**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**NIÊN LUẬN**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**TÊN ĐỀ TÀI**

**DỰ ĐOÁN TỶ LỆ RATING DỰA TRÊN GIÁ**

**Sinh viên thực hiện**

**Trần Hữu Hiền -DC18V7K307**

**Khoá: K44**

**Cần Thơ, 2021**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**NIÊN LUẬN**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**TÊN ĐỀ TÀI**

**DỰ ĐOÁN TỶ LỆ RATING DỰA TRÊN GIÁ**

**Người hướng dẫn Sinh viên thực hiện**

**TS. Lâm Nhựt Khang Trần Hữu Hiền – DC18V7K311**

**Khóa: K44**

**Cần Thơ, 2021**

**Lời cảm ơn!**

Để niên luận này đạt kết quả tốt đẹp, tôi đã nhận được sự hỗ trợ, giúp đỡ của nhiều cơ quan, tổ chức, cá nhân. Với tình cảm sâu sắc, chân thành, cho phép tôi được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân và cơ quan đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu đề tài.

Trước hết tôi xin gửi tới các thầy cô khoa Công nghệ thông tin trường Đại học Cần Thơ lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay tôi đã có thể hoàn thành luận văn, đề tài: “DỰ ĐOÁN TỶ LỆ RATING DỰA TRÊN GIÁ”.

Đặc biệt tôi xin gửi lời cảm ơn chân thành nhất tới giáo viên hướng dẫn – TS. Lâm Nhựt Khang đã quan tâm giúp đỡ, hướng dẫn tôi hoàn thành tốt niên luận này trong thời gian qua.

Tôi xin bày tỏ lòng biết ơn đến lãnh đạo Trường Đại học Cần Thơ, các Khoa Phòng ban chức năng đã trực tiếp và gián tiếp giúp đỡ tôi trong suốt quá trình học tập và nghiên cứu đề tài.

Không thể không nhắc tới sự chỉ đạo của các thầy cô bộ môn khác cùng sự giúp đỡ nhiệt tình của các anh chị và bạn bè cùng lớp và khác lớp, đã tạo điều kiện thuận lợi nhất cho tôi trong suốt thời gian thực hiện niên luận.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một học viên, luận văn này không thể tránh được những thiếu sót. Tôi rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để tôi có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này

Tôi xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | *.........., ngày....tháng....năm....* |
|  | **Sinh viên** |

**MỤC LỤC**

🙡🏵🙣

[Phần 1. Giới Thiệu 1](#_Toc29801018)

[1.1. Đặt vấn đề 1](#_Toc29801019)

[1.2. Những nghiên cứu liên quan 2](#_Toc29801020)

[1.3. Mục tiêu đề tài 4](#_Toc29801021)

[1.4. Đối tượng và phạm vi nghiên cứu 4](#_Toc29801022)

[1.5. Nội dung nghiên cứu 5](#_Toc29801023)

[1.6. Bố cục của quyền luận văn 5](#_Toc29801024)

[Phần 2. Nội dung 6](#_Toc29801025)

[2.1. Đặc tả yêu cầu 6](#_Toc29801026)

[2.2. Thiết kế giải pháp 7](#_Toc29801027)

[2.2.1. Thị giác máy tính (Computer Vison) 7](#_Toc29801028)

[2.2.2. Khái niệm cơ bản về xử lý ảnh 11](#_Toc29801029)

[2.2.2.1 Ảnh kỹ thuật số (Digital image) 11](#_Toc29801030)

[2.2.2.2 Độ phân giải của ảnh (Resolution) 15](#_Toc29801031)

[2.2.2.3 Điểm ảnh (Pixel) 16](#_Toc29801032)

[2.2.2.4 Mức xám của ảnh (Grayscale) 17](#_Toc29801033)

[2.2.2.5 Ảnh màu 18](#_Toc29801034)

[2.2.2.6 Ảnh trắng đen - xám 20](#_Toc29801035)

[2.2.2.7 Hệ màu RGB 22](#_Toc29801036)

[2.2.3. Các phép toán xử lý ảnh áp dụng trong đề tài 23](#_Toc29801037)

[2.2.3.1 Biến đổi ảnh xám (Grayscale convert) 23](#_Toc29801038)

[2.2.3.2 Làm mờ ảnh (Blur) 25](#_Toc29801039)

[2.2.3.3 Phép toán giãn nở (Dilation) 28](#_Toc29801040)

[2.2.3.4 Phép toán co (Erosion) 30](#_Toc29801041)

[2.2.3.5 Xác định cạnh trong ảnh (Edge Detection) 32](#_Toc29801042)

[2.2.3.6 Contours (đường viền) và Find contours (tìm đường viền) 37](#_Toc29801043)

[2.2.3.7 Giải thuật Euclidean 38](#_Toc29801044)

[2.2.4. Tổng quan về thư viện OpenCV 38](#_Toc29801045)

[2.2.4.1 Tổng quan 38](#_Toc29801046)

[2.2.4.2 Ứng dụng và chức năng của OpenCV 40](#_Toc29801047)

[2.2.5. Ngôn ngũ lập trình Python 41](#_Toc29801048)

[2.3. Cài đặt giải pháp 43](#_Toc29801049)

[2.3.1. Phân tích và thiết kế 43](#_Toc29801050)

[2.3.1.1 Đặc trưng của đối tượng 43](#_Toc29801051)

[2.3.1.2 Lưu đồ giải thuật 44](#_Toc29801052)

[2.3.1.3 Quy trình hoạt động của ứng dụng 45](#_Toc29801053)

[2.3.1.4 Hướng giải quyết và kế hoạch thực hiện 46](#_Toc29801054)

[2.3.2. Cài đặt 47](#_Toc29801055)

[2.3.2.1 Tiền xử lý ảnh 47](#_Toc29801056)

[2.3.2.2 Xác định đối tượng 51](#_Toc29801057)

[2.3.2.3 Tính kích thước đối tượng 52](#_Toc29801058)

[2.3.3. Kết quả 54](#_Toc29801059)

[2.4. Đánh giá kiểm thử 54](#_Toc29801060)

[2.4.1. Mục tiêu 54](#_Toc29801061)

[2.4.2. Kịch bản 55](#_Toc29801062)

[2.4.3. Kết quả 56](#_Toc29801063)

[Phần 3. Kết luận 56](#_Toc29801064)

[3.1. Kết luận đạt được 56](#_Toc29801065)

[3.2. Hướng phát triển 57](#_Toc29801066)

[Phần 4. TÀI LIỆU THAM KHẢO 58](#_Toc29801067)

[Phần 5. Phụ lục 59](#_Toc29801068)

**Danh mục từ viết tắt**

|  |  |
| --- | --- |
| **Chữ viết tắt** | **Nguyên nghĩa** |
| OpenCV | Open Source Computer Vision Library |
| RGB | Red, Green, Blue color space |
| IEEE | Institute of Electrical and Electronics Engineer |
| VOC | Visual Object Classes |
| SIFT | Scale-invariant feature transform |

**Danh mục hình ảnh**

[Hình 1: Mô tả nhận dạng đối tượng trong hình ảnh 6](#_Toc29800942)

[Hình 2: Biểu đồ mô tả đường hướng, tìm đường viền và các đặc điểm khác trong vùng hình ảnh của não 9](#_Toc29800943)

[Hình 3:Nhận thức hình ảnh của máy tính 10](#_Toc29800944)

[Hình 4:Hình mô tả bitmap 12](#_Toc29800945)

[Hình 5: Mô tả vector và bitmap 14](#_Toc29800946)

[Hình 6: Điểm ảnh trong hình 15](#_Toc29800947)

[Hình 7: Mô tả dpi cần thiết khi in 16](#_Toc29800948)

[Hình 8: Giới thiệu pixel trong hình ảnh 17](#_Toc29800949)

[Hình 9: Biểu diễn mức xám ảnh số 18](#_Toc29800950)

[Hình 10: Hình ảnh màu Mathematical bridge, Cambridge 19](#_Toc29800951)

[Hình 11: Chi tiết nội thuộc tính hình ảnh 19](#_Toc29800952)

[Hình 12: Ảnh màu kích thước 3x3 bằng ma trận, giá trị rgb 19](#_Toc29800953)

[Hình 13: Tách ma trận thành 3 ma trận cùng kích thước, mỗi ma trận lưu giá trị là red, green, blue 20](#_Toc29800954)

[Hình 14: Tách ma trận biểu diễn màu ra 3 ma trận, mỗi ma trận lưu giá trị 1 màu 20](#_Toc29800955)

[Hình 15: Ảnh xám của mathematical bridge 21](#_Toc29800956)

[Hình 16: Ma trận 3x3 tương tự như anh màu 21](#_Toc29800957)

[Hình 17: Biểu diễn ảnh xám 22](#_Toc29800958)

[Hình 18: Hệ màu RGB 22](#_Toc29800959)

[Hình 19: Màu được chọn là rgb (102, 255, 153), nghĩa là r=102, g=255, b=153 23](#_Toc29800960)

[Hình 20: Biểu diễn ảnh màu RGB 24](#_Toc29800961)

[Hình 21: Kết quả từ ảnh màu thành ảnh xám 25](#_Toc29800962)

[Hình 22: Phép toán biến đổi điểm ảnh xung quanh 25](#_Toc29800963)

[Hình 23: Công thức tính Convolution 1D 26](#_Toc29800964)

[Hình 24: Công thức tính Convolution 1D về dạng rời rạc 26](#_Toc29800965)

[Hình 25: Kết quả áp dụng công thức Convolution 1D 26](#_Toc29800966)

[Hình 26: Công thức tính Convolution 2D 26](#_Toc29800967)

[Hình 27: Áp dụng công thức Convolution 2D 27](#_Toc29800968)

[Hình 28: Minh hoạ hình ảnh bị nhiễu 27](#_Toc29800969)

[Hình 29: Mô tả ma trận nhân 28](#_Toc29800970)

[Hình 30: Kết quả áp dụng phép toán làm mờ 28](#_Toc29800971)

[Hình 31: Phép toán giãn nở trên ảnh đa mức xám 28](#_Toc29800972)

[Hình 32: Minh hoạ phần tử cấu trúc B của phép toán giãn nở 29](#_Toc29800973)

[Hình 33: Minh hoạ ma trận đa mức xám 29](#_Toc29800974)

[Hình 34: Kết quả minh hoa sau khi sử dụng phép toán giãn nở 30](#_Toc29800975)

[Hình 35: Công thức phép toán co trên ảnh đa mức xám 30](#_Toc29800976)

[Hình 36: Minh hoạ phần tử cấu trúc B của phép toán giãn co 30](#_Toc29800977)

[Hình 37: Minh hoạ ma trận ảnh đa mức xám với phép toán co 31](#_Toc29800978)

[Hình 38: Minh hoạ ma trận sau khi tính toán trừ các giá trị vùng đỏ cho 10 31](#_Toc29800979)

[Hình 39: Minh hoạ hình ảnh còn giữ lại vì tương thích với hình gốc 32](#_Toc29800980)

[Hình 40: Minh hoạ ma trận loại bỏ những điểm ảnh không phù hợp 32](#_Toc29800981)

[Hình 41: Minh hoạ tìm cạnh 33](#_Toc29800982)

[Hình 42: Minh hoạ sau khi phóng to cạnh trong hình 34](#_Toc29800983)

[Hình 43: Ma trận pixel độ sáng từ 0 – 255 của hình 34](#_Toc29800984)

[Hình 44: Ảnh RGB sau khi phát hiện cạnh 35](#_Toc29800985)

[Hình 45: Công thức tính ảnh Gradient và Angle 35](#_Toc29800986)

[Hình 46: Mô tả loại bỏ các pixel không cực đại toàn cục 36](#_Toc29800987)

[Hình 47: Ảnh minh họa về ngưỡng lọc 36](#_Toc29800988)

[Hình 48: Kết quả sử dụng Canny 37](#_Toc29800989)

[Hình 49: Mô tả đường viền 37](#_Toc29800990)

[Hình 50: Kết quả tìm đường viền sau khi áp dụng hàm 38](#_Toc29800991)

[Hình 51: Các module file nén OpenCV 4.1.2 39](#_Toc29800992)

[Hình 52: Logo ngôn ngữ lập trình Python 41](#_Toc29800993)

[Hình 53: Xếp hạng ngôn ngữ lập trình năm 2016 42](#_Toc29800994)

[Hình 54: Ngôn ngữ python được sử dụng rộng rãi ở các trường giảng dạy 43](#_Toc29800995)

[Hình 55: Mô tả mẫu đặc trưng khuyến nghị 44](#_Toc29800996)

[Hình 56: Mô tả quy trình hoạt động phát hiện và tính kích thước đồ vật 45](#_Toc29800997)

[Hình 57: Đối tượng tham chiếu được xác định từ trái sang phải 46](#_Toc29800998)

[Hình 58: Hình ảnh demo RGB mặc định 47](#_Toc29800999)

[Hình 59: Kết quả sau khi chuyển ảnh sang độ xám 48](#_Toc29801000)

[Hình 60: Kết quả sau khi làm mờ 49](#_Toc29801001)

[Hình 61: Kết quả sau khi phát hiện cạnh 49](#_Toc29801002)

[Hình 62: Kết quả thực hiện giãn nỡ các cạnh đối tượng 50](#_Toc29801003)

[Hình 63: Kết quả thực hiện phép co 50](#_Toc29801004)

[Hình 64: Kết quả vẽ đường viền và tô màu đường viền đối tượng 51](#_Toc29801005)

[Hình 65: Kết quả vẽ 4 vòng tròn nhỏ ở các gốc 52](#_Toc29801006)

[Hình 66: Kết quả vẽ các điểm giữa màu xanh dương 53](#_Toc29801007)

[Hình 67: Kết quả vẽ đường chéo màu hồng trên đối tượng 53](#_Toc29801008)

[Hình 68: Kết quả cuối cùng sau khi ứng dụng thực hiện tính toán 54](#_Toc29801009)

[Hình 69: Mẫu kịch bản 1 (khuyên dùng) 55](#_Toc29801010)

[Hình 70: Mẫu kịch bản 2 55](#_Toc29801011)

[Hình 71: Nội dung file Project.rar 60](#_Toc29801012)

[Hình 72: Chọn File tại thư mục hiện hành 60](#_Toc29801013)

[Hình 73: Chọn Open Windows PowerShell as administrator 60](#_Toc29801014)

[Hình 74: Giao diện PowerShell 61](#_Toc29801015)

[Hình 75: Thực hiện cú pháp chạy chương trình 61](#_Toc29801016)

[Hình 76: Kết quả thực thi 62](#_Toc29801017)

**Abstract**

**Tóm tắt**

# Giới Thiệu

## Đặt vấn đề

Trong kinh doanh truyền thống, muốn biết được khách hàng sẽ phản hồi thế nào với các mức giá mình đưa ra trong tương lai là rất khó khăn. Đó là lý do tôi chọn đề tài “Dự đoán tỷ lệ rating dựa trên giá”, nhằm mục đích xác định và dự đoán tỷ lệ rating của khách hàng dựa theo các mức giá đặt ra, nhằm giúp nhà quản lý quyết định được mức giá hợp lý.

## Mục tiêu đề tài

Đề tài có 2 phần, phần đầu lấy dữ liệu từ trang [www.booking.com](http://www.booking.com) bằng python, lưu vào csv, phần sau tập trung nghiên cứu mô hình LSTM để dự đoán tỷ lệ rating dựa trên dữ liệu csv đã được tạo.

Triển khai nghiên cứu bằng việc thu thập dữ liệu trên website [www.booking.com](http://www.booking.com) bằng python. Kết quả đầu ra là phần mềm khi nhập vào giá, sẽ ra kết quả dự đoán tỷ lệ rating.

## Đối tượng và phạm vi nghiên cứu

Dữ liệu giá phòng và tỷ lệ rating của trang website www.booking.com.

## Bố cục

Niên luận sẽ thực hiện nghiên cứu về việc dự đoán tỷ lệ rating dựa trên giá với những nội dung sau:

**PHẦN 1: GIỚI THIỆU**

Giới thiệu tổng quan về đề tài, xác định vấn đề, bối cảnh và nội dung bài toán cần giải quyết.

**PHẦN 2: NỘI DUNG**

Đặc tả yêu cầu giới thiệu về các giải pháp sẽ xử lý trong đề tài, cải tiến, khắc phục. Giới thiệu các cơ sở lý thuyết, khái niệm cơ bản về maching learning, mô hình LSTM, giới thiệu tổng quan về ngôn ngữ lập trình python và các khái niệm liên quan. Cuối cùng kết quả thực hiện.

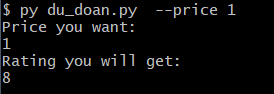
**PHẦN 3: KẾT LUẬN**

Thể hiện cách giải quyết bài toán thông qua hệ thống thực nghiệm. Nêu lên các thiết kế về chương trình và trình bày về thực nghiệm chương trình kèm theo hướng dẫn.

# Nội dung

## Đặc tả yêu cầu

Xây dựng được một phần mềm du\_doan.py, đầu vào sẽ là giá nhà quản lý muốn nhập, đầu ra sẽ là tỷ lệ rating dự đoán.



Hình 1: Mô tả nhận dự đoán tỷ lệ rating

*Có nhiều yếu tố ảnh hưởng đến việc dự đoán tỷ lệ rating ngoài giá* :

- Vị trí.

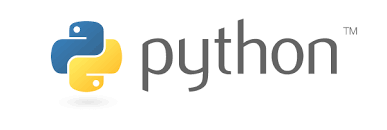
- Loại phòng (phòng đơn, phòng đôi).

- Số giường.

## Các khái niệm

### Ngôn ngũ lập trình Python

**- Python** là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level)  ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch **Python** và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.



Hình 52: Logo ngôn ngữ lập trình Python

Sau đây là các đặc điểm của **Python**:

* Ngữ pháp đơn giản, dễ đọc.
* Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented)
* Hỗ trợ module và hỗ trợ gói (package)
* Xử lý lỗi bằng ngoại lệ (Exception)
* Kiểu dữ liệu động ở mức cao.
* Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
* Có khả năng tương tác với các module khác viết trên **C/C++** (Hoặc **Java** cho **Jython**, hoặc **.Net** cho **IronPython**).
* Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

**Python dễ dàng kết nối với các thành phần khác:**

* Python có thể kết nối với các đối tượng COM, .NET (Ironpython, Python for .net), và CORBA, Java… Python cũng được hỗ trợ bởi Internet Communications Engine (ICE) và nhiều công nghệ kết nối khác.
* Có thể viết các thư viện trên C/C++ để nhúng vào Python và ngược lại.

**Python là ngôn ngữ có khả năng chạy trên nhiều nền tảng.**

* Python có cho mọi hệ điều hành: Windows, Linux/Unix, OS/2, Mac, Amiga, và những hệ điều hành khác. Thậm chí có cả những phiên bản chạy trên .NET, máy ảo Java, và điện thoại di động (Nokia Series 60). Với cùng một mã nguồn sẽ chạy giống nhau trên mọi nền tảng.

**Python rất đơn giản và dễ học**

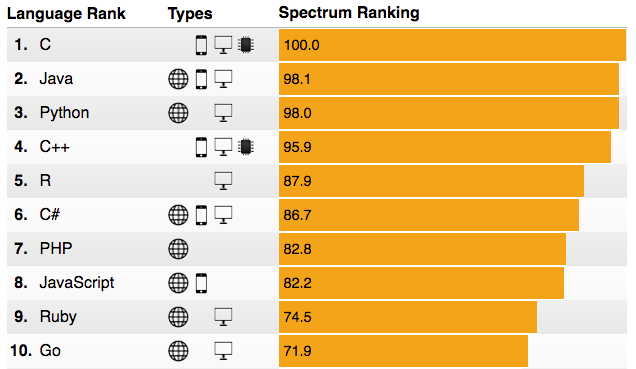
* Python có cộng đồng lập trình rất lớn, hệ thống thư viện chuẩn, và cả các thư viện mã nguồn mở được chia sẻ trên mạng.

**Python là ngôn ngữ mã nguồn mở**

* Cài đặt Python dùng giấy phép nguồn mở nên được sử dụng và phân tối tự do, ngay cả trong việc thương mại. Giấy phép Python được quản lý bởi Python Software Foundation.

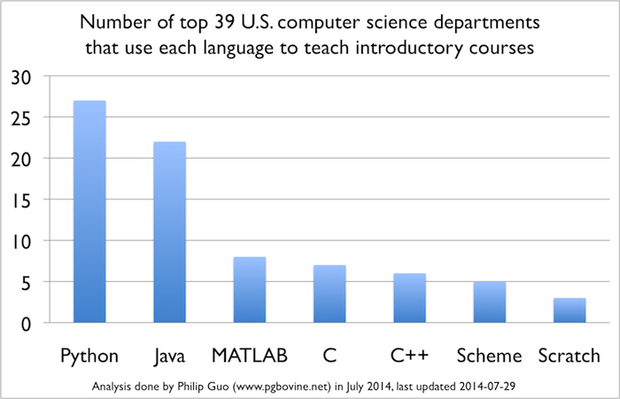
**- Xếp hạng ngôn ngữ:**

Hiện nay ngôn ngữ **Python** được xếp hạng thứ 3 trong Top 10 các ngôn ngữ lập trình phổ biến nhất đang được thế giới sử dụng. (Bảng xếp hạng năm 2016):

****

Hình 53: Xếp hạng ngôn ngữ lập trình năm 2016

Theo thống kê từ top 39 trường giảng dạy khoa học máy tính thì đa phần các trường sử dụng ngôn ngữ **Python** để giảng dạy:



Hình 54: Ngôn ngữ python được sử dụng rộng rãi ở các trường giảng dạy

- **Lịch sử:**

**Python** đã được hình thành vào cuối những năm 1980, và việc thực hiện nó vào tháng 12 năm 1989 bởi **Guido van Rossum** tại **Centrum Wiskunde & Informatica (CWI)** ở **Hà Lan** như là một kế thừa cho ngôn ngữ **ABC** (tự lấy cảm hứng từ SETL) có khả năng xử lý ngoại lệ và giao tiếp với Hệ điều hành Amoeba. **Van Rossum** là tác giả chính của **Python**, và vai trò trung tâm của ông trong việc quyết định hướng phát triển của **Python.**

### ****Machine Learning – ML (Học máy)****

ML là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Ví dụ như các máy có thể "học" cách phân loại thư điện tử xem có phải thư rác (spam) hay không và tự động xếp thư vào thư mục tương ứng. Học máy rất gần với suy diễn thống kê (statistical inference) tuy có khác nhau về thuật ngữ.

Học máy có liên quan lớn đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán NP-khó, vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được.

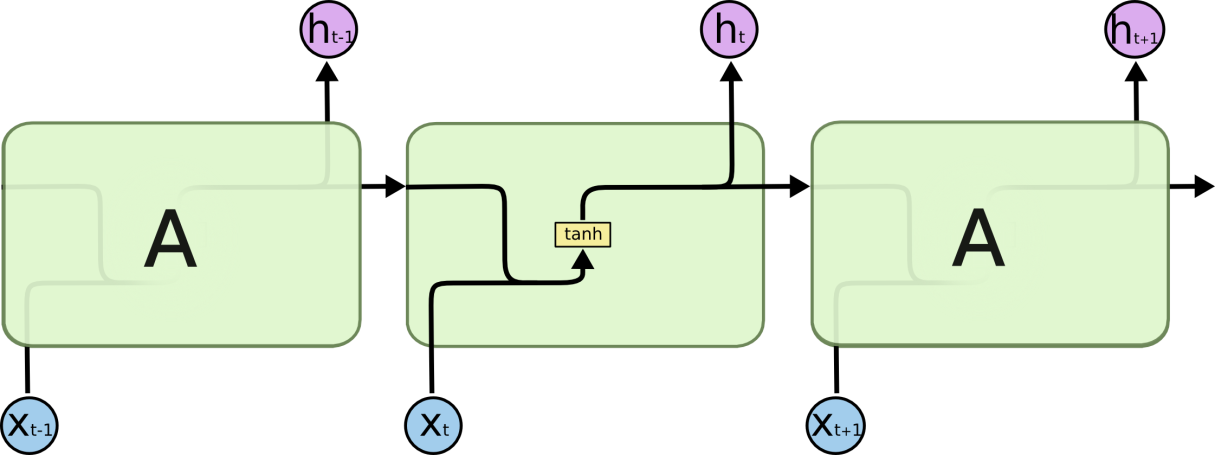
Dưới góc nhìn của trí tuệ nhân tạo, động lực chính học máy bởi là nhu cầu thu nhận tri thức (knowledge acquisition). Thật vậy, trong nhiều trường hợp ta cần kiến thức chuyên gia là khan hiếm (không đủ chuyên gia ngồi phân loại lừa đảo thẻ tín dụng của tất cả giao dịch hàng ngày) hoặc chậm vì một số nhiệm vụ cần đưa ra quyết định nhanh chóng dựa trên xử lý dữ liệu khổng lồ (trong mua bán chứng khoán phải quyết định trong vài khoảng khắc của giây chẳng hạn) và thiếu ổn định thì buộc phải cần đến máy tính. Ngoài ra, đại đa số dữ liệu sinh ra ngày nay chỉ phù hợp cho máy đọc (computer readable) tiềm tàng ngưồn kiến thức quan trọng. Máy học nghiên cứu cách thức để mô hình hóa bài toán cho phép máy tính tự động hiểu, xử lý và học từ dữ liệu để thực thi nhiệm vụ được giao cũng như cách đánh giá giúp tăng tính hiệu quả.

### Long short-term memory – LSTM (Bộ nhớ dài-ngắn hạn)

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

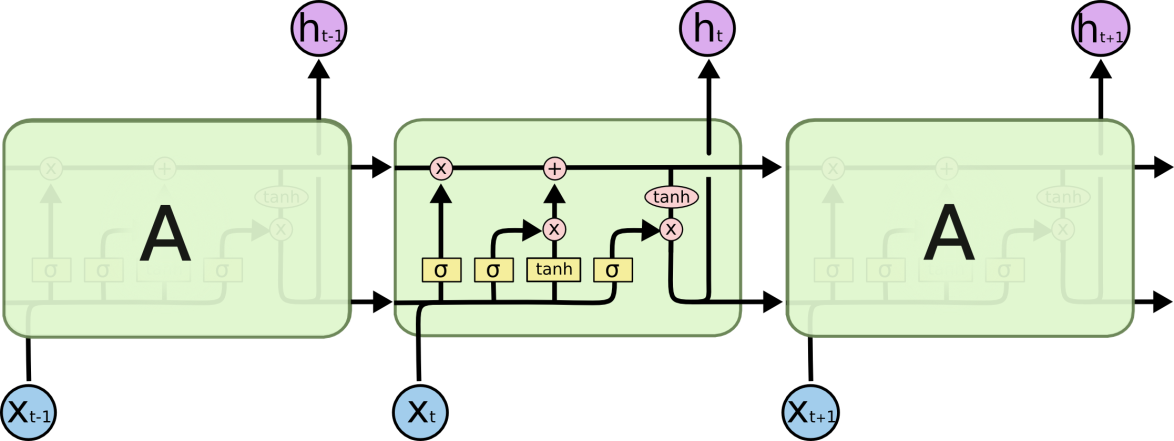
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh tanh tanh.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-SimpleRNN.png)

Mạng RNN chuẩn.

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-chain.png)

Mạng LSTM.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM2-notation.png)

Các kí hiệu của mạng LSTM

Ở sơ đồ trên, mỗi một đường mang một véc-tơ từ đầu ra của một nút tới đầu vào của một nút khác. Các hình trong màu hồng biểu diễn các phép toán như phép cộng véc-tơ chẳng hạn, còn các ô màu vàng được sử dụng để học trong các từng mạng nơ-ron. Các đường hợp nhau kí hiệu việc kết hợp, còn các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và chuyển tới các nơi khác nhau.

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ.

Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-C-line.png)

Trạng thái tế bào

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-gate.png)

Cổng

Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0,1] [0, 1] [0,1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 0 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 1 1 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là “tầng cổng quên” (forget gate layer). Nó sẽ lấy đầu vào là ht−1 h\_{t-1} ht−1​ và xt x\_t xt​ rồi đưa ra kết quả là một số trong khoảng [0,1] [0, 1] [0,1] cho mỗi số trong trạng thái tế bào Ct−1 C\_{t-1} Ct−1​. Đẩu ra là 1 1 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 0 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.

## Cài đặt giải pháp

### Phân tích và thiết kế

#### Đặc trưng của đối tượng

Trong đề tài lần này nhóm ưu tiên xét các đối tượng nhỏ cụ thể là các vật dụng nhỏ được sử dụng hằng ngày như: đồng xu, viên thuốc, máy tính bỏ túi, cây thước, v…v…

Trong đó ưu tiên các độ vật nằm trên nền đen, giúp cho việc nhận dạng đối tượng dễ dàng hơn. Tránh các trường hợp nhận dạng đối tượng không chính xác.



Hình 55: Mô tả mẫu đặc trưng khuyến nghị

#### Lưu đồ giải thuật

Hình Ảnh

Tải hình ảnh

Chuyển ảnh xám

Làm mờ ảnh

Phát hiện cạnh.

Chỉnh giãn nở.

Đóng khoảng cách giữa đối tượng

Tìm đường viền trong tại cạnh.

Sắp xếp các đường viền từ trái sang phải và khởi tạo biến‘pixelsPerMetric’

Lặp đường viền

Đối tượng hình ảnh

So sánh độ dài đường viền < 100

Tính hộp giới hạn xoay của đương viền.

Sắp xếp các điểm trong đường viền từ bên trái, trên, phải, dưới.

Vẽ đường viền

Tính điểm giữa phía bên trái, trên, phải, dưới

Vẽ điểm giữa trên hình ảnh.

Vẽ điểm giữa trên các điểm giữa

Tính khoản cách Euclide giữa các điểm

Hết đường viền

Còn đường viền

Sai

Đúng

Nếu pixelspPerMetric chưa khởi tạo

Tạo pixelPerMetric dựa vào tham số đầu vào

Lặp các điểm viền

Còn

Không Còn

Hiển thị kích thước đối tượng trên hình ảnh.

Sai

Đúng

Hiển thị ảnh.

#### Quy trình hoạt động của ứng dụng

Hình ảnh chứa đối tượng

Kích thước và khoảng cáchmỗi đối tượng trong hình

Tiền xử lý ảnh

Xác định các đối tượng trong hình

Tính kích thước và tính khoảng cách đối tượng

Hình 56: Mô tả quy trình hoạt động phát hiện và tính kích thước đồ vật

Khi muốn tính toán kích thước mỗi đối tượng, tôi cần chuẩn bị dữ liệu. Dữ liệu là những hình ảnh chứa các đối tượng, đối với những đối tượng trong hình ảnh thì tôi sẽ sắp xếp ngẫu nhiên những đối tượng với hình nền là màu đen. Sau đó tôi tiến hành tiền xử lý ảnh như là: chuyển hình ảnh thành hình ảnh xám (BRG2GRAY), làm mờ ảnh. Sau đó tới phần xác định đối tượng bằng cách: phát hiện cạnh của đối tượng(giãn nở đường viền, thu giãn nở đường viền), tìm đường viền trong cạnh, sắp xếp các đường viền từ trái sang phải,...v…v… Tính kích thước đối tượng sau từng lần lặp và cuối cùng là hiển thị kết quả gồm: vẽ khung hình chữ nhật xung quanh đối tượng và hiển thị số inch sau khi tính toán. Tiếp theo tính khoảng cách đối tượng trong hình.

#### Hướng giải quyết và kế hoạch thực hiện

Đo kích thước của các đối tượng trong một hình ảnh tương tự như tính toán khoảng cách từ máy ảnh của nhóm đến một đối tượng - trong cả hai trường hợp, nhóm cần xác định tỷ lệ đo số pixel trên một số liệu nhất định.

Nhóm định nghĩa tỷ lệ **“pixel\_per\_metric”**, tỷ lệ mà nhóm đã xác định rõ ràng hơn như sau:

Để xác định kích thước của một đối tượng trong ảnh, trước tiên nhóm cần thực hiện ***“hiệu chỉnh”*** bằng cách sử dụng một đối tượng tham chiếu. Đối tượng tham chiếu của nhóm có hai thuộc tính quan trọng:

- **Thuộc tính 1**: Nhóm phải biết kích thước thực tế của đối tượng trong hình (như là chiều rộng hoặc chiều cao) của một đơn vị đo lường (như mm, inch, v..v..).

- **Thuộc tính 2**: Nhóm có thể tìm thấy đối tượng tham chiếu này trong một hình ảnh, dựa trên vị trí của đối tượng (chẳng hạn như đối tượng tham chiếu luôn được đặt ở góc trên cùng bên trái của hình ảnh) hoặc thông qua các lần xuất hiện (như là một màu đặc biệt hoặc hình dạng độc đáo và khác biệt với tất cả các đối tượng khác trong hình ảnh). Trong cả hai trường hợp, nhóm sẽ chọn nhận dạng duy nhất theo một cách nào đó.

Trong ví dụ này, nhóm sẽ sử dụng tờ tiền Việt Nam làm đối tượng tham chiếu và trong tất cả các ví dụ, đảm bảo nó luôn là đối tượng ngoài cùng bên trái trong hình ảnh.



Hình 57: Đối tượng tham chiếu được xác định từ trái sang phải

Trong hình trên nhóm sẽ sử dụng một tờ tiền 10.000 VNĐ làm đối tượng tham chiếu của và đảm bảo nó luôn được đặt là đối tượng ngoài cùng bên trái trong hình ảnh, để nhóm trích xuất nó bằng cách sắp xếp các đường viền dựa trên vị trí của chúng.

Bằng cách đảm bảo tờ tiền 10.000 VNĐ là đối tượng ngoài cùng bên trái, chúng em có thể sắp xếp các đối tượng trong hình ảnh từ trái sang phải, lấy tờ tiền 10.000 VNĐ làm gốc (sẽ luôn là đường viền đầu tiên trong danh sách được sắp xếp) và sử dụng nó để xác định **“pixel\_per\_metric”** (pixel trên mỗi mét), chúng em định nghĩa là:

**pixels\_per\_metric = object\_width / know\_width**

**Ví dụ:**

Một tờ tiền 10.000VNĐ có thông số như sau :132mm x 60mm, vậy know\_width = 132mm => 5.196 inch. Bây giờ, giả sử rằng đối tượng trong hình (được đo bằng pixel) được tính là rộng 150 pixel trong code (dựa trên hộp giới hạn liên kết của nó). Áp dụng vào công thức để tính pixels\_per\_metic.

Do đó pixels\_per\_metic:

***pixels\_per\_metric = 150px / 5.196inch = 28.868 inch***

Do đó, có khoảng 29 pixel trên mỗi 5 inch trong hình ảnh. Sử dụng tỷ lệ này, nhóm có thể tính kích thước của các đối tượng khác trong một hình ảnh

### Cài đặt

#### Tiền xử lý ảnh

- Tải hình ảnh:

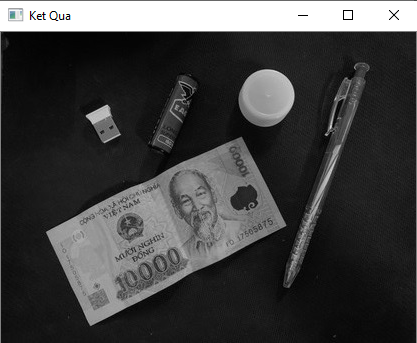


Hình 58: Hình ảnh demo RGB mặc định

Hình trên là hình ảnh đáp ứng được đặc trưng mà nhóm khuyến nghị các đồ vật nằm gần với nhau (không quá gần với nhau), với nền hình ảnh là màu đen làm nổi bật các đối tượng khác màu trong nền trong hình ảnh.

- Chuyển ảnh RGB sang grayscale (ảnh xám):

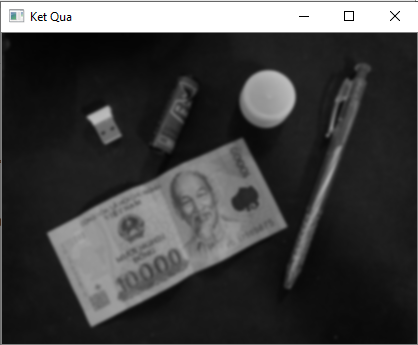
Khi đã load được hình ảnh tôi nhận thấy hình ảnh vừa load, nhóm sẽ chuyển ảnh RGB thành ảnh xám. Với phương thức xám màu là COLOR\_BGR2GRAY



Hình 59: Kết quả sau khi chuyển ảnh sang độ xám

- Làm mờ:

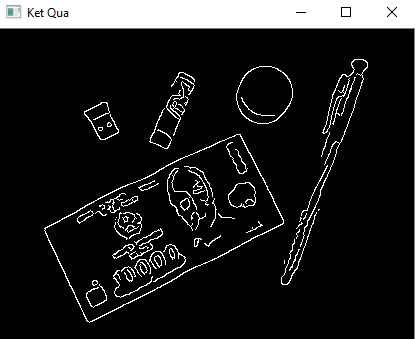
Tiếp theo dùng thư viện GaussianBlur trong OpenCV để làm mờ



Hình 60: Kết quả sau khi làm mờ

-Phát hiện cạnh:

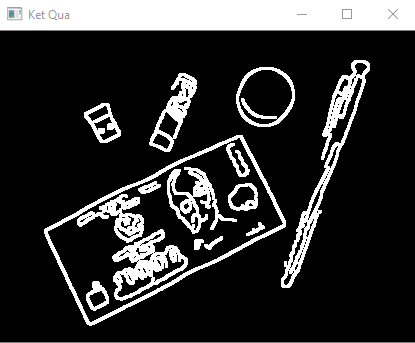
Nhóm sử dụng hàm xử lý ảnh thông dụng nhất trong OpenCV là hàm Canny. Với tham số tối thiểu 50 cao nhất là 100.



Hình 61: Kết quả sau khi phát hiện cạnh

- Thực hiện giãn nở:

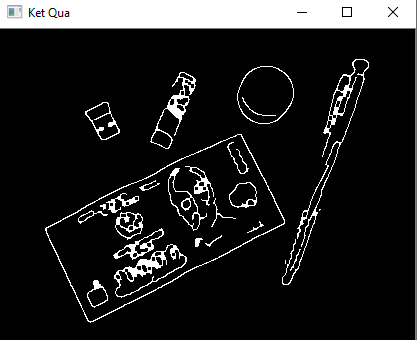
Sau khi có được các cạnh của đối tượng, nhóm thực hiện giãn nỡ các cạnh.



Hình 62: Kết quả thực hiện giãn nỡ các cạnh đối tượng

- Thực hiện co:

Thục hiện phép co trên các phép giãn nỡ, sẽ có kết quả khác so với phát hiện cạnh.



Hình 63: Kết quả thực hiện phép co

- Tìm đường viền và sắp xếp các đường viền từ trái sang phải:

Để thực hiện việc tìm đường viền, nhóm sử dụng hàm findContours, sau khi thực hiện hàm này chương trình sẽ trả về mảng ma trận nhiều chiều.

#### Xác định đối tượng

Trong hình ảnh có thể có một đối tượng hoặc nhiều đối tượng nên nhóm sẽ dùng vòng lặp để duyệt qua các đường viền đã sắp xếp.

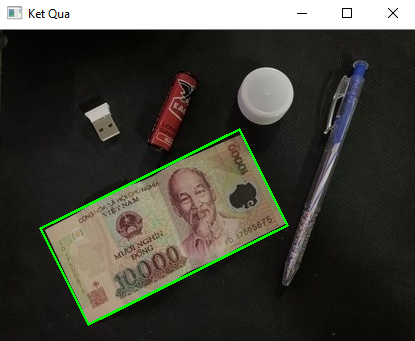
- Tạo hình chữ nhật của đối tượng.

- Tạo 4 gốc của hình chữ nhật.

- Chuyển 4 gốc thành mảng giá trị kiểu int để xử lý tiếp.

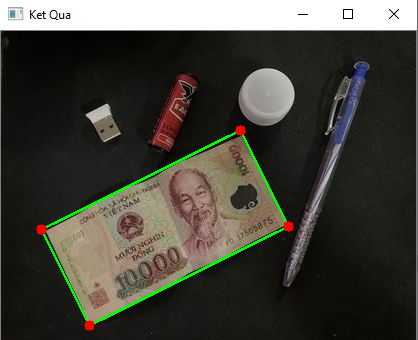
- Sắp xếp điểm của hình chữ hình

- Vẽ đường viền vào hình ảnh, tuỳ chỉnh các tham số màu sắc, độ dày



Hình 64: Kết quả vẽ đường viền và tô màu đường viền đối tượng

- Lặp các đường viền đối tượng và tạo 4 vòng tròn nhỏ ở 4 gốc đối tượng



Hình 65: Kết quả vẽ 4 vòng tròn nhỏ ở các gốc

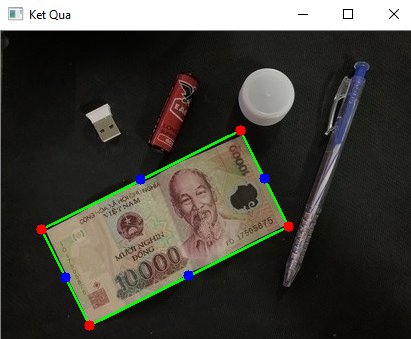
#### Tính kích thước đối tượng

Cuối cùng nhóm sẽ thực hiện tính toán kích thước của đối tượng bằng cách lấy một đối tượng đại diện hay còn gọi là đối tượng tham chiếu, trong bức hình demo là tờ tiền 10.000 VNĐ.

- Tính điểm giữa của đường viền: (trên – gốc trái, trên – gốc phải), (dưới – gốc trái, dưới – gốc phải).

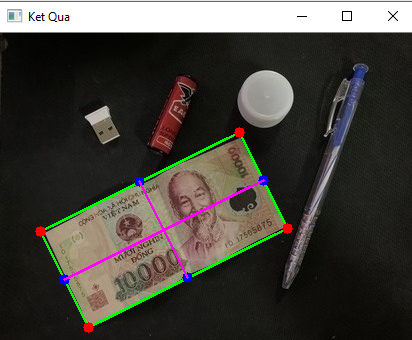
- Tính điểm giữa của đường viền: (trên – gốc trái, dưới – gốc trái), (trên – gốc phải, dưới – gốc phải).

- Vẽ các điểm giữa vừa tìm được bằng màu xanh dương



Hình 66: Kết quả vẽ các điểm giữa màu xanh dương

- Vẽ tiếp đường chéo màu hồng từ chấm tròn màu xanh dương



Hình 67: Kết quả vẽ đường chéo màu hồng trên đối tượng

- Tính khoảng cách sử dụng thuật toán Eculidean ở 4 điểm giữa để tìm chiều rộng và chiều dài:

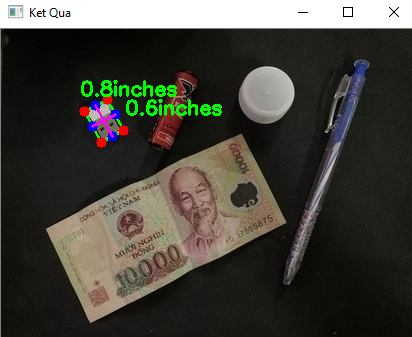
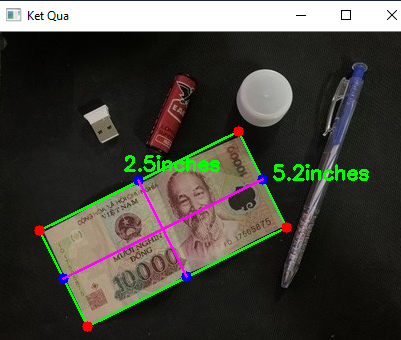
dA = (trên – gốc trái, trên – gốc phải), (dưới – gốc trái, dưới – gốc phải).

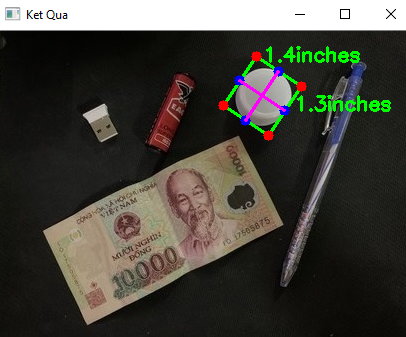
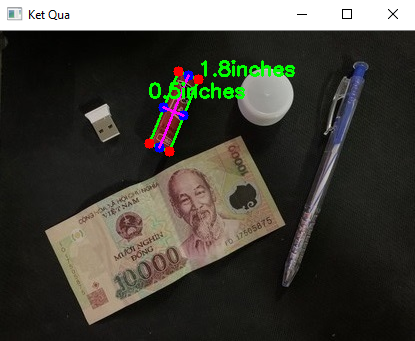
dB = (trên – gốc trái, dưới – gốc trái), (trên – gốc phải, dưới – gốc phải).

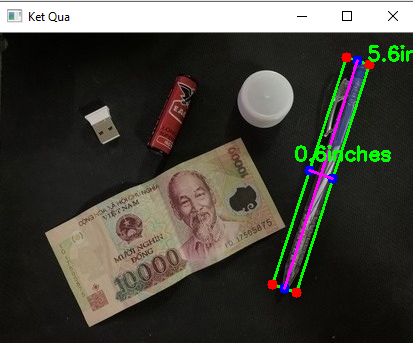
- Khi đã có chiều dài (dB) từ phép toán, nhóm sẽ kết hợp với tham số chiều dài do người dùng cung cấp để tìm ra pixel chuẩn làm đại diện để tính đối tượng.

- Cuối cùng tính kích thước đối tượng chiều rộng và chiều dài bằng cách lấy chiều dài (dB) chia cho pixelPerMetric và hiển thị kết quả kích thước đối tượng.

### Kết quả







Hình 68: Kết quả cuối cùng sau khi ứng dụng thực hiện tính toán

## Đánh giá kiểm thử

### Mục tiêu

Mục tiêu trong phần đánh giá nhóm đề xuất như sau:

- Xác định chính xác đối tượng cần đo kích thước.

- Tốc độ xử lý tính toán các đối tượng nhanh nhằm nâng cấp thời gian thực.

- Tính chiều dài và chiều rộng của đối tượng ít sai số với kích thước thực tế.

### Kịch bản

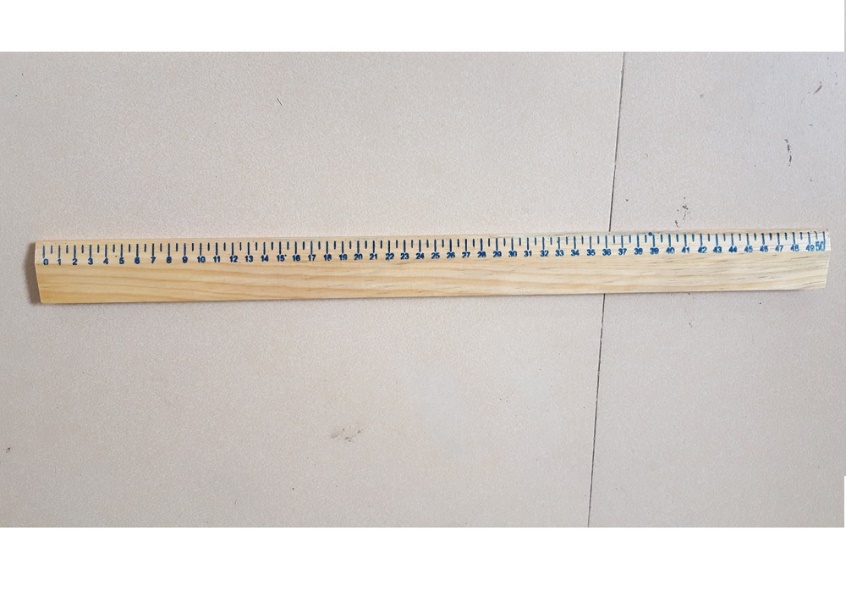
Nhóm có 2 mẫu kịch bản:

- **Mẫu 1**:Kịch bản đề xuất, với hình nền đen, nếu nhiều đối tượng thì đối tượng đại diện đặt vị trí bên trái ngoài cùng, hướng trực diện, cường độ ánh sáng bình thường.



Hình 69: Mẫu kịch bản 1 (khuyên dùng)

- **Mẫu 2**:Kịch bản không đề xuất, bất kì hình ảnh nào.



Hình 70: Mẫu kịch bản 2

Thực nghiệm tổng 2 mẫu gồm 10 hình ảnh. Mẫu kịch bản 1: 5 tấm hình, mẫu kịch bản 2: 5 tấm hình.

### Kết quả

Qua thực nghiệm trên các mẫu kịch bản nhóm cho ra kết quả sau:

- Mẫu kịch bản số 1: đạt 70% mục tiêu.

- Mẫu kịch bản số 2: không đạt 100% mục tiêu.

# Kết luận

## Kết luận đạt được

Tuy thời gian làm báo cáo niên luận tại Trường Đại Học Cần Thơ không nhiều nhưng cũng để lại trong nhóm nhiều giá trị quý báu cho hành trang sau này. Được sự giúp đỡ rất tận tình của các anh, chị chung lớp, đặc biệt là thầy **Phạm Thế Phi** giảng viên tại Trường Đại Học Cần Thơ, trong suốt quá trình thời gian thực hiện báo cáo niên luận chuyên ngành cùng với sự nỗ lực của thành viên trong nhóm. Chính vì vậy quá trình thực tập cũng đạt được các điểm đáng chú ý sau:

* Qua thực nghiệm đề tài chưa được tối ưu tốt trong nhận diện đối tượng chủ yếu do hướng chụp, cường độ ánh sáng, nền ảnh, các đối tượng nhễu, giải thuật chưa tối ưu trong nhận dạng đối tượng.
* Để tăng độ chính xác cần phải nhập vào kích thước thực tế của đối tượng đại diện.
* Dữ liệu thử nghiệm còn hạn chế.
* Được trãi nghiệm công nghệ trí tuệ nhân tạo ở thời đại công nghiệp 4.0.
* Ứng dụng đề tài có thể tái sử dụng và phát triển.
* Hiểu và học kinh nghiệm làm việc nhóm để nâng cao hiệu quả trong công việc.
* Đáp ứng và thực hiện đúng yêu cầu được giao.
* Nghiêm chỉnh chấp hành nội quy của Trường quy định.
* Cũng cố cơ sở lý thuyết chuyên môn cũng như thực tế sau khi thực hiện bài báo cáo.
* Tạo sự nhanh nhẹn, chủ động hơn trong công việc đối với từng thành viên của nhóm.

Sau khi thực hiện đề tài, nhóm thu được cách sử dụng ngôn ngữ lập trình Python, sử dụng những thư viện xử lý ảnh chuyên nghiệp hỗ trợ rất tốt để xử lý hình ảnh cụ thể là thư hiện OpenCV, biết được quá trình trình xử lý hình ảnh để cho ra kết quả như ý muốn.

Đề tài giúp nhóm có thêm kinh nghiệm trong việc nhận dạng ảnh, xử lý, tính toán kích thước của đối tượng trong ảnh, ngoài ra nhờ những kinh nghiệm lập trình cơ bản ở những môn học trong trường kết hợp với sự hướng dẫn của giáo viên nên nhóm đã có thêm kinh nghiệm lập trình, cụ thể là trình bày code, chú thích code, tối ưu những đoạn code cần thiết cũng như loại bỏ các đoạn code dư thừa.

Trong đề tài nhóm đã nhận thấy có khuyết điểm nghiêng về phần thực hành cụ thể là phần code và phần lý thuyết cụ thể là cơ sở lý thuyết và áp dụng cơ sơ lý thuyết vào đề tài. Trong phần code nhóm vẫn chưa có khả năng hiểu sâu tối ưu giải thuật, hình ảnh demo ít chưa có nhiều dữ liệu cho mọi trường hợp kiểm thử cụ thể là hình demo nghiêng, hình demo bị mờ, hình demo là khuông dạng hình tròn không phải là hình chữ nhật. Về phần lý thuyết nhóm chưa hiểu sâu cách vận dụng cụ thể một trường hợp nào khác, cần nhiều thời gian để nghiên cứu

## Hướng phát triển

Nguyên nhân dẫn đến các đến khuyết điểm và hạn chế của đề tài này chủ yếu là về mặt kiến thức và thời gian, tôi chưa nắm bắt kịp những giải thuật xử lý ảnh phúc tạp. Chưa tìm hiểu, tập hợp các thông tin liên quan đến đề tài đầy đủ, thời gian thực hành code Python còn yếu.

Nhóm xác định hướng phát triển cần tìm hiểu sâu hơn về thư viện OpenCV và một số thư viên liên quan đến xử lý ảnh khác. Ngoài ra cần có những dữ liệu nhiều dạng khác nhau để có thể kiểm thử và so sánh với thực tế

# TÀI LIỆU THAM KHẢO

Online

[1] <https://ktmt.github.io/blog/2014/09/23/xu-ly-anh-bang-python-va-opencv-mo-dau/> , lần truy cập gần nhất 11/01/2020

[2] <https://nttuan8.com/bai-5-gioi-thieu-ve-xu-ly-anh/>, lần truy cập gần nhất 11/01/2020

[3] <https://viblo.asia/p/computer-vision-object-detection-nhan-dien-vat-the-chi-voi-10-dong-code-su-dung-imageai-naQZRbdjZvx>, lần truy cập gần nhất 11/01/2020

[4] <https://viblo.asia/p/tim-hieu-ve-opencv-Do754NrXZM6>, lần truy cập gần nhất 11/01/2020

[5] <https://nttuan8.com/bai-11-object-detection-voi-faster-r-cnn/#Bai_toan_object_detection>, lần truy cập gần nhất 11/01/2020

[6] <https://blog.vietnamlab.vn/2018/06/29/nhan-dien-khuon-mat-voi-python-va-opencv/>, lần truy cập gần nhất 11/01/2020

[7] <https://blog.vietnamlab.vn/2019/03/22/xu-ly-anh-voi-opencv-tut5-can-bang-histogram/>, lần truy cập gần nhất 11/01/2020

[8] <https://techtalk.vn/top-10-thu-vien-python-tot-nhat-cho-data-scientist-nua-dau-nam-2019.html>, lần truy cập gần nhất 11/01/2020

[9] <http://www.matlabthayhai.info/2015/11/bai-1-gioi-thieu-khai-quat-ve-anh-so.html>, lần truy cập gần nhất 11/01/2020

[10] <https://nttuan8.com/bai-5-gioi-thieu-ve-xu-ly-anh/>, lần truy cập gần nhất 11/01/2020

[11] <https://univtechnews.wordpress.com/2017/04/17/tai-lieu-xu-ly-anh-contour-moments/>, lần truy cập gần nhất 11/01/2020

[12] <https://minhng.info/tutorials/xu-ly-anh-opencv-hien-thuc-canny-edge.html>, lần truy cập gần nhất 11/01/2020

[13] <https://www.stdio.vn/articles/xu-ly-anh-voi-opencv-cac-phep-toan-hinh-thai-hoc-421>, lần truy cập gần nhất 11/01/2020

[14] <https://blog.vietnamlab.vn/2018/10/09/xu-ly-anh-voi-opencv/>, lần truy cập gần nhất 11/01/2020

# Phụ lục

Trước khi thực hiện hướng dẫn demo chi tiết, tôi cần người dùng phải cài đặt các yêu cầu sau để có thể hoạt động thành công.

**Yêu cầu:**

**Hệ điểu hành Windows 10 (64 bit):**

https://www.microsoft.com/en-gb/software-download/windows10

**Ngôn ngữ lập trình Python:**

https://www.python.org/downloads/release/python-368/

**Trình soạn thảo Visual Code** (Người dùng có thể sử dụng các trình soạn thảo khác)

https://code.visualstudio.com/Download

**Thư viện xử lý ảnh**:

Mở cmd với quyền Administator và gõ các lệnh sau để cài đặt thư viện

*OpenCV*

**Pip install opencv-python**

*Imutils*

**Pip install imutils**

*Numpy*

**Pip install numpy**

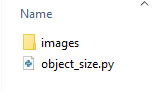
*Argparse*

**Pip install argparse**

*Scipy*

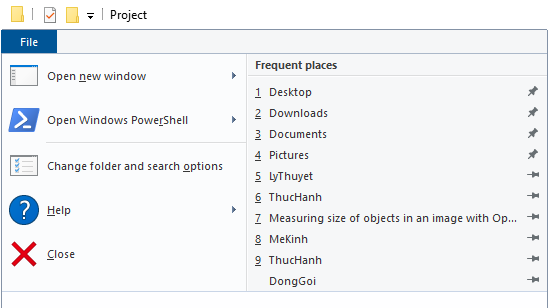
**Pip install scipy**

**Bước 1**: Giải nén file winrar Project.rar tại thư mục cá nhân. Kết quả như hình dưới.

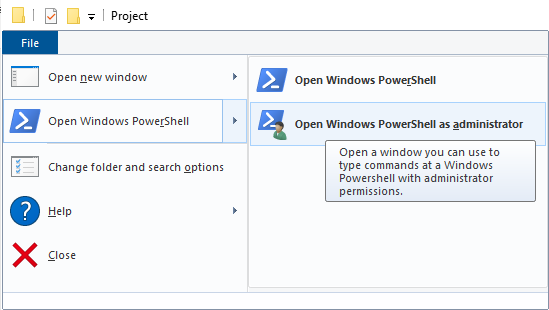


Hình 71: Nội dung file Project.rar

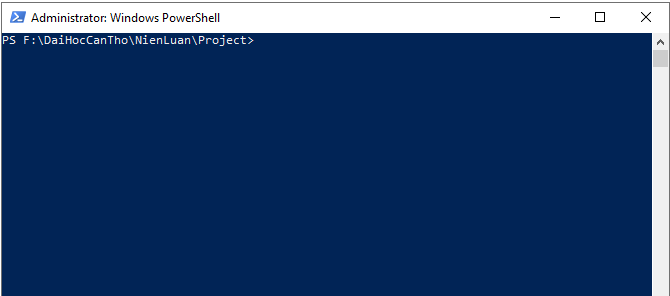
**Bước 2**: Tại thư mục hiện hành, chọn File -> Open Windows PowerShell -> Open Windows PowerShell as administator



Hình 72: Chọn File tại thư mục hiện hành



Hình 73: Chọn Open Windows PowerShell as administrator



Hình 74: Giao diện PowerShell

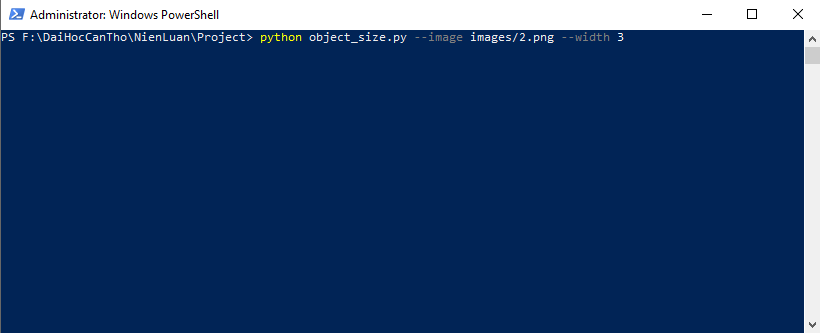
**Bước 3**: Tại màn hình PowerShell người dùng gõ cú pháp như sau:

**python object\_size.py --image PATH --width X**

Giải thích:

*PATH* : đường dẫn nơi lưu hình ảnh và tên hình.

*X*: là số inch thực tế của một đồ vật nào đại diện trong hình.



Hình 75: Thực hiện cú pháp chạy chương trình

**Bước 4**: Kết quả sau khi nhấn Enter



Hình 76: Kết quả thực thi

Sau khi người dùng nhìn thấy kết quả trên màn hình, thì người dùng đã xác định được inch của đồ vật đầu tiên từ trái sang phải.

Người dùng muốn đo đồ vật kế tiếp thì nhấn phím 1 phím bất kỳ. Cho đến khi hết các đối tượng được nhận dạng trong hình.

Người dùng muốn thoát trong quá trình đo đồ vật thì phải thực hiện hết các lần đo bằng cách nhấn giữ 1 phím bất kỳ.