**TRƯỜNG ĐẠI HỌC THĂNG LONG**

**BỘ MÔN TIN HỌC**

**---🙟🕮🙝---**



**BÁO CÁO BÀI TẬP NHÓM**

**MÔN HỌC: LẬP TRÌNH ỨNG DỤNG IOT**

**Đề tài: Smart Security Camera**

**Sinh viên thực hiện:** **Hoàng Gia Minh – A33624**

**Tống Duy Chiến – A32697**

**Võ Tiến Đạt – A33779**

**Nguyễn Đình Kiên – A34006**

**Hà Nội, 2/2022**

Mục lục

[1. Giới thiệu 3](#_Toc99024276)

[2. Mô tả Camera 3](#_Toc99024277)

[1. RASPBERRY PI 3 3](#_Toc99024278)

[2. Raspberry Pi Camera Module NoIR V2 5](#_Toc99024279)

[3. OpenCV 6](#_Toc99024280)

[3.1 OpenCV là gì ? 6](#_Toc99024281)

[3.2 Cách cài OpenCV trên raspberry pi 7](#_Toc99024282)

[4. Thuật toán 8](#_Toc99024283)

[4.1 Thiết lập camera 9](#_Toc99024284)

[4.2 Thiết lập gửi mail 10](#_Toc99024285)

[4.3 Thuật toán hoạt động 11](#_Toc99024286)

[5. Vai trò các thành viên 13](#_Toc99024287)

[6. Nguồn tham khảo 14](#_Toc99024288)

# Giới thiệu

Nhà thông minh ngày càng được quan tâm hơn so với các ứng dụng hiện tại của Internet vạn vật (IoT) do sự tiện lợi và tiện ích mà nó mang lại cho cuộc sống hàng ngày của chúng ta. Một trong những ứng dụng nổi trội đó là Hệ thống giám sát của ngôi nhà sử dụng camera để nhận diện chủ nhân, đồng thời gởi tin nhắn cho chủ nhân khi phát hiện người lạ xung quanh khu vực giám sát của ngôi nhà, được phát triển dựa trên ứng dụng IoT nhằm đáp ứng nhu cầu tự động, và cung cấp sự thuận lợi cho quá trình quản lí và điều khiển.

Trong bài báo cáo này, nhóm đề xuất xây dựng một camera thông minh thân thiện với người dùng, thông qua sử dụng Raspberry Pi 3 B+ và camera Raspberry Pi để làm Camera. Máy ảnh sử dụng OpenCV để nhận diện đối tượng và gửi thông báo cho chủ nhân bất cứ khi nào nó nhìn thấy người lạ xuất hiện. Nó cũng có thể hoạt động như một webcam, ta có thể theo dõi căn nhà từ xa khi ta đi vắng.

# Mô tả Camera

Camera IoT được kết nối mạng ghi nhận hình ảnh trong bán kính khu vực cho phép, dữ liệu hình ảnh được ghi lại gửi đến server để phân tích các điểm khuôn mặt thông qua Face Detection API bằng Python, Raspberry pi 3 và OpenCV, đem hình ảnh ghi nhận được đem so sánh với hệ thống khuôn mặt các thành viên trong ngôi nhà được lưu trữ trong database, nếu nhận diện giống với cái điểm khuôn mặt của một trong những thành viên trong ngôi nhà thì không gửi thông báo và tiếp tục ghi nhận hình ảnh, nếu server ghi nhận hình ảnh khác với các điểm trên thành viên trong nhà hệ thống sẽ gửi cảnh báo qua mail.



Hình .1 Smart Camera làm từ raspberry pi3

## RASPBERRY PI 3

RASPBERRY PI 3 Model B+ là một phiên bản nâng cấp của Raspberry Pi 3 Model B. Bản nâng cấp mang đến tốc độ nhanh hơn về mọi mặt. Cụ thể, điểm nâng cấp chính của Raspberry Pi 3 Model B+ là vi xử lý và kết nối mạng. Model B+ dùng vi xử lý Broadcom BCM2837B0 4 nhân 1.4GHz. Với các công việc đòi hỏi tốc độ mạng nhanh, Pi 3 Model B+ có thể đáp ứng với kết nối Wi-Fi 2 băng tần 2.4GHz và 5GHz (dual band), Ethernet gigabit (qua cổng USB 2.0) tốc độ lên đến 300Mbps, gấp 3 lần so với Pi 3 Model B. Thiết bị cũng hỗ trợ Bluetooth 4.2 và Bluetooth LE giúp kết nối tốt hơn với các thiết bị thông minh khác. Cuối cùng, Model B+ còn có Power over Ethernet (PoE) giúp cung cấp nguồn điện cho thiết bị thông qua dây cắm Ethernet nhưng phải thông qua một HAT mở rộng.

Ngoài những nâng cấp trên thì ngoại hình và kích thước Model B+ vẫn y hệt Model B nên hoàn toàn tương thích với mọi [case](https://raspberrypi.vn/danh-muc/vo-case-raspberry-pi) và [phụ kiện](https://raspberrypi.vn/danh-muc/phu-kien-raspberry-pi) trước đây dành cho Model B. Cấu hình chi tiết Raspberry Pi 3 Model B+:

* SoC: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1,4 GHz
* RAM: 1 GB LPDDR2 SDRAM
* Wi-Fi b/g/n/ac
* Bluetooth 4.2, BLE
* Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
* 40-pin GPIO
* HDMI
* 4 x cổng USB 2.0
* Khe cắm thẻ Micro SD
* Hỗ trợ Power-over-Ethernet (PoE)
* Cải thiện PXE network và USB mass-storage booting
* Tản nhiệt tốt hơn Model B



Hình .2 Raspberry Pi 3 Model B+

## Raspberry Pi Camera Module NoIR V2

Raspberry Pi Camera Module NoIR V2 mang trong mình mọi tính năng, công dụng của Camera Module thông thường, tuy nhiên có 1 điểm khác biệt là nó không sử dụng bộ lọc hồng ngoại (NoIR = No Infrared). Điều này có nghĩa là hình ảnh bạn chụp bằng ánh sáng ban ngày sẽ nhìn hơi mờ, nhưng nó mang lại cho bạn khả năng nhìn trong bóng tối với ánh sáng hồng ngoại. Nó là một sản phẩm hoàn hảo để quan sát vào ban đêm (như camera giám sát), chụp ảnh trong môi trường ánh sáng thấp (như hoàng hôn chẳng hạn).

Chiếc camera này tương thích với tất cả các phiên bản của Raspberry Pi. Nó có thể được truy cập thông qua các API MMAL và V4L, và có rất nhiều thư viện của bên thứ ba xây dựng cho nó, bao gồm cả các thư viện Picamera Python. Raspberry Pi Camera Module NoIR V2 là một bước nhảy vọt về chất lượng hình ảnh, màu sắc trung thực và hiệu suất ánh sáng thấp . Đặc biệt nó hỗ trợ video lên tới 1080P30, 720P60 và video mode VGA90, cũng như chế độ chụp hình. Dĩ nhiên, nó vẫn sử dụng đoạn cáp 15cm qua cổng CSI trên Raspberry Pi.

Các ứng dụng của Raspberry Pi Camera NoIR:

* Chụp ảnh hồng ngoại
* Chụp ảnh ánh sáng thấp
* Theo dõi cây cối trong vườn. Khi được sử dụng với đèn LED hồng ngoại, bạn có thể nhìn thấy trong bóng tối, vì vậy bạn có thể theo dõi sức khỏe thực vật hay thậm chí động vật ăn đêm
* Camera an ninh CCTV

Thông số kỹ thuật:

* Ống kính tiêu cự cố định
* Cảm biến độ phân giải 8 megapixel cho khả năng chụp ảnh kích thước 3280 x 2464
* Hỗ trợ video 1080p30, 720p60 và 640x480p90
* Kích thước 25mm x 23mm x 9mm
* Trọng lượng chỉ hơn 3g
* Kết nối với Raspberry Pi thông qua cáp ribbon đi kèm dài 15 cm
* Camera Module được hỗ trợ với phiên bản mới nhất của Raspbian

## OpenCV

## 3.1 OpenCV là gì ?

OpenCV là tên viết tắt của open source computer vision library – có thể được hiểu là một thư viện nguồn mở cho máy tính. Cụ thể hơn OpenCV là kho lưu trữ các mã nguồn mở được dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực.

OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các learning machine khác. Thư viện OpenCV được phát hành với giấy phép BDS. Do đó các dịch vụ nó cung cấp là hoàn toàn miễn phí và được hạn chế tối đa các rào cản thông thường. Cụ thể, bạn được phép sử dụng phần mềm này cho cả hoạt động thương mại lẫn phi thương mại. OpenCV sở hữu giao diện thiên thiện với mọi loại ngôn ngữ lập trình, ví dụ như C++, C, Python hay Java… Trong bài lab này, nhóm sử dụng Python. Ngoài ra, nó cũng dễ dàng tương thích với các hệ điều hành khác nhau, bao gồm từ Windows, Linux, Mac OS, iOS cho đến cả Android.

OpenCV có cấu trúc module, tức là nó bao gồm cả những thư viện liên kết tĩnh lẫn thư viện liên kết động. Nắm rõ các module của OpenCV sẽ giúp bạn đọc hoàn toàn thấu hiểu OpenCV là gì.

* Core functionality (core): Module này sở hữu cơ chế rất nhỏ gọn. Nó được dùng để định hình các cấu trúc của cơ sở dữ liệu cơ bản, bao gồm cả những mảng đa chiều. Ngoài ra nó còn xác định các chức năng của những module đi kèm khác nữa.
* Image Processing (imgproc): Đây là module được dùng cho quá trình xử lý hình ảnh. Nó cho phép người dùng thực hiện các hoạt động như lọc hình ảnh tuyến tính và phi tuyến, thực hiện phép biến hình, thay đổi không gian màu, xây dựng biểu đồ và rất nhiều thao tác khác liên quan.
* Video Analysis (video): Giống như tên gọi của nó, module này cho phép phân tích các video. Kết quả được trả về bao gồm các ước tính chuyển động, thực hiện tách nền và các phép toán theo dõi vật thể.
* Camera Calibration and 3D Reconstruction (calib3d): Module này cung cấp các thuật toán hình học đa chiều cơ bản và hiệu chuẩn máy ảnh single và stereo. Ngoài ra nó còn đưa ra các dự đoán kiểu dáng của đối tượng và sử dụng thuật toán thư tín âm thanh nổi cùng các yếu tố tái tạo 3D.
* 2D Features Framework (features2d): Module này giúp phát hiện các tính năng nổi trội của bộ nhận diện, bộ truy xuất thông số và thông số đối chọi.
* Ngoài ra còn có rất nhiều module khác với đa dạng tính năng, ví dụ như: FLANN, Google test wrapper…

## 3.2 Cách cài OpenCV trên raspberry pi

Bước 1: Cài đặt cài phần thiết yếu trên hệ điều hành

* sudo apt-get update
* sudo apt-get upgrade

Sau đó cài đặt một số công cụ dành cho nhà phát triển, bao gồm CMake , giúp chúng tôi định cấu hình quy trình xây dựng OpenCV, cùng các thư viện hỗ trợ khác:

* sudo apt-get install build-essential cmake pkg-config
* sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
* sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
* sudo apt-get install libxvidcore-dev libx264-dev
* sudo apt-get install libgtk2.0-dev
* sudo apt-get install libatlas-base-dev gfortran

Tiếp đến cài đặt python2 hoặc python3 tùy nhu cầu, và OpenCV

* sudo apt-get install python2.7-dev python3-dev
* sudo pip3 install opencv-python

Bước 2: Cài đặt môi trường ảo để thử nghiệm

* sudo pip install virtualenv virtualenvwrapper
* sudo rm -rf ~/.cache/pip

Bạn có thể cấu hình môi trường ảo qua lệnh:

* nano ~/.profile

Tiếp đến, hêm các tập lệnh sau:

* export WORKON\_HOME=$HOME/.virtualenvs
* source /usr/local/bin/virtualenvwrapper.sh

Để tạo môi trường, ta gõ lệnh:

* mkvirtualenv cv ( áp dụng cho python cũ hơn 3)
* mkvirtualenv cv -p python3 (áp dụng cho python3)

Tạn sẽ thấy (cv) trên thiết bị đầu cuối. Nếu ta đóng thiết bị đầu cuối hoặc gọi một thiết bị đầu cuối mới, bạn sẽ phải kích hoạt lại môi trường ảo Python của mình:

* source ~/.profile</strong>
* workon cv

# Thuật toán

Về chức năng cơ bản của Smart Camera, ta có thể chia làm 3 phần:

* Camera ghi hình, quay như một web cam
* Xử lí hình ảnh và phát hiện người lạ
* Gửi cảnh báo tới email người dùng

## 4.1 Thiết lập camera

class VideoCamera(object):

# Khai báo giá trị đầu vào cho camera. Camera lật góc quay là false. Tiến hành

# quay khi tự khở tạo đối tượng, với thời gian ghi delay là 2s.

    def \_\_init\_\_(self, flip = False):

        self.vs = PiVideoStream().start()

        self.flip = flip

        time.sleep(2.0)

# Hàm chức năng tắt cam

    def \_\_del\_\_(self):

        self.vs.stop()

# Nếu như camera bị lật ngược, hàm dưới sẽ có chức năng xoay ngược lại khung hình # để trở về góc quay đúng.

    def flip\_if\_needed(self, frame):

        if self.flip:

            return np.flip(frame, 0)

        return frame

# Lấy một hình ảnh từ camera, vì openCV chỉ xử lí ảnh JPEG, trong khi camera mặc # định chụp ảnh thô. Cho nên ta phải mã hóa nó thành JPEG.

    def get\_frame(self):

        frame = self.flip\_if\_needed(self.vs.read())

        ret, jpeg = cv2.imencode('.jpg', frame)

        return jpeg.tobytes()

# Hàm xác nhận đối tượng, chuyển đổi sang ảnh xám, nếu khung len nhận diện lớn hơn 0 thì đối tượng đã được xác nhận.

    def get\_object(self, classifier):

        found\_objects = False

        frame = self.flip\_if\_needed(self.vs.read()).copy()

        gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

        objects = classifier.detectMultiScale(

            gray,

            scaleFactor=1.1,

            minNeighbors=5,

            minSize=(30, 30),

            flags=cv2.CASCADE\_SCALE\_IMAGE

        )

        if len(objects) > 0:

            found\_objects = True

        # Draw a rectangle around the objects

        for (x, y, w, h) in objects:

            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

        ret, jpeg = cv2.imencode('.jpg', frame)

        return (jpeg.tobytes(), found\_objects)

## 4.2 Thiết lập gửi mail

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.image import MIMEImage

# Khai báo email cho con camera

fromEmail = 'email@gmail.com'

fromEmailPassword = 'password'

# Email này sẽ nhận cảnh cáo từ gmail của camera

toEmail = 'email2@gmail.com'

# Hàm gửi mail

def sendEmail(image):

    msgRoot = MIMEMultipart('related')

    msgRoot['Subject'] = 'Security Update'

    msgRoot['From'] = fromEmail

    msgRoot['To'] = toEmail

    msgRoot.preamble = 'Raspberry pi security camera update'

    msgAlternative = MIMEMultipart('alternative')

    msgRoot.attach(msgAlternative)

    msgText = MIMEText('Smart security cam found object')

    msgAlternative.attach(msgText)

    msgText = MIMEText('<img src="cid:image1">', 'html')

    msgAlternative.attach(msgText)

    msgImage = MIMEImage(image)

    msgImage.add\_header('Content-ID', '<image1>')

    msgRoot.attach(msgImage)

    smtp = smtplib.SMTP('smtp.gmail.com', 587)

    smtp.starttls()

    smtp.login(fromEmail, fromEmailPassword)

    smtp.sendmail(fromEmail, toEmail, msgRoot.as\_string())

    smtp.quit()

## 4.3 Thuật toán hoạt động

import cv2

import sys

from mail import sendEmail

from flask import Flask, render\_template, Response

from camera import VideoCamera

from flask\_basicauth import BasicAuth

import time

import threading

email\_update\_interval = 600 # sends an email only once in this time interval

video\_camera = VideoCamera(flip=True) # creates a camera object, flip vertically

object\_classifier = cv2.CascadeClassifier("models/fullbody\_recognition\_model.xml") # an opencv classifier

# App Globals (do not edit)

app = Flask(\_\_name\_\_)

app.config['BASIC\_AUTH\_USERNAME'] = 'CHANGE\_ME\_USERNAME'

app.config['BASIC\_AUTH\_PASSWORD'] = 'CHANGE\_ME\_PLEASE'

app.config['BASIC\_AUTH\_FORCE'] = True

basic\_auth = BasicAuth(app)

last\_epoch = 0

def check\_for\_objects():

    global last\_epoch

    while True:

        try:

            frame, found\_obj = video\_camera.get\_object(object\_classifier)

            if found\_obj and (time.time() - last\_epoch) > email\_update\_interval:

                last\_epoch = time.time()

                print ("Sending email...")

                sendEmail(frame)

                print ("done!")

        except:

            print ("Error sending email: ", sys.exc\_info()[0])

@app.route('/')

@basic\_auth.required

def index():

    return render\_template('index.html')

def gen(camera):

    while True:

        frame = camera.get\_frame()

        yield (b'--frame\r\n'

               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video\_feed')

def video\_feed():

    return Response(gen(video\_camera),

                    mimetype='multipart/x-mixed-replace; boundary=frame')

if \_\_name\_\_ == '\_\_main\_\_':

    t = threading.Thread(target=check\_for\_objects, args=())

    t.daemon = True

    t.start()

    app.run(host='0.0.0.0', debug=False)

Camera sẽ khởi chạy, nếu phát hiện ra đối tượng, chương trình sẽ chạy tiến trình check\_for\_object để lấy ảnh đối tượng, kèm theo thời gian ghi nhận được.Tiến trình này sẽ xảy ra song song trong lúc camera ghi hình thông qua một luồng gọi bởi hàm thread. Hình ảnh sẽ được xử lí theo mẫu có sẵn ( Toàn thân, nửa trên, hoặc khuôn mặt ) ở object\_classifier.

Chế độ webcam sẽ được chạy trên trang <pi:ip>:5000. Trang webcam có xác thực riêng tư, ta sẽ thiết lập tài khoản và mật khẩu qua các hàm app\_config.

# Vai trò các thành viên

* + Võ Tiến Đạt + Nguyễn Đình Kiên: Kiểm thử phần cứng, thiết kế camera, và powerpoint
  + Hoàng Gia Minh + Tống Duy Chiến: Tìm hiểu thuật toán với OpenCV, tổng hợp word

# Nguồn tham khảo

Code: <https://github.com/HackerShackOfficial/Smart-Security-Camera>

OpenCV: <https://www.youtube.com/watch?v=fB2Pan42sYc&list=PLZPCoTKpEddAay-lItE-pn27uNNrApORH&index=4>

Đề tài : <https://www.youtube.com/watch?v=Y2QFu-tTvTI&list=LL&index=6>