**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM  
KHOA CÔNG NGHỆ THÔNG TIN**

**-----\*\*\*-----**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN “THỊ GIÁC MÁY TÍNH”**

***Đề tài:* *NHẬN DẠNG TIỀN CỦA VIỆT NAM***

***GVHD: Hồ Thị Hương Thơm***

***Sinh viên thực hiện: Lê Sỹ Đức Mạnh -73879***

***Đào Quang Huy-73352***

**BÁO CÁO ĐỒ ÁN MÔN**

**THỊ GIÁC MÁY TÍNH**

*Đề tài:*

NHẬN DIỆN TIỀN VỚI MÁY TÍNH

Giảng viên hướng dẫn:

**Hồ Thị Hương Thơm**

Nhóm sinh viên thực hiện:

Lê Sỹ Đức Mạnh-73879

Đào Quang Huy-73352

**Hải Phòng, ngày 7 tháng 12 năm 2020**

Mục lục

[**1.** **Giới thiệu đề tài** 4](#_Toc58253588)

[**2.** **Sơ đồ hoạt động của tệp tin rút trích đặc trưng và nhận diện** 4](#_Toc58253589)

[*HÌnh 2.1. Sơ đồ hoạt động của tệp tin rút trích đặc trưng* 4](#_Toc58253590)

[*HÌnh 2.2. Sơ đồ hoạt động của tệp tin rút nhận diện* 5](#_Toc58253591)

[**3.** **Cách vận hành hệ thống** 6](#_Toc58253592)

[**a.** **SIFT(Scale-invariant feature transform)** 6](#_Toc58253593)

[**b.** **SURF (Speeded-Up Robust Features)** 7](#_Toc58253594)

[**c.** **AKAZE** 8](#_Toc58253595)

[**5.** **Phần mềm sử dụng** 8](#_Toc58253596)

[**6.** **Một số hình ảnh về sản phẩm thực tế** 8](#_Toc58253597)

[9](#_Toc58253600)

[**7.** **Một số hình ảnh mẫu về tiền để rút trích đặc trưng** 10](#_Toc58253601)

[*Hình 9. Các hình mẫu về tiền để rút trích đặc trưng* 10](#_Toc58253602)

[**8.** **So sánh thời gian nhận diện giữa các thuật toán và thời gian rút trích đặc trưng** 10](#_Toc58253603)

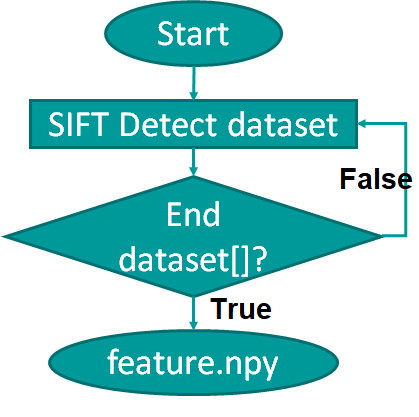
[*Hình 10. Biểu đồ so sánh thời gian nhận diện và thời gian rút trích đặc trưng* 10](#_Toc58253604)

[**9.** **Hướng phát triển** 11](#_Toc58253605)

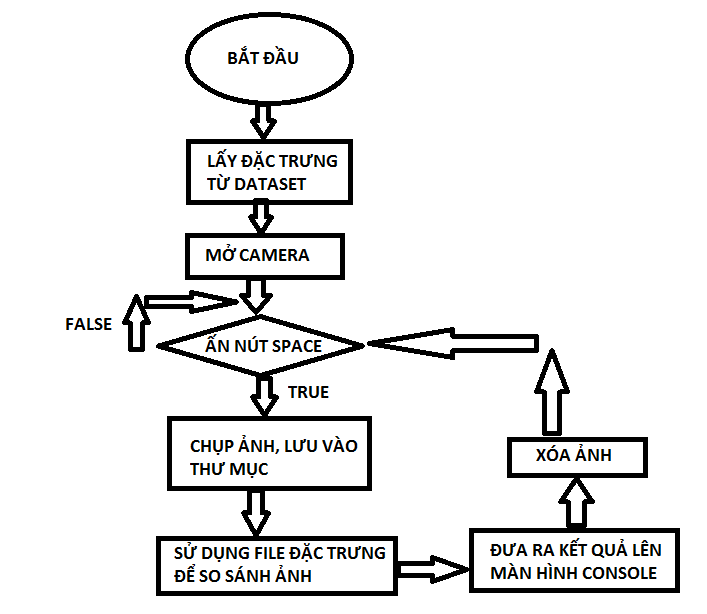
1. **Giới thiệu đề tài**

Đề tài nhóm chúng em làm về một thiết bị có khả năng nhận diện tiền và hiển thị lên màn hình **Console** thông tin tiền đã nhận diện, 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 : Sử dụng DroidCamApp để kết nối điện thoại với máy tính, biến điện thoại thành camera di động của máy tính

- Bước 2: Chạy file CreateFeatureFile.py để rút ra file đặc trưng feature.npytừ folder TrainingData chứa các ảnh đã được chụp lại trước

- Bước 3 : Chạy file Detect.py để tiến hành mở camera

- Bước 4 : Nhét tiền vào hộp, yêu cầu tiền khi đưa vào phải được duỗi thẳng

- Bước 5 : Khi tiền đã được đưa vào hộp hiển thị rõ rang trên màn hình điện thoại tiến hành ấn phím space (phím khoảng trắng) để tiến hành chụp ảnh và đưa ra kết quả

- Bước 6: Ấn phím ESC để thoát khỏi chương trình

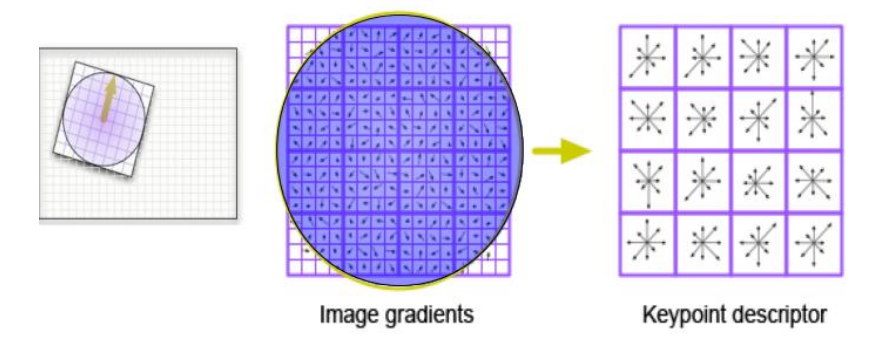
**Phần cứng**

1. **Thuật toán sử dụng**
2. **SIFT(Scale-invariant feature transform)**

SIFT (Scale-invariant feature transform) là một feature descriptor được sử dụng trong computer vision và xử lý hình ảnh được dùng để nhận dạng đối tượng, matching image, hay áp dụng cho các bài toán phân loại...

Với đầu vào là một hình ảnh >>> SIFT >>> các keypoint. Mỗi đối tượng trong hình ảnh sẽ cho ra rất nhiều các keypoint khác nhau, để ta phân biệt được các keypoint này với nhau sẽ thông qua một vector 128 chiều hay còn gọi là descriptor. Các descriptor này sẽ được dùng để nhận dạng đối tượng trong ảnh, hay dùng cho các bài toán classication.

Hình ảnh sau khi áp dụng biến đổi SIFT, ứng với mỗi keypoint ta sẽ thu được: toạ độ keypoint, scale và orientation của keypoint, descriptor. Các mũi tên trong hình dưới vẽ nhờ vào scale và orientation.

****

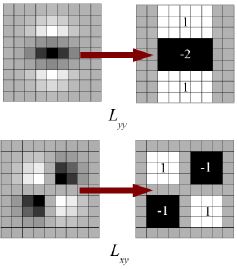
##### **Đặc điểm :**

* Các keypoint sẽ ít bị phụ thuộc bởi cường độ sáng, nhiễu, góc xoay của ảnh do các descriptor được tạo ra từ gradients do đó nó đã bất biến với các thay đổi về độ sáng (ví dụ: thêm 10 vào tất cả các pixel hình ảnh sẽ mang lại cùng một mô tả chính xác).
* Nhanh và hiệu quả, tốc độ xử lý gần như với thời gian thực (realtime)
* Có thể xử lý khi xoay ảnh

1. **SURF (Speeded-Up Robust Features)**

SIFT để phát hiện và mô tả keypoint. Nhưng nó tương đối chậm và mọi người cần phiên bản tăng tốc hơn. Năm 2006, ba người Bay, H., Tuytelaars, T. và Van Gool, L, đã xuất bản một bài báo khác, “SURF: Tăng tốc độ mạnh mẽ” giới thiệu một thuật toán mới gọi là SURF. Như tên cho thấy, nó là một phiên bản tăng tốc của SIFT.

Trong SIFT, Lowe đã xấp xỉ Laplacian của Gaussian với Chênh lệch của Gaussian để tìm không gian tỷ lệ. SURF đi xa hơn một chút và ước tính LoG với Bộ lọc hộp. Hình ảnh dưới đây cho thấy một minh chứng về sự gần đúng như vậy. Một ưu điểm lớn của phép tính gần đúng này là, tích chập với bộ lọc hộp có thể dễ dàng tính được với sự trợ giúp của hình ảnh tích phân. Và nó có thể được thực hiện song song cho các quy mô khác nhau. Ngoài ra, SURF dựa vào yếu tố quyết định của ma trận Hessian cho cả quy mô và vị trí.



SURF bổ sung rất nhiều tính năng để cải thiện tốc độ trong từng bước. Phân tích cho thấy nó nhanh hơn 3 lần so với SIFT trong khi hiệu suất tương đương với SIFT. SURF xử lý tốt các hình ảnh bị mờ và xoay, nhưng không tốt trong việc xử lý thay đổi điểm nhìn và thay đổi độ sáng.

1. **AKAZE**

*PF Alcantarilla* và cộng sự. đã trình bày thuật toán *Accelerated-KAZE (AKAZE* ) vào năm 2013 [16] , cũng dựa trên lọc khuếch tán phi tuyến giống như KAZE nhưng không gian tỷ lệ phi tuyến tính của nó được xây dựng bằng cách sử dụng một khuôn khổ tính toán hiệu quả có tên là *Fast Explicit Diffusion (FED* ). Máy dò AKAZE dựa trên *yếu tố quyết định của Ma trận Hessian* . Chất lượng bất biến quay được cải thiện bằng cách sử dụng *bộ lọc Scharr* . Cực đại của các phản hồi của máy dò ở các vị trí không gian được chọn làm điểm đặc trưng. Bộ mô tả của AKAZE dựa trên *Nhị phân khác biệt cục bộ được sửa đổi (MLDB*) thuật toán cũng có hiệu quả cao. Các tính năng của AKAZE luôn thay đổi theo tỷ lệ, xoay, giới hạn liên kết và có tính phân biệt hơn ở các tỷ lệ khác nhau vì không gian tỷ lệ phi tuyến.

1. **Phần mềm sử dụng**

* OpenCV 3.4.3
* Python3
* Sift/Suft/Akaze library
* Numpy
* DroidCamApp

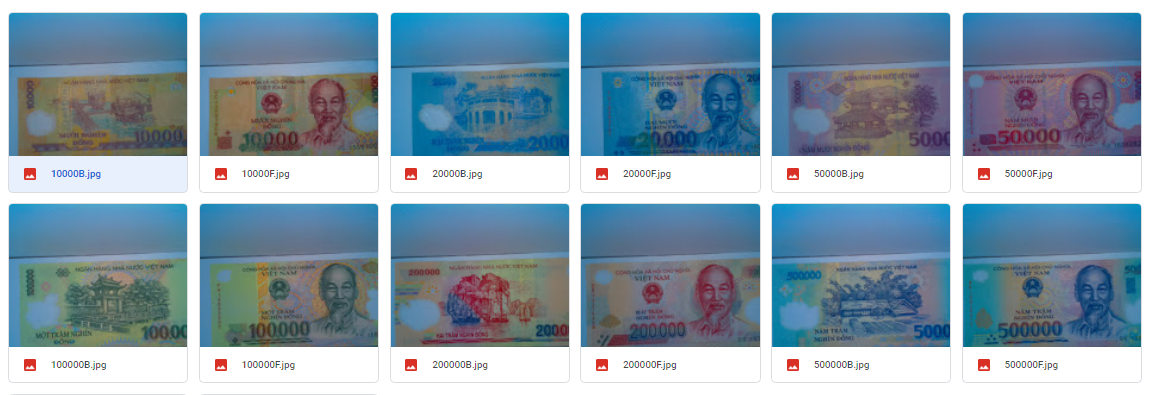
1. **Một số hình ảnh về sản phẩm thực tế**

****

****

****

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. **Hướng phát triển**

Sử dụng thư viện nhúng có sẵn trong python, và một số linh kiện phần cứng như *mạch Raspberry Pi 3B+,camera pi* tạo ra một thiết bị có khả năng nhận diện tiền việt nam hiển thị lên màn hình LCD.