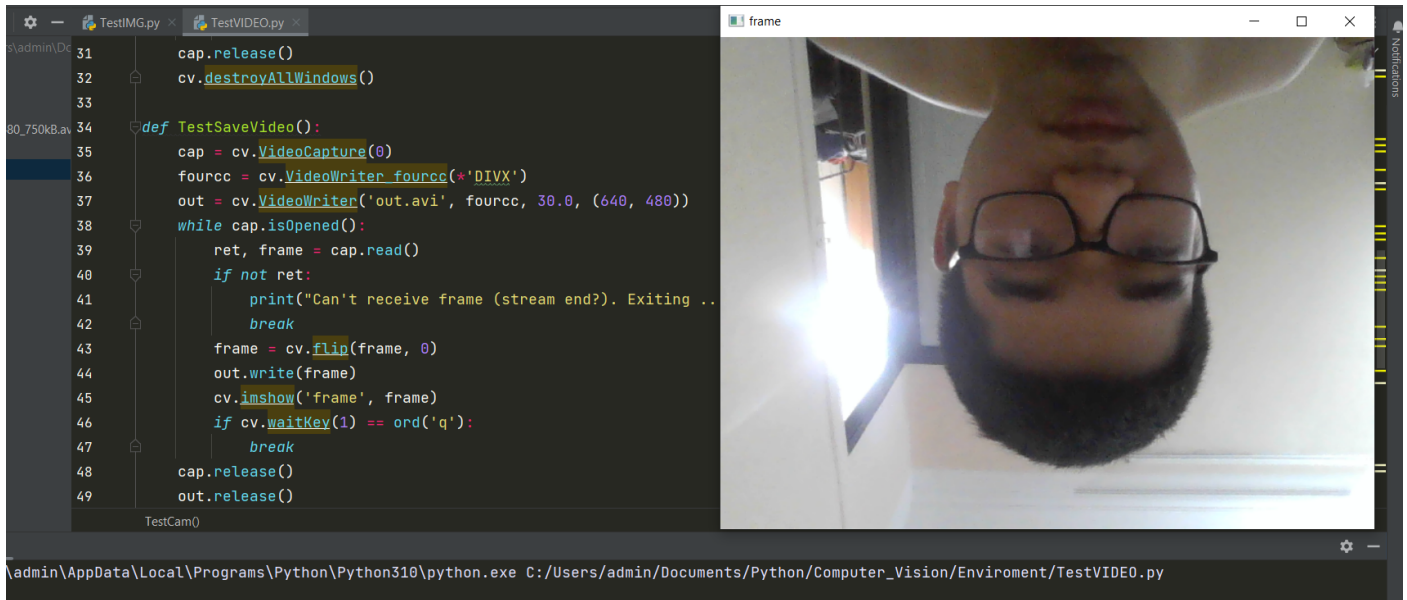
Các thử nghiệm
Với ảnh:



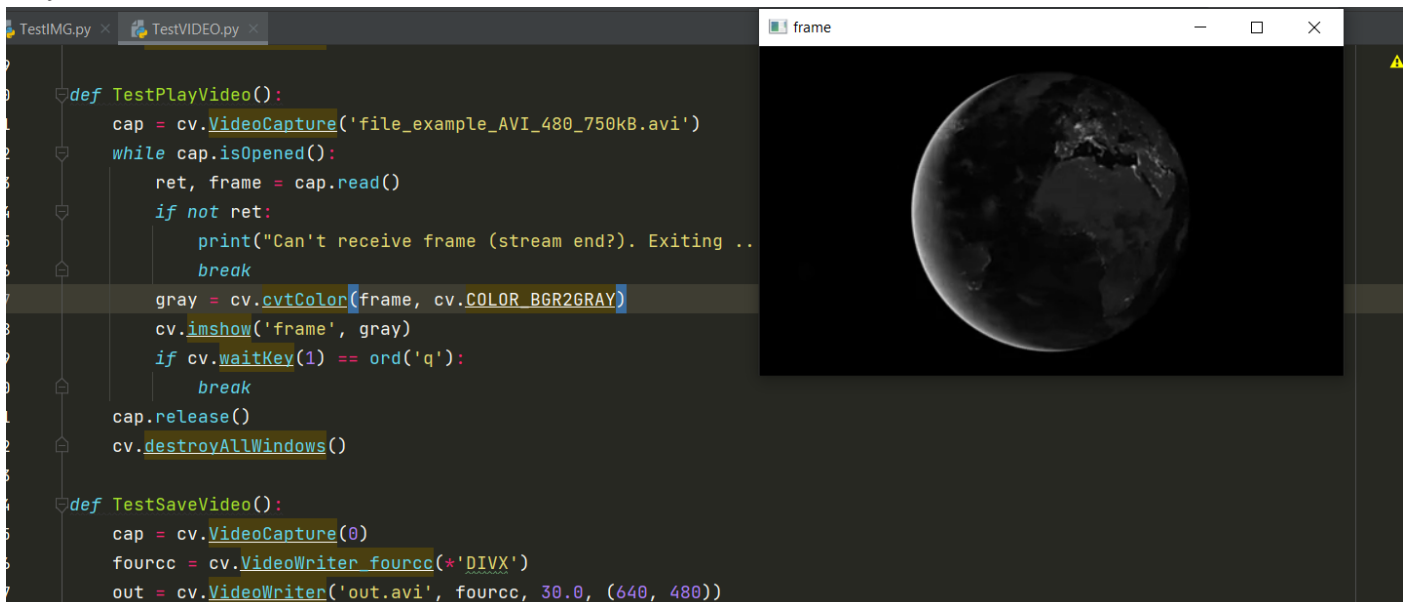
Với Camera:



Save Video:



Play Video:



-----HẾT-----