



☎ 086.262.8846



(https://pivietnam.com.vn/)

Từ khóa tìm kiếm



Trang chủ (https://pivietnam.com.vn/) / Tin tức công nghệ (https://pivietnam.com.vn/tin-tuc-cong-nghe)

/ Nhận diện khuôn mặt với OpenCV trên Raspberry Pi (Phần 1)

TIN MỚI



VIDEO



Nhận diện khuôn mặt với OpenCV trên Raspberry Pi (Phần 1)

🕒 16:20 - 13/03/2019

Nhận diện khuôn mặt với OpenCV trên Raspberry Pi (Phần 1)

» NVIDIA Jetson AGX Orin hiệu suất AI mạnh nhất của NVIDIA Jetson Family với 275 TOPS, 2048 NVIDIA® CUDA® cores, 64 Tensor Cores (https://pivietnam.com.vn/nvidia-jetson-agx-orin-hieu-suot-ai-manh-nhat-cua-nvidia-jetson-family-with-275-tops-2048-nvidia-cuda-cores-64-tensor-cores-pivietnam-com-vn.html)

» NVIDIA Jetson AGX Orin Hardware Layout and GPIO Expansion Header Pinout (https://pivietnam.com.vn/nvidia-jetson-agx-orin-hardware-layout-and-gpio-expansion-header-pinout-pivietnam-com-vn.html)

» NVIDIA Jetson là nền tảng hàng đầu thế giới dành cho Trí tuệ nhân tạo (AI) và Deep Learning (https://pivietnam.com.vn/nvidia-jetson-la-nen-tang-hang-dau-the-gioi-danh-cho-tri-tue-nhan-cao-ai-va-deep-learning-pivietnam-com-vn.html)

» Cách sử dụng Camera CSI với hệ điều hành Raspberry Pi Bullseyes mới nhất (https://pivietnam.com.vn/cach-su-dung-camera-csi-voi-he-dieu-hanh-raspberry-pi-bulleyes-moi-nhat-pivietnam-com-vn-mlab-vn.html)

» Jetson Stats dùng cho Giám sát và Điều khiển trên NVIDIA Jetson Ecosystem [Xavier NX, Nano, AGX Xavier, TX1, TX2] (https://pivietnam.com.vn/jetson-stats-dung-cho-giam-sat-va-dieu-khien-tren-nvidia-jetson-ecosystem-xavier-nx-nano-agx-xavier-tx1-tx2-pivietnam-com-vn.html)

Nhận diện khuôn mặt với OpenCV trên Raspberry Pi (Phần 1)

Xin chào các bạn!

086.262.8846 (Mr Thùy)

Message us

(https://m.me/599911456)

GetButton

Ở phần trước mình đã hướng dẫn các bạn cài đặt thư viện Opencv dùng cho việc xử lý ảnh. Cụ thể hôm nay mình sẽ giới thiệu đến các bạn một số tính năng cơ bản như nhận diện khuôn mặt với đầu vào là một ảnh và hình ảnh thu được trực tiếp từ Camera. Camera cho Raspberry Pi ([Link here \(http://mlab-vn/camera-csi-cho-pi-pivietnam-com-vn\)](http://mlab-vn/camera-csi-cho-pi-pivietnam-com-vn)).

Đầu tiên chúng ta cần tải xuống tệp sau từ Github :

<https://github.com/opencv/opencv/tree/master/data/haarcascades>
(<https://github.com/opencv/opencv/tree/master/data/haarcascades>)

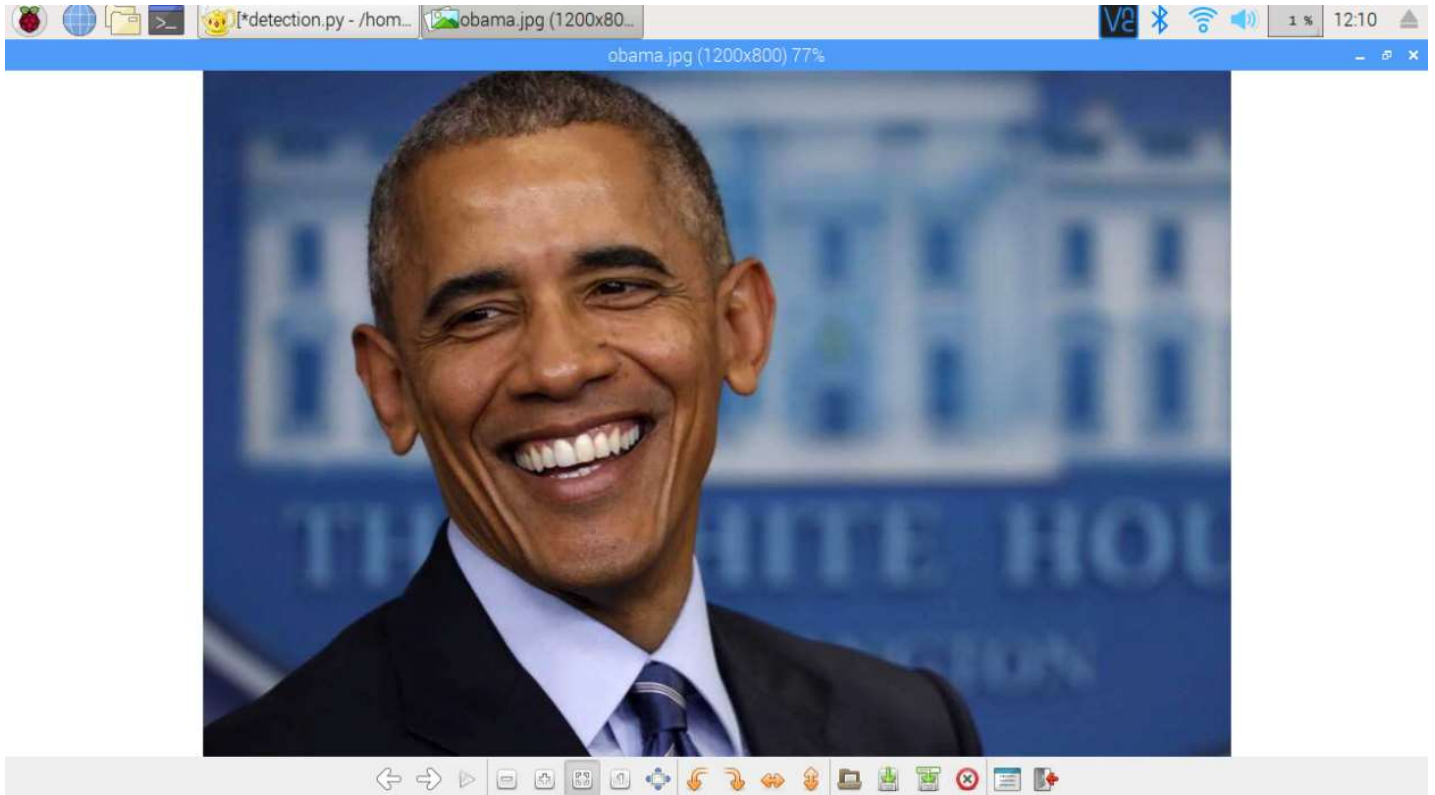
Chúng ta cần sử dụng 2 tệp xml ở đây là :

+ **haarcascade_frontalface_default.xml** : để nhận diện khuôn mặt

+ **haarcascade_eye.xml** : để nhận diện mắt

1. Nhận diện khuôn mặt từ một ảnh.

Các bạn cần chuẩn bị một hình ảnh khuôn mặt (một hoặc nhiều đều ok). Ở đây mình sử dụng một ảnh với tên là "obama.jpg".



Các bạn copy 2 file trên ra Desktop của các bạn sau đó các bạn mở Terminal lên thực hiện thao tác sau :

```
cd Desktop/
```

```
sudo nano face.py
```

Chúng ta vừa tạo ra một file Python có tên là " **face.py** ". Sau đó các bạn gõ đoạn code sau :

```
import cv2
```

086.262.8846 (Mr Thùy)

Message us

GetButton

#Tải một tập tin mô hình nhận diện khuôn mặt và nhận diện mắt

```
face_cascade = cv2.CascadeClassifier('/home/pi/Desktop/haarcascade_frontalface_default.xml')
```

```
eye_cascade = cv2.CascadeClassifier('/home/pi/Desktop/haarcascade_eye.xml')
```

#Đọc hình ảnh đầu vào

```
img = cv2.imread("obama.jpg")
```

```
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

#Tìm kiếm các khuôn mặt trong ảnh bằng cách sử dụng file mô hình nhận diện khuôn mặt đã được tải lên

```
faces = face_cascade.detectMultiScale(gray_img, scaleFactor= 1.3, minNeighbors= 5, minSize = (100, 100), flags = cv2.CA
```

```
print(type(faces))
```

```
print(faces)
```

#Vẽ một hình chữ nhật xung quanh mỗi khuôn mặt được tìm thấy

for x,y,w,h in faces:

```
    roi_gray = gray_img[y:y + h, x:x + w]
```

```
    roi_color = img[y:y + h, x:x + w]
```

```
    img = cv2.rectangle(img, (x,y),(x+w,y+h),(255,0,0), 2)
```

---Các tham số để vẽ một hình chữ nhật

```
# cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
```

img là biến image, nó có thể là " frame" như trong ví dụ này

```
# x1,y1 -----
```

```
# |                |
```

```
# |                |
```

```
# |                |
```

```
# ----- x+w,y+h
```

(255,0,0) là màu (R,G,B)



086.262.8846 (Mr Thùy)

Số 2 cuối cùng trong bộ tham số là độ dày từ 1 đến 3 của đường viền

Message us

(<https://m.me/599911456>)

GetButton

```
eyes = eye_cascade.detectMultiScale(roi_gray)
```

```
for (ex,ey,ew,eh) in eyes:
```

```
    cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
```

```
resized = cv2.resize(img, (int(img.shape[1]/2),int(img.shape[0]/2)))
```

```
#Hiển thị hình ảnh
```

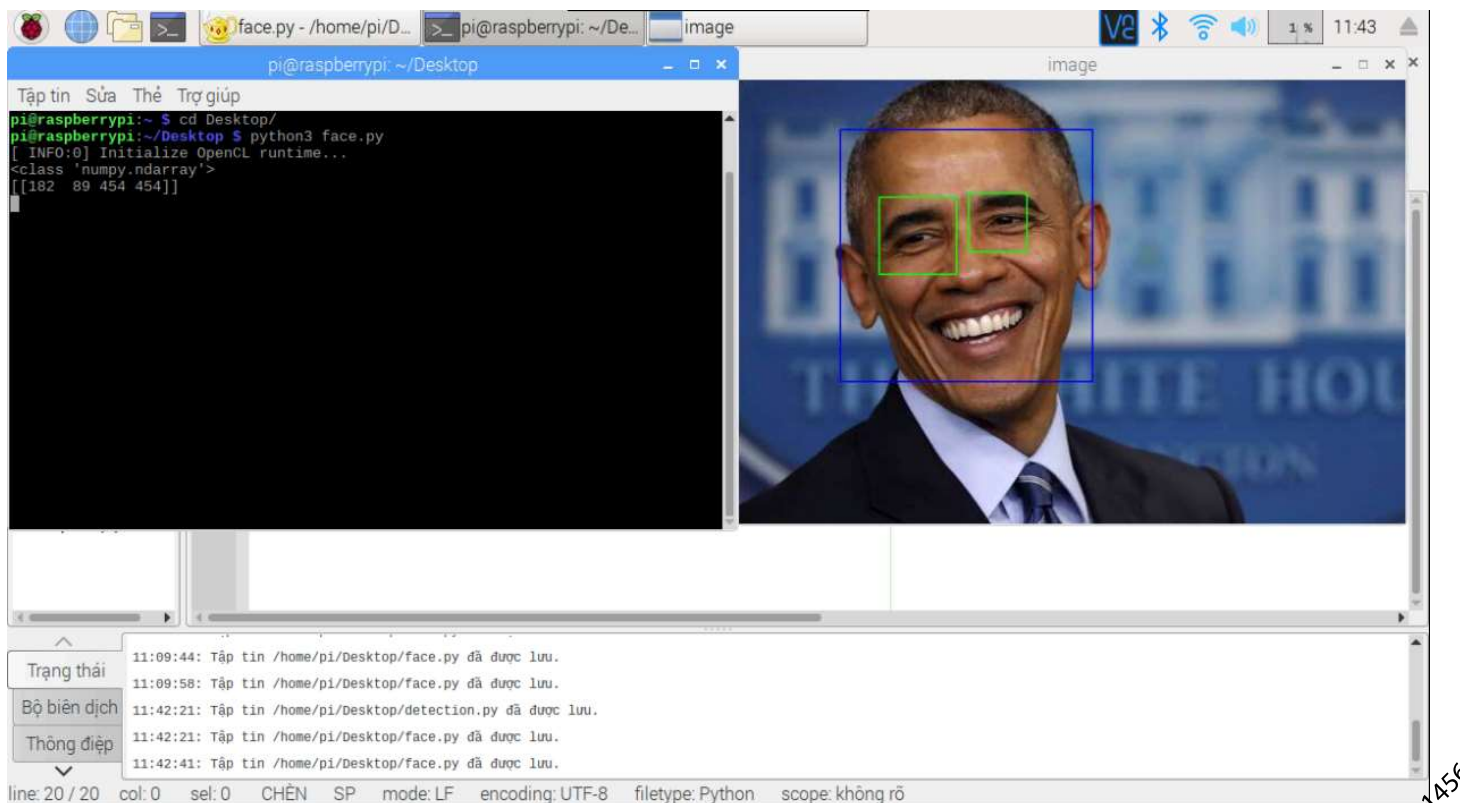
```
cv2.imshow("image",resized)
```

```
cv2.waitKey(1)
```

```
cv2.destroyAllWindows()
```

+ Tiếp theo dùng lệnh : **Python3 face.py** để chạy file.+ Sau đó chúng ta dùng tổ hợp phím **Ctrl + X** để thoát, chọn **Y** và **Enter** để lưu file.


Chúng ta sẽ thu được hình ảnh như sau :



Mình sẽ giải thích ngắn gọn về đoạn code :

+ Đầu tiên chúng ta sử dụng hàm **cv2.imread()** để đọc ảnh.

+ Tiếp theo đó chúng ta chuyển ảnh của chúng ta sang màu xám.

 086.262.8846 (Mr Thùy)

Message us

(<https://m.me/59991456>)

 GetButton

+ Sau đó sử dụng mô hình nhận diện (**face_cascade** và **eye_cascade**) để tìm khuôn mặt khả dĩ nhất trong hình ảnh của chúng ta.

+ Mình có đặt lại kích thước ảnh khi hiển thị bằng cách sử dụng hàm **cv2.resize()**

2. Nhận diện khuôn mặt với hình ảnh trực tiếp từ camera

Giờ chúng ta lắp camera với Raspberry Pi và thử nhận diện khuôn mặt. Về cơ bản thì nguyên lý cũng gần giống như ví dụ trên.

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2

# initialize the camera and grab a reference to the raw camera capture
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 24
rawCapture = PiRGBArray(camera, size=(640, 480))
# Load a cascade file for detecting faces
face_cascade = cv2.CascadeClassifier('/home/pi/Desktop/Face_recognition/haarcascade_frontalface_defa
print("\n [INFO] Initializing face capture. Look the camera and wait ...")
# capture frames from the camera
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    # convert frame to array
    image = frame.array
    # Convert to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # Look for faces in the image using the loaded cascade file
    faces = face_cascade.detectMultiScale(gray, scaleFactor = 1.2, minNeighbors = 5, minSize = (100, 100), flags = cv2.CAS

    print("Found "+str(len(faces))+ " face(s)")
    # Draw a rectangle around every found face
    for (x,y,w,h) in faces:
        roi_gray = gray[y:y + h, x:x + w]
        cv2.rectangle(image,(x,y),(x+w,y+h),(255,0,0),2)
        print(x,y,w,h)
    # display a frame
    cv2.imshow("Frame", image)
    # wait for 'q' key was pressed and break from the loop
    if cv2.waitKey(1) & 0xff == ord("q"):
        exit()
    # clear the stream in preparation for the next frame
    rawCapture.truncate(0)
```

Message us

(<https://m.me/599911456>)



086.262.8846 (Mr Thùy)

GetButton

+ Đầu tiên chúng ta sẽ khởi tạo **PiCamera** và **PiRGBArray** object với kích thước frame là 320×240, framerate = 24.

```
camera = PiCamera()
camera.resolution = (320, 240)
camera.framerate = 24
rawCapture = PiRGBArray(camera, size=(320, 240))
```

+ Tiếp theo chúng ta truy cập vào video stream với method **capture_continuous()**. Method này trả về một infinite iterator của các frame nhận được từ RPi Camera.

```
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
```

+ Tiếp đến chúng ta chuyển frame thành dạng array và chuyển đổi nó sang dạng grayscale.

```
image = frame.array
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

+ Từ tệp mô hình đã tải lên chúng ta sẽ tìm kiếm khuôn mặt.

```
faces = face_cascade.detectMultiScale(gray, scaleFactor = 1.2, minNeighbors = 5, minSize = (100, 100), flags = cv2.C
```

+ Chúng ta sẽ vẽ một hình chữ nhật xung quanh khuôn mặt mà chúng ta tìm được bằng vòng lặp :

```
for (x,y,w,h) in faces:
    roi_gray = gray[y:y+h, x:x+w]
    cv2.rectangle(image,(x,y),(x+w,y+h),(255,0,0),2)
```

+ Giờ thì chúng ta sẽ hiển thị các frame :

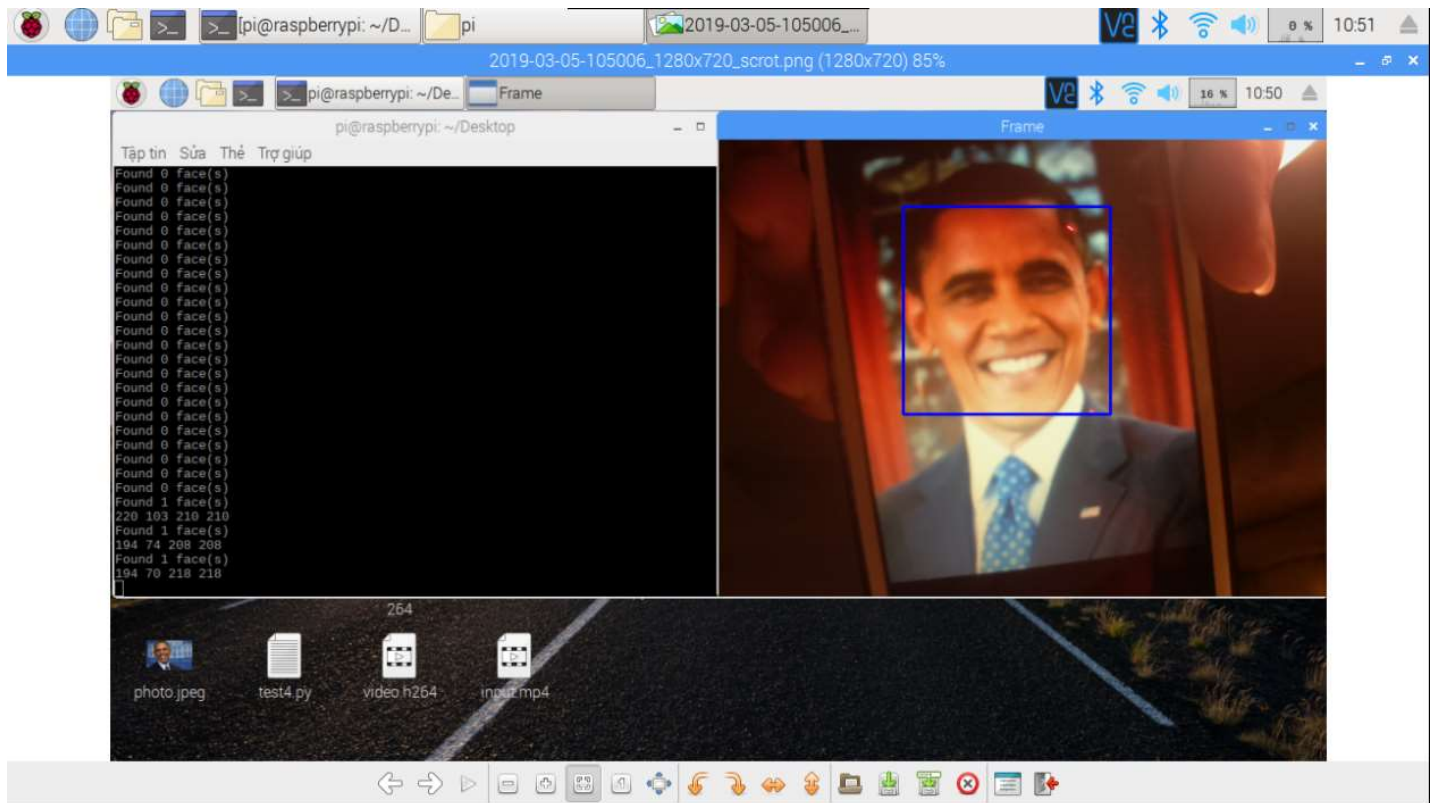
```
cv2.imshow("Frame", image)
```

+ Việc ngắt chương trình có thể thực hiện bằng việc nhấn phím **"q"** hoặc tổ hợp phím **Ctrl + C**

+ Cuối cùng chúng ta bắt buộc phải xóa frame đang được hiển thị trước khi nhận frame tiếp theo.

```
rawCapture.truncate(0)
```

Ta sẽ thu được kết quả như hình :



Các bạn hãy thử các ví dụ vừa rồi và xem kết quả như thế nào nhé !!!

Ở bài viết tiếp theo mình sẽ hướng dẫn các bạn cách nhận dạng các khuôn mặt với OpenCV


Nếu các bạn có bất kỳ ý tưởng mới nào đừng ngần ngại mà hãy inbox trực tiếp cho fanpage (<https://www.facebook.com/mlab.com.vn/>) . Xin chào và hẹn gặp lại các bạn trong các bài viết tiếp theo !!!

PIVIETNAM.COM.VN (.../.../...) CHÚC CÁC BẠN THÀNH CÔNG !!!

Thực hiện bài viết : **Đào Văn Hậu** (<https://www.facebook.com/profile.php?id=100005358645613>)

Để cập nhật các tin tức công nghệ mới các bạn làm theo hướng dẫn sau đây :

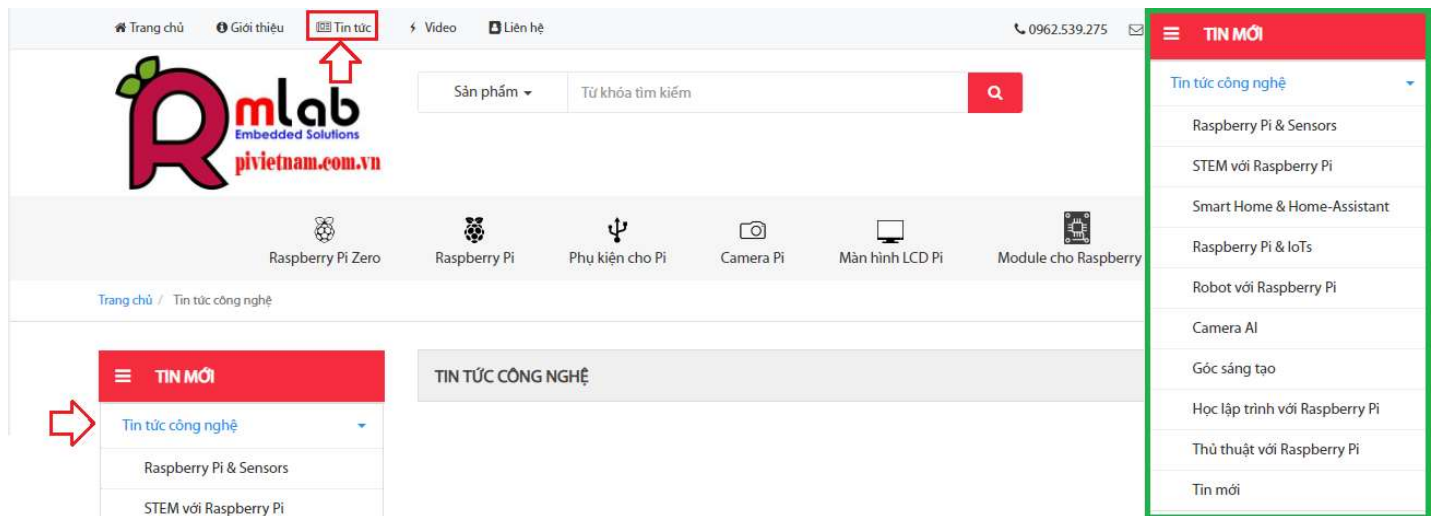
Các bạn vào Trang chủ >> Tin tức. ở mục này có các bài viết kỹ thuật thuộc các lĩnh vực khác nhau các bạn có thể lựa chọn lĩnh vực mà mình quan tâm để đọc nhé !!!

 086.262.8846 (Mr Thùy)

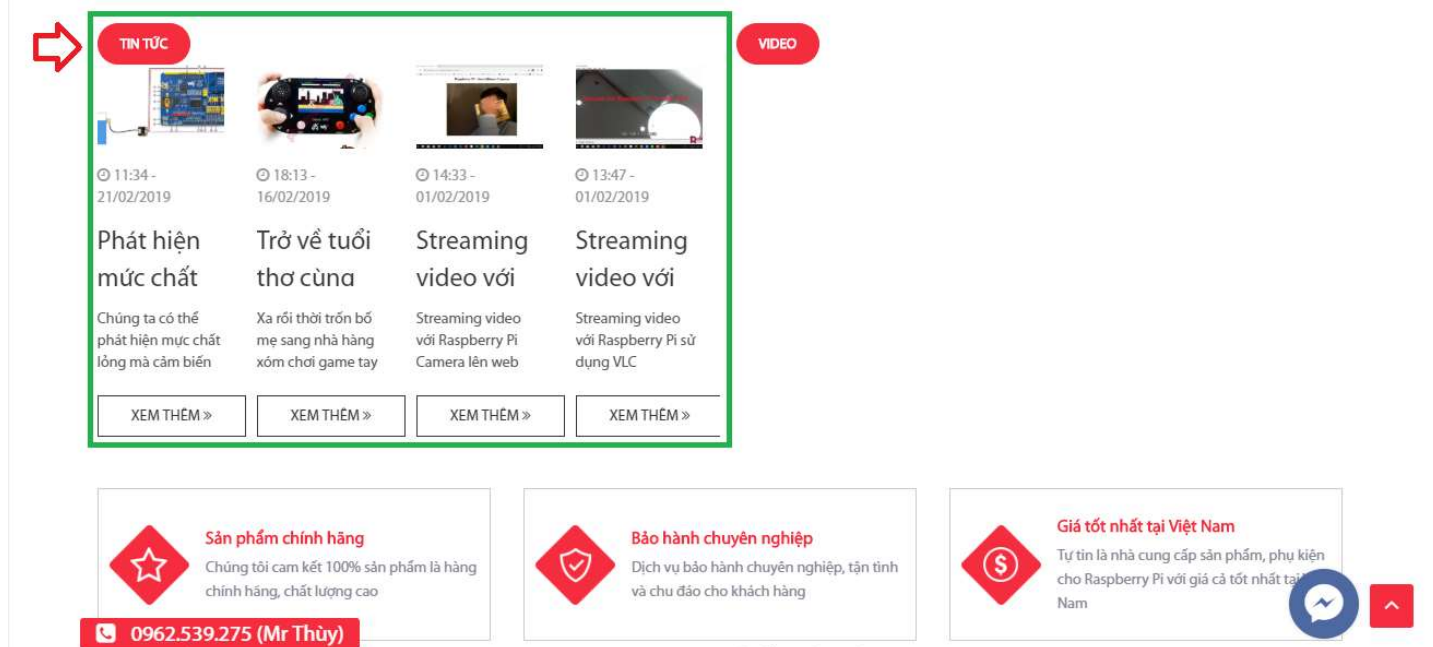
Message us

(<https://m.me/599911456>)

 GetButton



Các bạn cũng có thể kéo xuống cuối trang để xem những tin tức công nghệ mới nhất.



VIDEO



VỀ CHÚNG TÔI

Website uy tín cung cấp Raspberry Pi chính hãng, và các phụ kiện, board mạch mở rộng cho Raspberry Pi tại Việt Nam.

📍 Số 30F9 - Ngõ 104 Lê Thanh Nghị - Hai Bà Trưng - Hà Nội

☎ 02436.231.170

✉ smarttechvn.group@gmail.com

📞 086.262.8846 (Mr Thùy)

Message us

(<https://m.me/599911456>)

GetButton

HOTLINE TƯ VẤN TRỰC TIẾP

086.262.8846 (Mr Thùy) (tel:0962539275)

(Thời gian làm việc 8h - 17h30, thứ 2 tới thứ 7. Hỗ trợ Online ngoài giờ hành chính và chủ nhật.)

VỀ CHÚNG TÔI

[Giới thiệu \(https://pivietnam.com.vn/ve-chung-toi\)](https://pivietnam.com.vn/ve-chung-toi)

[Lịch sử hình thành \(https://pivietnam.com.vn/lich-su-hinh-thanh\)](https://pivietnam.com.vn/lich-su-hinh-thanh)

[Đội ngũ lãnh đạo \(https://pivietnam.com.vn/doi-ngu-lanh-dao\)](https://pivietnam.com.vn/doi-ngu-lanh-dao)

[Tuyển dụng \(https://pivietnam.com.vn/tuyen-dung-quy-i\)](https://pivietnam.com.vn/tuyen-dung-quy-i)

[Liên hệ \(https://pivietnam.com.vn/lien-he\)](https://pivietnam.com.vn/lien-he)

CHÍNH SÁCH

[Hướng dẫn mua hàng online \(https://pivietnam.com.vn/huong-dn-mua-hang-online-mlab-vn\)](https://pivietnam.com.vn/huong-dn-mua-hang-online-mlab-vn)

[Chính sách vận chuyển và giao nhận \(https://pivietnam.com.vn/chinh-sach-van-chuyen-va-giao-nhan-mlab-vn\)](https://pivietnam.com.vn/chinh-sach-van-chuyen-va-giao-nhan-mlab-vn)

[Thông tin chuyển khoản \(https://pivietnam.com.vn/thong-tin-chuyen-khoan-mlab-vn\)](https://pivietnam.com.vn/thong-tin-chuyen-khoan-mlab-vn)

[Hỗ trợ sau bán hàng \(https://pivietnam.com.vn/ho-tro-sau-ban-hang-mlab-vn\)](https://pivietnam.com.vn/ho-tro-sau-ban-hang-mlab-vn)

[Chính sách bảo hành \(https://pivietnam.com.vn/chinh-sach-bao-hanh-mlab-vn\)](https://pivietnam.com.vn/chinh-sach-bao-hanh-mlab-vn)

[Chính sách đổi trả, hoàn tiền \(https://pivietnam.com.vn/chinh-sach-doi-tra-hoan-tien-mlab-vn\)](https://pivietnam.com.vn/chinh-sach-doi-tra-hoan-tien-mlab-vn)

[Chính sách bảo mật thông tin \(https://pivietnam.com.vn/chinh-sach-bao-mat-thong-tin-mlab-vn\)](https://pivietnam.com.vn/chinh-sach-bao-mat-thong-tin-mlab-vn)

ĐĂNG KÝ NHẬN BẢN TIN

[Đăng ký](#)

FACEBOOK FANPAGE



086.262.8846 (Mr Thùy)

Công ty TNHH MLAB

Message us

<https://m.me/599911456>

GetButton

Số chứng nhận kinh doanh: 0106356768. Nơi cấp: Sở kế hoạch và đầu tư Thành Phố Hà Nội. Ngày cấp: 07/11/2013

Trụ sở : Số 30F9 - Ngõ 104 Lê Thanh Nghị - Hai Bà Trưng - Hà Nội

Email mua bán hàng : smarttechvn.group@gmail.com

Email hỗ trợ kỹ thuật : mlab.services.tech@gmail.com

Website : <https://pivietnam.com.vn/>

Số điện thoại : 02436.231.170 or 086.262.8846



(tel:0962539275)



086.262.8846 (Mr Thùy)

Message us

(<https://m.me/599911456>)

GetButton