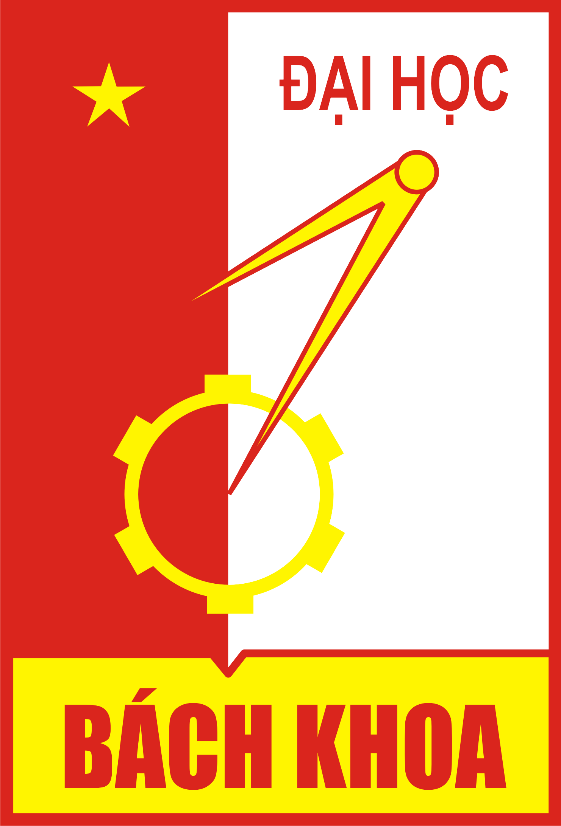
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────



BÁO CÁO MÔN: PROJECT 1

**Tìm hiểu sử dụng thư viện OpenCV   
ứng dụng vào nhận diện vật thể qua màu sắc**

**điều khiển slide**

Giáo viên hướng dẫn : **Nguyễn Đức Tiến**

Sinh viên thực hiện : **Nguyễn Văn Diện**

Lớp : **CNTT 1.1 – K60**

***Hà Nội, tháng 12 năm 2017***

**Mục lục**

[CHƯƠNG 1. Tổng quan đề tài 4](#_Toc501060922)

[1.1. Mô tả yêu cầu bài toán 4](#_Toc501060923)

[1.2. Cài đặt môi trường 4](#_Toc501060924)

[1.2.1. Cài đặt Python 4](#_Toc501060925)

[1.2.2. Cài đặt thư viện OpenCV 4](#_Toc501060926)

[CHƯƠNG 2. Xây dựng chương trình 5](#_Toc501060927)

[2.1. Lấy video từ camera và hiển thị lên màn hình 5](#_Toc501060928)

[2.2. Lựa chọn 1 vùng trên video và hiển trị lên 1 cửa sổ mới 5](#_Toc501060929)

[2.3. Nhận diện dải màu của vùng được chọn 6](#_Toc501060930)

[2.4. Tìm kiếm vùng có màu được chọn trong video 7](#_Toc501060931)

[2.5. Theo dõi vị trí khi vùng màu di chuyển trong khung video và vẽ lại vị trí mà nó đi qua 7](#_Toc501060932)

[2.6. Xử lý thuật toán áp dụng chuyển động vào chuyển tiếp slide 8](#_Toc501060933)

[CHƯƠNG 3. KIỂM THỬ CHƯƠNG TRÌNH 9](#_Toc501060934)

LỜI NÓI ĐẦU

Ngày nay việc phân tích và sử lý hình ảnh đem lại những ứng dụng vô cùng to lớn trong cuộc sống con người. Qua tìm hiểu em đã biết đến OpenCV (Open Source Computer Vision Library) – một bộ công cụ phần mềm để xử lý hình ảnh và video thời gian thực. Trong khuôn khổ Project1, em quyết định lựa chọn đề tài “*Tìm hiểu sử dụng thư viện opencv ứng dụng vào nhận diện vật thể qua màu sắc điều khiển slide*”. Em muốn tạo ra một chương trình có thể giúp thầy cô, các bạn học hay mọi người có thể dễ dàng, thoải mái thực hiện thuyết trình bằng slide. Đề tài này chỉ là một phần nghiên cứu nhỏ nhưng em mong muốn sẽ phát triển và mở rộng hơn trong tương lai.  
Em cảm ơn thầy Nguyễn Đức Tiến đã giúp đỡ và chỉ bảo em rất nhiều trong quá trình nghiên cứu và thực hiện đề tài này.

# Tổng quan đề tài

Em đã tạo ra một ứng dụng nhỏ, chạy trực tiếp trên Windowns gửi các ký tự giúp ’back’ và ’next’ slide Powerpoint. Ứng dụng được viết trên nền tảng OpenCV 3.3.0, Python 2.7, sử dụng PyCharm

## Mô tả yêu cầu bài toán

* Tìm hiểu, cài đặt thử nghiệm sử dụng thư viện OpenCV
* Nhận diện màu sắc của vật thể
* Theo dõi chuyện động của vật thể, vẽ lại đường đi. Ứng dụng theo dõi chuyển động của vật thể đó vào điều khiển Next và Back slide.

## Cài đặt môi trường

### Cài đặt Python

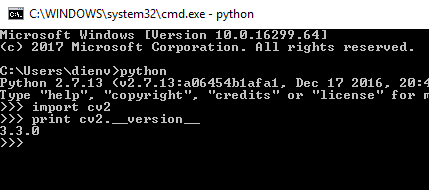
Phiên bản em dùng là 2.7.13. Có thể cài đặt các phiên bản khác mới hơn, song một vài câu lệnh có thể bị thay đổi trên phiên bản Python 2.\* và 3.\*

Link download Python: <https://www.python.org/downloads/>

### Cài đặt thư viện OpenCV

Link hướng dẫn: <https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_setup/py_table_of_contents_setup/py_table_of_contents_setup.html#py-table-of-content-setup>

Kiểm tra phiên bản:



# Xây dựng chương trình

Sau đây em sẽ trình bày từng bước để hoàn thiện nên chương trình

## Lấy video từ camera và hiển thị lên màn hình

import cv2  
  
camera = cv2.VideoCapture(0)  
  
cv2.namedWindow("image")  
  
while True:  
 # grab the current frame  
 (grabbed, frame) = camera.read()  
   
 # show the frame to our screen  
 cv2.imshow("image", frame)  
  
 key = cv2.waitKey(1) & 0xFF  
 # if the 'q' key is pressed, stop the loop  
 if key == ord("q"):  
 break  
   
# cleanup the camera and close any open windows  
camera.release()  
cv2.destroyAllWindows()

## Lựa chọn 1 vùng trên video và hiển trị lên 1 cửa sổ mới

* Khởi tạo các biến

x\_start, y\_start, x\_end, y\_end = 0, 0, 0, 0  
cropping = False  
getROI = False  
refPt = []

refPt : danh sách các biến x,y xác định khung của hình chữ nhật mà ta sẽ cắt từ video

cropping: (boolen) thể hiện xem có hay không việc cắt hình

getROI: (boolen) đã có vùng được chọn hay chưa

* Hàm click\_and\_crop()

def click\_and\_crop(event, x, y, flags, param):  
 # grab references to the global variables  
 global x\_start, y\_start, x\_end, y\_end, cropping, getROI  
  
 # if the left mouse button was clicked, record the starting  
 # (x, y) coordinates and indicate that cropping is being  
 # performed  
 if event == cv2.EVENT\_LBUTTONDOWN:  
 x\_start, y\_start, x\_end, y\_end = x, y, x, y  
 cropping = True  
  
 elif event == cv2.EVENT\_MOUSEMOVE:  
 if cropping == True:  
 x\_end, y\_end = x, y  
  
 # check to see if the left mouse button was released  
 elif event == cv2.EVENT\_LBUTTONUP:  
 # record the ending (x, y) coordinates and indicate that  
 # the cropping operation is finished  
 x\_end, y\_end = x, y  
 cropping = False  
 getROI = True

* Hiển thị vùng đc chọn lên cửa sổ mới

refPt = [(x\_start, y\_start), (x\_end, y\_end)]  
  
roi = frame[refPt[0][1]:refPt[1][1], refPt[0][0]:refPt[1][0]]  
cv2.imshow("ROI", roi)

* Xử lý khi đã cắt được vùng mong muốn thì xóa bỏ khung cắt trên video

if not getROI:  
  
 while True:  
 # grab the current frame  
 (grabbed, frame) = camera.read()  
  
 if not grabbed:  
 break  
  
 if not cropping and not getROI:  
 cv2.imshow("image", frame)  
  
 elif cropping and not getROI:  
 cv2.rectangle(frame, (x\_start, y\_start), (x\_end, y\_end), (0, 255, 0), 2)  
 cv2.imshow("image", frame)  
  
 elif not cropping and getROI:  
 cv2.rectangle(frame, (x\_start, y\_start), (x\_end, y\_end), (0, 255, 0), 2)  
 cv2.imshow("image", frame)  
 break  
  
 key = cv2.waitKey(1) & 0xFF  
 # if the 'q' key is pressed, stop the loop  
 if key == ord("q"):  
 noROI = True  
 break  
  
 # if there are two reference points, then crop the region of interest  
 # from teh image and display it  
 refPt = [(x\_start, y\_start), (x\_end, y\_end)]  
  
 roi = frame[refPt[0][1]:refPt[1][1], refPt[0][0]:refPt[1][0]]  
 cv2.imshow("ROI", roi)

## Nhận diện dải màu của vùng được chọn

Cần import thư viện numpy đêt thực hiện xử lý với mảng

import numpy as np

hsvRoi = cv2.cvtColor(roi, cv2.COLOR\_BGR2HSV)  
print('min H = {}, min S = {}, min V = {}; max H = {}, max S = {}, max V = {}'.format(hsvRoi[:, :, 0].min(),  
 hsvRoi[:, :, 1].min(),  
 hsvRoi[:, :, 2].min(),  
 hsvRoi[:, :, 0].max(),  
 hsvRoi[:, :, 1].max(),  
 hsvRoi[:, :, 2].max()))  
  
lower = np.array([hsvRoi[:, :, 0].min(), hsvRoi[:, :, 1].min(), hsvRoi[:, :, 2].min()])  
upper = np.array([hsvRoi[:, :, 0].max(), hsvRoi[:, :, 1].max(), hsvRoi[:, :, 2].max()])

## Tìm kiếm vùng có màu được chọn trong video

* + Làm mờ video

blurred = cv2.GaussianBlur(frame, (11, 11), 0)  
hsv = cv2.cvtColor(blurred, cv2.COLOR\_BGR2HSV)

Mục đích: Loại bỏ nhiễu hạt trên frame video

* + Thực hiện một loạt các phương pháp xử lý erosion -> dilation -> erosion đối với vùng hình ảnh nằm trong giới hạn màu nhằm loại bỏ các điểm mờ

kernel = np.ones((9, 9), np.uint8)

mask = cv2.inRange(hsv, lower, upper)  
mask = cv2.morphologyEx(mask, cv2.MORPH\_OPEN, kernel)  
mask = cv2.morphologyEx(mask, cv2.MORPH\_CLOSE, kernel)

* + Tìm contours của vùng màu đã nhận

cnts = cv2.findContours(mask.copy(), cv2.RETR\_EXTERNAL,  
 cv2.CHAIN\_APPROX\_SIMPLE)[-2]

Contours là đường nối các điểm liên tục (theo biên) có cùng màu

* + Tìm tâm và vẽ đường tròn quanh tâm với bán kính tối thiểu bao quanh vùng đó

if len(cnts) > 0:  
 # find the largest contour in the mask, then use  
 # it to compute the minimum enclosing circle and  
 # centroid  
 c = max(cnts, key=cv2.contourArea)  
 ((x, y), radius) = cv2.minEnclosingCircle(c)  
 M = cv2.moments(c)  
 center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))  
  
 # only proceed if the radius meets a minimum size. Correct this value for your obect's size  
 if radius > 0.5:  
 # draw the circle and centroid on the frame,  
 # then update the list of tracked points  
 cv2.circle(frame, (int(x), int(y)), int(radius), (0, 255, 0), 2)  
 # cv2.putText(frame, 'center: {}, {}'.format(int(x), int(y)), (int(x - radius), int(y - radius)),  
 # cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, (0, 0, 255), 2)

## Theo dõi vị trí khi vùng màu di chuyển trong khung video và vẽ lại vị trí mà nó đi qua

* Khởi tạo 1 danh sách lưu lại vị trí tâm của vùng màu, cần import thư viện collections

from collections import deque

MAX\_LEN = 10  
pts = deque(maxlen=MAX\_LEN)

* Update danh sách bằng cách thêm các vị trí tâm của vòng tròn

update the points queue  
pts.appendleft(center)

* Vẽ các điểm trên frame video

for i in xrange(1, len(pts)):  
 # if either of the tracked points are None, ignore  
 # them  
 if pts[i - 1] is None or pts[i] is None:  
 continue  
  
 # otherwise, compute the thickness of the line and  
 # draw the connecting lines  
 thickness = int(np.sqrt(MAX\_LEN / float(i + 1)) \* 2.5)  
 cv2.line(frame, pts[i - 1], pts[i], (0, 0, 255), thickness)

## Xử lý thuật toán áp dụng chuyển động vào chuyển tiếp slide

Sẽ sử dụng hai vùng biên hai bên, khi vật đi qua đó thì sẽ gửi ký tự tương ứng (qua trái là “back - dấu mũi tên qua trái” và qua phải là “next - dấu mũi tên qua phải”.

* Tạo thêm 1 list để lưu tọa độ x on frame

arr = deque(maxlen=MAX\_LEN)

* Update

arr.appendleft(int(x))

* Hàm kiểm tra vị trí

def checkPosition(x):  
 if x < 100 :  
 return 1  
 if x > 500 :  
 return 2  
 else :  
 return 0

* Hàm sendKey

import win32com.client as comclt

if (len(arr) == 10):  
  
 if checkPosition(arr[0]) == 1 and old\_position != 1:  
   
 wsh.SendKeys("{down}") # seaand the keys you want  
  
 if checkPosition(arr[0]) == 2 and old\_position != 2:  
   
 wsh.SendKeys("{up}") # seaand the keys you wantaa  
 old\_position = checkPosition(arr[0])

# KIỂM THỬ CHƯƠNG TRÌNH

PHỤ LỤC

Link video demo: <https://www.youtube.com/watch?v=Xp6KB37mBk4>

Link code full: <https://github.com/dienvx1997bn/Project1>

**TÀI LIỆU THAM KHẢO**

**OpenCV 3.0.0-dev documentation:** [**https://docs.opencv.org/3.0-beta/doc/py\_tutorials/py\_tutorials.html**](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html)

**Ball Tracking with OpenCV PyImageSearch:** [**https://www.pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/**](https://www.pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/)

**Nikgens - select-color-and-follow.py :** [**https://gist.github.com/nikgens/03447f19bea0216cf64243092795806d**](https://gist.github.com/nikgens/03447f19bea0216cf64243092795806d)