**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**BÁO CÁO ĐỒ ÁN MÔN**

**NHẬP MÔN THỊ GIÁC MÁY TÍNH**

*Đề tài:*

NHẬN DIỆN TIỀN VỚI MÁY TÍNH NHÚNG RASPBERRY PI

Giảng viên hướng dẫn:

**Nguyễn Vĩnh Tiệp**

Nhóm sinh viên thực hiện:

**Tống Anh Quân(\*) – 16520985 Nguyễn Lương Duy Khánh – 16520584**

**TP. Hồ Chí Minh, 01/2020**

Mục lục

[**1.** **Giới thiệu đề tài** 4](#_Toc29328072)

[**2.** **Sơ đồ hoạt động của tệp tin rút trích đặc trưng và nhận diện** 4](#_Toc29328073)

[**3.** **Cách vận hành hệ thống** 5](#_Toc29328076)

[**4.** **Phần cứng** 6](#_Toc29328077)

[a) Máy tính nhúng Raspberry Pi 3 B+ 6](#_Toc29328078)

[b) Màn hình LCD16x02 (I2C support) 7](#_Toc29328080)

[c) Camera Pi 8](#_Toc29328083)

[d) Các phần cứng hỗ trợ khác 8](#_Toc29328085)

[**5.** **Phần mềm sử dụng** 8](#_Toc29328086)

[**6.** **Sơ đồ kết nối phần cứng** 9](#_Toc29328087)

[**7.** **Các hiển thị trên màn hình LCD khi hệ thống nhận diện** 9](#_Toc29328089)

[**8.** **Một số hình ảnh về sản phẩm thực tế** 10](#_Toc29328091)

[**9.** **Một số hình ảnh mẫu về tiền để rút trích đặc trưng** 11](#_Toc29328094)

[**10.** **So sánh thời gian nhận diện giữa các thuật toán và thời gian rút trích đặc trưng** 12](#_Toc29328096)

[**11.** **Cải tiến sau khi được thầy nhận xét ở seminar trên lớp** 13](#_Toc29328098)

[a. Khi thực hiện seminar: 13](#_Toc29328099)

[b. Khi báo cáo đồ án 13](#_Toc29328100)

Mục lục hình ảnh

[*HÌnh 2.1. Sơ đồ hoạt động của tệp tin rút trích đặc trưng* 4](#_Toc29326569)

[*HÌnh 2.2. Sơ đồ hoạt động của tệp tin rút nhận diện* 5](#_Toc29326570)

[*Hình 4.a. Máy tính nhúng Raspberry Pi 3B+* 6](#_Toc29326574)

[*Hình 4.b. Mặt trước và mặt sau module 16x02* 7](#_Toc29326577)

[*Hình 4.c. Camera Pi V1* 8](#_Toc29326579)

[*Hình 6. Sơ đồ kết nối phần cứng* 9](#_Toc29326583)

[*Hình 7. Các hiện thị trên LCD khi hệ thống nhận diện* 9](#_Toc29326585)

[*Hình 8.1 Hệ thống phần cứng bên trong hộp* 10](#_Toc29326587)

[*Hình 8.2. Dáng vẻ bên ngoài và nơi đưa tiền vào 11*](#_Toc29326588)

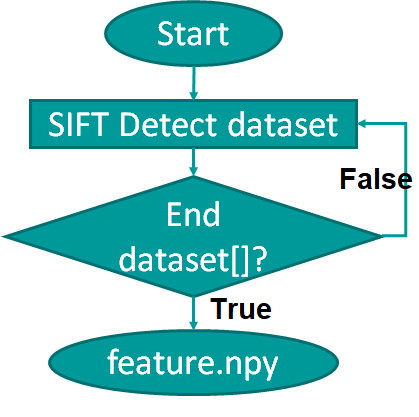
[*Hình 9. Các hình mẫu về tiền để rút trích đặc trưng 12*](#_Toc29326590)

[*Hình 10. Biểu đồ so sánh thời gian nhận diện và thời gian rút trích đặc trưng* 12](#_Toc29326592)

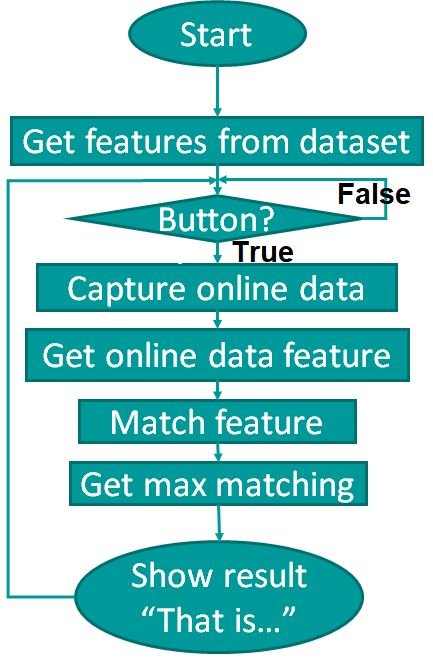
1. **Giới thiệu đề tài**

Đề tài nhóm chúng em làm về một thiết bị có khả năng nhận diện tiền và hiển thị lên màn hình LCD thông tin tiền đã nhận diện, board mạch chính mà nhóm đang sử dụng là máy tính nhúng Raspberry Pi 3B+, đề tài sử dụng thuật toán SIFT/SURF/AKAZE làm phương pháp chính để trích xuất các đặc trưng từ hình ảnh và so sánh. Không chỉ vậy, chỉ cần có trước các dữ liệu dataset thì với đề tài này, mọi vật thể đều có thể được nhận diện một cách tương đối chính xác và dễ dàng so với các phương pháp dùng Deep Learning hay Machine Learning,.. bên cạnh đó thì thời gian đáp ứng khá ổn cho những ứng dụng phân loại hay nhận diện cỡ nhỏ với thời gian trong khoảng 5s cho một vật thể nhận diện tuỳ vào phương pháp sử dụng.

1. **Sơ đồ hoạt động của tệp tin rút trích đặc trưng và nhận diện**

****

## *HÌnh 2.1. Sơ đồ hoạt động của tệp tin rút trích đặc trưng*

****

## *HÌnh 2.2. Sơ đồ hoạt động của tệp tin rút nhận diện*

1. **Cách vận hành hệ thống**

- Bước 1 : Truy cập vào thư mục DetectMoney và chạy tệp ‘TestCamerabyStreamVideo.py’ với câu lệnh python3 để kiểm tra xem camera đã hoạt động hay chưa

- Bước 2 : Ta đưa tờ tiền vào khi vẫn đang chạy tệp ở Bước 1 để chỉnh chon ngay ngắn

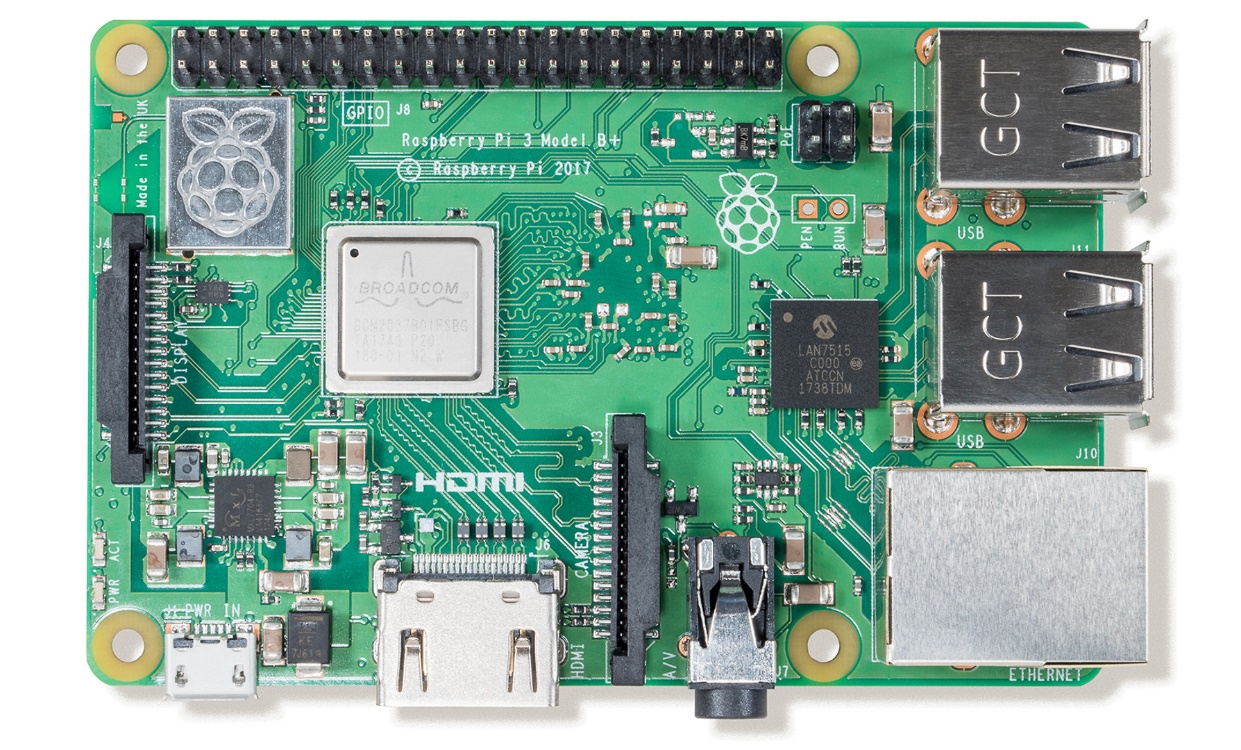
- Bước 3 : Ta mở tệp ‘TakeAPictureToGetDataSet.py’ và thay đổi lại tên đường dẫn ứng với từng tờ tiền trước khi chạy tệp này với câu lệnh python3

- Bước 4 : Ta mở tệp ‘CreateFeatureFile.py’ và chỉnh sửa các đường dẫn tương ứng với các đường dẫn đã được tạo ra ở Bước 2. Sau đó chạy tệp này với câu lệnh python3

- Bước 5 : Ta mở tệp ‘Detect.py’ và chỉnh sửa các đường dẫn tương ứng với các đường dẫn đã được tạo ra ở Bước 2. Sau đó chọn thuật toán mình muốn chạy hoặc không làm gì cả (Hệ thống sẽ dùng thuật toán mặc định là AKAZE, khi thay đổi thuậ toán thì ta phải thay đổi thuật toán trong tệp trích đặc trưng ‘CreateFeatureFile.py’ tương ứng, riêng với SUFT và SIFT thì có thể dùng chung ).

**-** Sau khi đưa tiền vào và chạy file Detect.py, ta có thể xem lại hình ảnh của tờ tiền mà ta đã bỏ vào bằng cách mở file ‘userimg.img’.

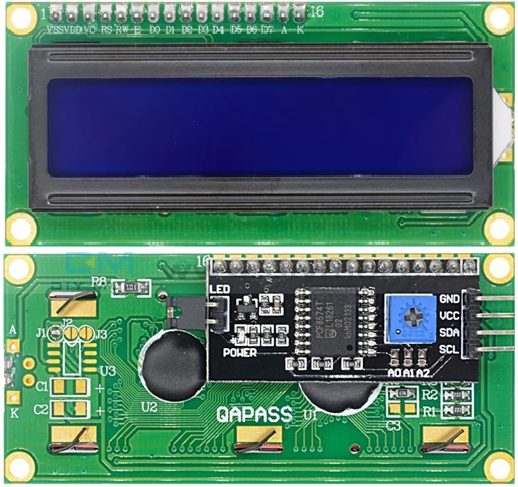
1. **Phần cứng** 
   1. Máy tính nhúng Raspberry Pi 3 B+



## *Hình 4.a. Máy tính nhúng Raspberry Pi 3B+*

Dưới đây là thông số kỹ thuật của máy tính nhúng Raspberry Pi 3B+:

* Vi xử lý: Broadcom BCM2837B0, quad-core A53 (ARMv8) 64-bit SoC @1.4GHz
* RAM: 1GB LPDDR2 SDRAM
* Kết nối: 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet over USB 2.0 (Tối đa 300Mbps).
* Cổng USB: 4 x 2.0
* Mở rộng: 40-pin GPIO
* Video và âm thanh: 1 cổng full-sized HDMI, Cổng MIPI DSI Display, cổng MIPI CSI Camera, cổng stereo output và composite video 4 chân.
* Multimedia: H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
* Lưu trữ: MicroSD
* Nguồn điện sử dụng: 5V/2.5A DC cổng microUSB, 5V DC trên chân GPIO, Power over Ethernet (PoE) (yêu cầu thêm PoE HAT).
  1. Màn hình LCD16x02 (I2C support)



## *Hình 4.b. Mặt trước và mặt sau module 16x02*

Dưới đây là thông số kỹ thuật của LCD 16x02 (I2C support)

* Địa chỉ của module I2C trải dài từ 0x20-0x3F
* Đèn nền trắng và phông màu xanh
* Nguồn điện hoạt động ổn định: 5V
* Điều chỉnh độ tương phản thông qua biến trở trên module I2C

I2C là module màu đen được gắn thêm vào LCD như hình trên, module này giúp cho các phần cứng giao tiếp với LCD cần ít dây hơn vì module hỗ trợ chuẩn giao tiếp I2C (2 dây tín hiệu và 2 dây nguồn) và khi không dùng module I2C số dây lên đến 14 dây. Việc này giúp cho người dùng dễ dàng lập trình hơn là dùng LCD đơn thuần

* 1. Camera Pi



## *Hình 4.c. Camera Pi V1*

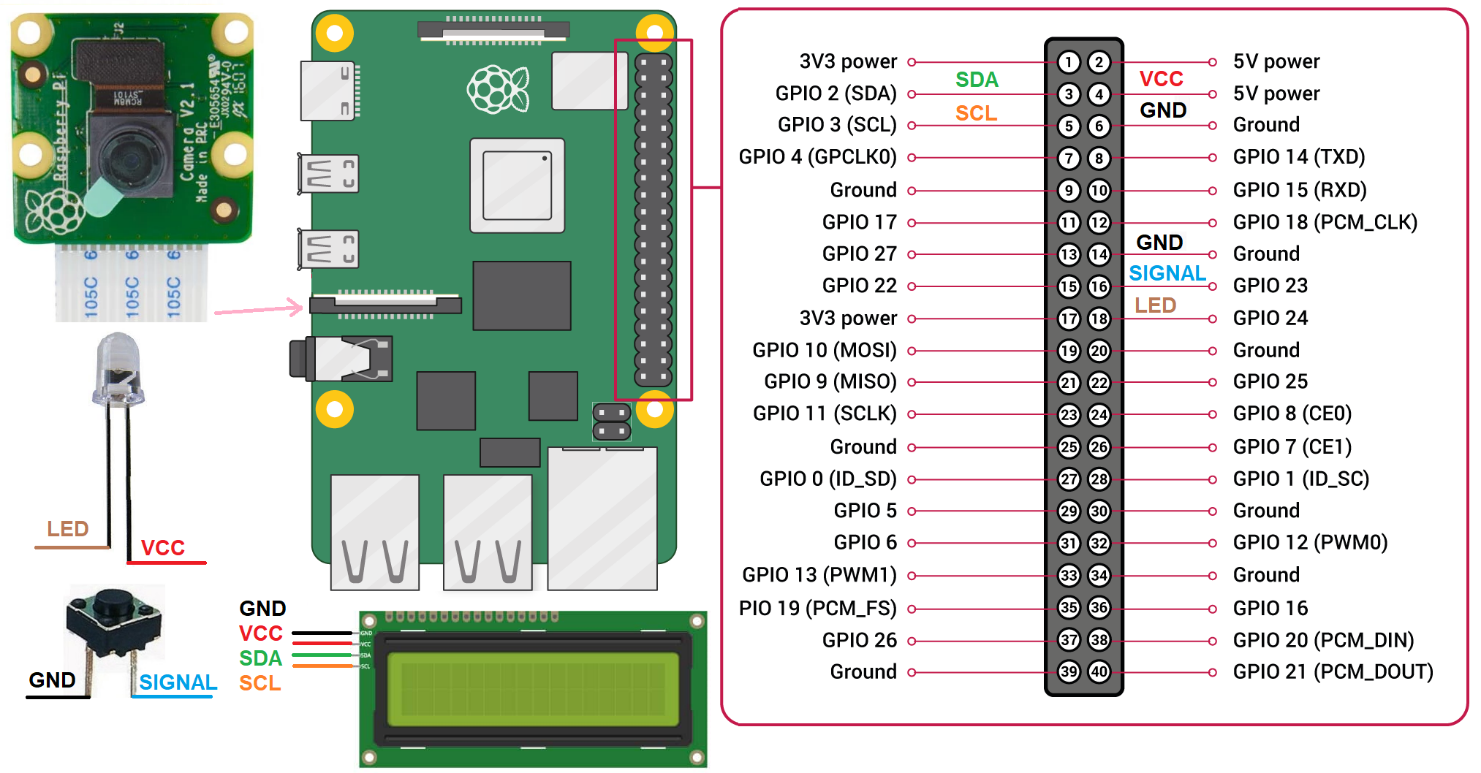
Dưới đây là thông số kỹ thuật của Camera Pi V1:

* Cảm biến: OV5647
* Độ phân giải: 5MP
* Độ phân giải hình: 2592x1944 pixel.
* Quay phim HD 1080P 30, 720P 60, VGA 640x480P 60.
* Lens: Fixed Focus.
  1. Các phần cứng hỗ trợ khác

Các phần cứng hỗ trợ khác như là : Dây, đèn led, nút nhấn,…

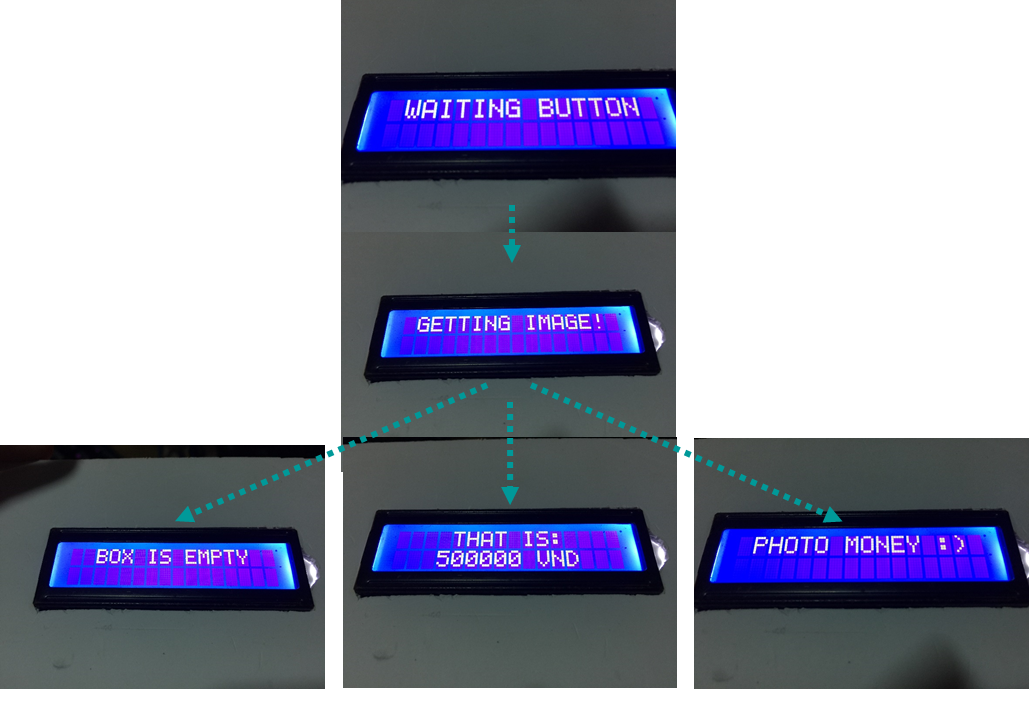
1. **Phần mềm sử dụng**

* OpenCV 3.4.3
* Python3
* Sift/Suft/Akaze library
* Numpy,Time,…
* Raspberry library (GPIO,I2C,Camera,..)

1. **Sơ đồ kết nối phần cứng**

## *Hình 6. Sơ đồ kết nối phần cứng*

1. **Các hiển thị trên màn hình LCD khi hệ thống nhận diện**

****

## *Hình 7. Các hiện thị trên LCD khi hệ thống nhận diện*

- Khi vừa chạy tệp tin ‘Detect.py’ . LCD sẽ hiển thị lên màn hình dòng chữ “WAITING BUTTON” để báo cho người dùng biết hệ thống đang đợi người dùng nhấn nút để bắt đầu nhận diện (Giả sử tiền đã được đưa vào trong hộp trước )

- Sau đó hệ thống sẽ hiển thị dòng chữ ‘GETTING IMAGE!’ nghĩa là hệ thống đang chụp lại tờ tiền đã được đưa vào trước đó để làm mẫu đem đi so sánh với các đặc trưng

- Sau khi chụp xong, hệ thống sẽ bắt đầu nhận diện với dògn chữ ‘DETECTING xxxxxxVND’ nhưng do ở đây hệ thống đã bị phân rã ra nên nhóm chưa có dịp chụp lại.

- Sau đó, ứng với mỗi trường hợp khác nhau sẽ cho ra hiển thị khác nhau:

* Khi hộp trống LCD sẽ hiển thị dòng chữ: ‘BOX IS EMPTY’
* Khi nhận diện được tờ tiền đã đưa vào LCD sẽ hiển thị dòng chữ: ‘ THAT IS: xxxxxVND’
* Khi tiền đưa vào là tiền dưới dạng photo LCD sẽ hiển thị dòng chữ: ‘ PHOTO MONEY’

1. **Một số hình ảnh về sản phẩm thực tế**

**Ảnh có chứa bàn, động cơ

Mô tả được tạo tự động**

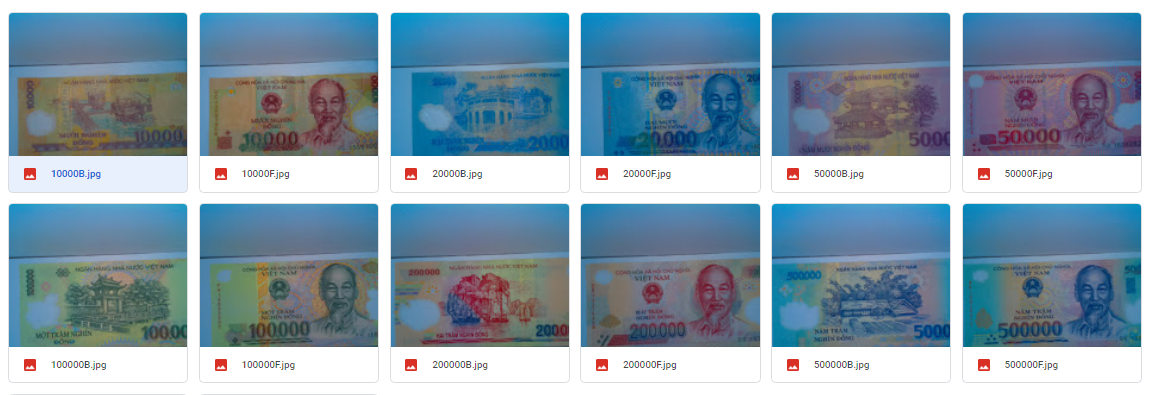
## *Hình 8.1 Hệ thống phần cứng bên trong hộp*

Ảnh có chứa máy tính

Mô tả được tạo tự động

## *Hình 8.2. Dáng vẻ bên ngoài và nơi đưa tiền vào*

1. **Một số hình ảnh mẫu về tiền để rút trích đặc trưng**

****

## *Hình 9. Các hình mẫu về tiền để rút trích đặc trưng*

1. **So sánh thời gian nhận diện giữa các thuật toán và thời gian rút trích đặc trưng**

## *Hình 10. Biểu đồ so sánh thời gian nhận diện và thời gian rút trích đặc trưng*

- Với thuật toán Bf-Matcher: Thuật toán SIFT nhận diện nhanh hơn với thời gian 2s trong khi SURF thực hiện hết 7s

- Với thuật toán Flann-Matcher: Thuật toán SIFT nhận diện nhanh hơn với thời gian 2.3s trong khi SURF thực hiện hết 21s

- Riêng với thuật toán Akaze-Matcher: Chỉ có thuật toán Akaze mới có thể xài được và thuật toán này chỉ mất 1.8s

- Về thời gian rút trích đặc trưng thì thuật toán SUFT chỉ mất 20s , thuật toán Akaze mất 26s và thời gian chiếm nhiều thời gian nhất đó là SIFT với 60s

* Kết luận: Akaze nhận diện nhanh nhất và có thời gian rút trích tương đối ổn trong khi SURF có thời gian rút trích là thấp nhất nhưng thời gian để nhận diện ra vật thể lại rất lâu. Và cuối cùng là SIFT, SIFT mất khá nhiều chi phí và thời gian cho việc rút trích đặc trưng nhưng thời gian xử lý nhận diện cũng rất nhanh. Có thể xem như là tương đương với thuật toán Akaze.

1. **Cải tiến sau khi được thầy nhận xét ở seminar trên lớp**
2. Khi thực hiện seminar:

Khi thực hiện seminar, nhóm chưa biết cách rút trích các đặc trưng của hình ảnh để lưu vào một tệp tin bên ngoài để tái sử dụng nhiều lần, làm cho hệ thống tốn khá nhiều thời gian để nhận diện hình ảnh vì đa số thời gian là để rút trích các đặc trưng.

1. Khi báo cáo đồ án

Sau khi đã được thầy nhận xét và gợi ý thì nhóm đã thực hiện việc tạo một mã nguồn riêng để rút trích hình ảnh một lần duy nhất (nếu có thay đổi dataset cũng như vật thể thì mới cần rút trích lại) sau khi rút trích thì tất cả dữ liệu sẽ được lưu vào một tệp có đuôi .npy. Do đó, khi chạy tệp nhận diện, hệ thống chỉ cần gọi tệp rút trích vào, sau đó lưu tất cả dữ liệu vào một mảng và khi nhận diện thì chỉ cần trỏ đến và só sánh. Việc này giúp cho tối ưu đáng kể việc nhận diện, thời gian nhận diện thấp nhất mà nhóm đạt được là 1.8s cho 12 tấm hình.