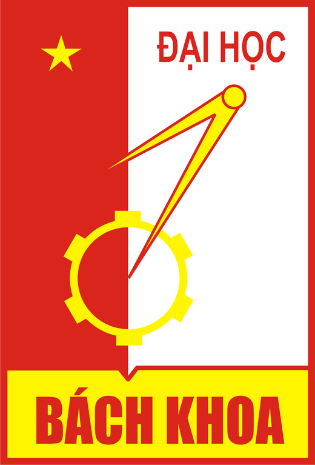
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN ĐIỆN TỬ - VIỄN THÔNG  
\*\*\*\*\* 🙠🕮🙢 \*\*\*\*\***



**BÁO CÁO ĐỒ ÁN I**

**ĐỀ TÀI: PHÂN LOẠI MÀU SẮC TRÊN UX/UI SỬ DỤNG THUẬT TOÁN KNN TRONG MACHINE LEARNING**

Giảng viên hướng dẫn**: TS. Nguyễn Huy Hoàng**

Thực hiện:

|  |  |
| --- | --- |
| **Họ và tên** | **MSSV** |
| **Lê Huy Hoàng** | **20182535** |

***Hà Nội, 6/2021***

**Mục Lục**

[**Danh mục hình ảnh i**](#_Toc79177305)

[**Danh mục bảng biểu ii**](#_Toc79177306)

[**TỪ VIẾT TẮT iii**](#_Toc79177307)

[**LỜI MỞ ĐẦU 1**](#_Toc79177308)

[**CHƯƠNG 1: Ý TƯỞNG THỰC HIỆN 2**](#_Toc79177309)

[**1.1. Lý do chọn đề tài 2**](#_Toc79177310)

[**1.2. Ứng dụng của sản phẩm 2**](#_Toc79177311)

[**CHƯƠNG 2: TỔNG QUAN VỀ HỆ THỐNG PHÂN BIỆT MÀU SẮC 3**](#_Toc79177312)

[**2.1. Khái quát hệ thống 3**](#_Toc79177313)

[**2.2. Yêu cầu chức năng 4**](#_Toc79177314)

[**2.3. Yêu cầu phi chức năng 4**](#_Toc79177315)

[**2.4. Những thuật toán được sử dụng trong hệ thống 4**](#_Toc79177316)

[***2.4.1. Giới thiệu sơ qua về xử lý ảnh* 4**](#_Toc79177317)

[**2.4.1.1. Điểm ảnh 5**](#_Toc79177318)

[**2.4.1.2. Mức xám của ảnh 5**](#_Toc79177319)

[**2.4.1.3. Các loại nhiễu trong ảnh 6**](#_Toc79177320)

[**2.4.1.4. Thay đổi kích thước của bức ảnh. 7**](#_Toc79177321)

[**2.4.1.5. Độ tương phản của ảnh 8**](#_Toc79177322)

[***2.4.2. Thuật toán tìm đặc trưng mức xám 3 kênh R,G,B của bức ảnh.* 9**](#_Toc79177323)

[**2.4.2.1. Tìm giá trị mức xám mà tại đó số lần xuất hiện mức xám đó [0,255] trên toàn bức ảnh là lớn nhất 10**](#_Toc79177324)

[**2.4.2.2. Dựa trên giá trị trung bình mức xám của bức ảnh tại mỗi kênh 10**](#_Toc79177325)

[***2.4.3. Tổng hợp lại các đặc trưng của mỗi bức ảnh để tạo data* 10**](#_Toc79177326)

[***2.4.4. Thuật toán KNN trong Machine Learning* 11**](#_Toc79177327)

[**2.4.4.1 Giới thiệu tổng quan về Machine Learning 11**](#_Toc79177328)

[**2.4.4.2. Giới thiệu qua về KNN 12**](#_Toc79177329)

[**2.4.4.3. Những vấn đề liên quan đến KNN 13**](#_Toc79177330)

[**2.4.4.4. Mô tả hoạt động thuật toán KNN 14**](#_Toc79177331)

[**2.4.4.5. Đánh giá độ chính xác của mô hình 15**](#_Toc79177332)

[**2.4.4.6. Ưu điểm và nhược điểm của KNN 15**](#_Toc79177333)

[**CHƯƠNG 3: THIẾT KẾ CHI TIẾT TỪNG KHỐI 16**](#_Toc79177334)

[**3.1. Khối lưu trữ dữ liệu 16**](#_Toc79177335)

[***3.1.1. Mục đích* 16**](#_Toc79177336)

[***3.1.2. Chức năng* 16**](#_Toc79177337)

[***3.1.3. Sơ đồ khối của khối lưu trữ* 16**](#_Toc79177338)

[***3.1.4. Lưu đồ thuật toán tìm giá trị mức xám mà tại đó số lần xuất hiện mức xám đó trên bức ảnh là lớn nhất* 17**](#_Toc79177339)

[***Hình 3.2. Lưu đồ thuật toán tìm giá trị mức xám mà tại đó số lần xuất hiện mức xám đó trên bức ảnh là lớn nhất* 17**](#_Toc79177340)

[***3.1.5. Lưu đồ thuật toán tìm giá trị trung bình mức xám của bức ảnh tại mỗi kênh* 18**](#_Toc79177341)

[**3.2. Khối đồ họa (UX/UI) 19**](#_Toc79177342)

[***3.2.1. Mục đích* 19**](#_Toc79177343)

[***3.2.2. Chức năng* 19**](#_Toc79177344)

[***3.2.3. Sơ đồ khối của khối đồ họa* 19**](#_Toc79177345)

[**3.3. Khối thuật toán KNN 20**](#_Toc79177346)

[***3.3.1. Mục đích* 20**](#_Toc79177347)

[***3.3.2. Chức năng* 20**](#_Toc79177348)

[***3.3.3. Sơ đồ khối* 20**](#_Toc79177349)

[**3.4. Lưu đồ thuật toán của thuật toán KNN 21**](#_Toc79177350)

[**CHƯƠNG 4: KIỂM THỬ HỆ THỐNG 22**](#_Toc79177351)

[**4.1.Một số dự đoán màu trên hệ thống UX/UI 22**](#_Toc79177352)

[***4.1.1. Dự đoán màu cho ảnh màu tím* 22**](#_Toc79177353)

[***4.1.2. Dự đoán màu xanh cho ảnh màu xanh* 23**](#_Toc79177354)

[**4.2. Tác động của môi trường bên ngoài đến màu sắc 24**](#_Toc79177355)

[***4.2.1. Kết quả thông qua hình vẽ* 24**](#_Toc79177356)

[***4.2.2. Kết quả thông qua bảng so sánh* 25**](#_Toc79177357)

[**4.2.2.1. Trong môi trường tối 25**](#_Toc79177358)

[**4.2.2.2. Trong môi trường bình thường 25**](#_Toc79177359)

[**KẾT LUẬN 27**](#_Toc79177360)

[**TÀI LIỆU THAM KHẢO 28**](#_Toc79177361)

# 

# Danh mục hình ảnh

[**Hình 2.1. Giao diện dành cho người dùng 3**](#_Toc79177285)

[**Hình 2.2. Ảnh đen trắng 5**](#_Toc79177286)

[**Hình 2.3. Ảnh RGB 6**](#_Toc79177287)

[**Hình 2.4. Ảnh có nhiễu muối tiêu 7**](#_Toc79177288)

[**Hình 2.5. Giảm kích thước bức ảnh 8**](#_Toc79177289)

[**Hình 2.6. Biểu đồ mức xám của bức ảnh tối 8**](#_Toc79177290)

[**Hình 2.7. Biểu đồ mức xám của bức ảnh sáng 9**](#_Toc79177291)

[**Hình 2.8. Mối quan hệ giữa AI, Machine Learning, và Deep Learning 11**](#_Toc79177292)

[**Hình 2.9. Thuật toán KNN 12**](#_Toc79177293)

[**Hình 2.10. Norm 1 và norm 2 trong không gian hai chiều 13**](#_Toc79177294)

[**Hình 2.11. Cách thức hoạt động của thuật toán KNN 14**](#_Toc79177295)

[**Hình 3.1. Sơ đồ khối của khối lưu trữ 16**](#_Toc79177296)

[**Hình 3.3. Lưu đồ thuật toán tìm giá trị trung bình mức xám 18**](#_Toc79177297)

[**Hình 3.4. Sơ đồ khối của khối đồ họa 19**](#_Toc79177298)

[**Hình 3.5. Sơ đồ khối của khối thuật toán KNN 20**](#_Toc79177299)

[**Hình 3.6. Lưu đồ thuật toán KNN 21**](#_Toc79177300)

[**Hình 4.1. Dự đoán màu cho ảnh màu tím 22**](#_Toc79177301)

[**Hình 4.2. Dự đoán màu cho ảnh màu tím 23**](#_Toc79177302)

[**Hình 4.3. Dự đoán màu cho ảnh màu xanh 23**](#_Toc79177303)

[**Hình 4.4. So sánh độ chính xác của thuật toán KNN trên những điều kiện bên ngoài tác động 24**](#_Toc79177304)

# Danh mục bảng biểu

[**Bảng 1: Kết quả phân loại màu sắc trong môi trường tối 25**](#_Toc79177268)

[**Bảng 2: Kết quả phân loại màu sắc trong môi trường bình thường 25**](#_Toc79177269)

[**Bảng 3: Kết quả phân loại màu sắc trong môi trường sáng 26**](#_Toc79177270)

# TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Từ Tiếng Anh** | **Nghĩa Tiếng Việt** |
| **AI** | **Artificial Intelligence** | **Trí tuệ nhân tạo** |
| **ML** | **Machine Learning** | **Máy học** |
| **KNN** | **K-nearest-neighbors** | **K láng giềng gần nhất** |
| **KD tree** | **K-dimensional tree** | **Cây K chiều** |
| **UX** | **User Experience** | **Trải nghiệm người dùng** |
| **UI** | **User Interface** | **Giao diện người dùng** |
| **CSV** | **Comma separated values** | **Các giá trị được phân tách bởi dấu phẩy** |

# LỜI MỞ ĐẦU

Cuộc sống ngày càng phát triển,công nghệ thông tin đã trở thành một phần không thể thiếu đối với kỷ nguyên 4.0 hiện nay, không thể không nhắc đến đó chính là trong thời đại này. Trí tuệ nhân tạo đã len lỏi một phần nào đó vào trong cuộc sống mỗi người chung ta mặc dù chúng ta không nhận ra được. Machine Learning là một phần của trí tuệ nhân tạo cho phép máy có thể học được cách suy nghĩ như một con người.

Hiện nay có rất nhiều ứng dụng liên quan đến Machine Learning, gồm dự đoán giá nhà, phân loại nam hay nữ, phân loại màu sắc,v.v. Tất cả chúng đều dựa trên những thuật toán cơ bản trong Machine Learning để dự đoán. Qua đó em xin đưa ra một đề tài về phân loại màu sắc sử dụng thuật toán KNN trong Machine Learning.

Trong quá trình thự c hi ện đồ án 1, dù r ất cố gắng nhưng do vố n kiến thứ c hạn hẹp nên không thể tránh khỏ i nhữ ng sai sót. Em mong chờ nhận được những đóng góp, phê bình, chia sẻ của thầy có thể sẽ hoàn thiện bản thân hơn trên quá trình em còn trên ghế nhà trường.

Em xin chân thành cảm ơn thầy!

# CHƯƠNG 1: Ý TƯỞNG THỰC HIỆN

## 1.1. Lý do chọn đề tài

Hiện nay, AI nói chung và xử lý ảnh đang là lĩnh vực đang được cả xã hội quan tâm và được đầu tư. Có rất nhiều ứng dụng được sử dụng bên ngoài thực tế. Hôm nay em xin trình bày về ứng dụng phân biệt màu sắc trên UX/UI dựa trên thuật toán KNN trong Machine Learning. Việc chọn project này giúp cho em hiểu rõ về lý thuyết và thực hành hơn. Và quan trọng hơn hết là nó phù hợp với kiến thức của em cũng như thời gian để hoàn thiện sản phẩm là khả thi.

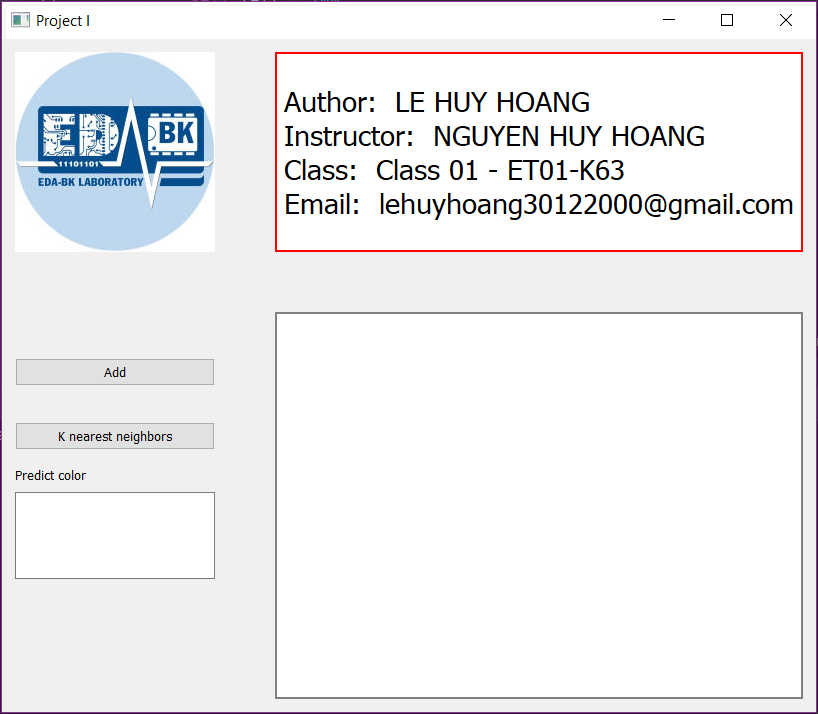
## 1.2. Ứng dụng của sản phẩm

Ứng dụng có thể phát triển thành phân biệt màu sắc của quần áo, hoặc phát triển lên là truy xuất người trên camera dựa theo màu sắc quần áo của người đó là gì?

# CHƯƠNG 2: TỔNG QUAN VỀ HỆ THỐNG PHÂN BIỆT MÀU SẮC

## 2.1. Khái quát hệ thống

Sản phẩm này sử dụng các bức hình để dự đoán màu sắc. Với đầu vào là một bức ảnh có kích thước bất kì và đầu ra là màu sắc chính của bức ảnh đó là cam, vàng, đỏ, đen, trắng,.. Đối với ảnh đầu vào có kích thước bất kì thì chúng ta cần phải đưa nó về cùng một kích thước để nó hiển thị trên UX/UI một cách chính xác nhất.

****

******Hình 2.1. Giao diện dành cho người dùng**

Hình 2.1 đang mô tả về giao diện UX/UI gồm những cái gì và mục đích của nó để làm gì?

* Bức hình và các chữ trong hình chữ nhật màu đỏ nói về người hướng dẫn đồ án 1 của em.
* Nút Add : Khi add một bức hình từ các file trong máy tính thì nó sẽ hiển thị bức ảnh đấy lên màn hình màu trắng.
* Tiếp theo nút K-nearest-neighbor: Dự đoán bức hình này màu chủ đạo của nó là gì ? và sẽ in màu đó ra phần predict\_color.

## 2.2. Yêu cầu chức năng

* Hệ thống phải hoạt động đủ nhanh để đưa ra dự đoán một cách nhanh nhất nhưng cũng phải có sự chính xác nhất định (có thể không đạt được 100%).
* Tập dữ liệu phải đủ các tính chất để khi dự đoán thì khả năng chính xác sẽ được tăng cao hơn.

## 2.3. Yêu cầu phi chức năng

* Giao diện phải dễ dàng đối với người sử dụng.
* Sử dụng thuật toán KNN trong Machine Learning.
* Để có thể sử dụng được thuật toán KNN ta cần phải có một bộ dữ liệu đủ lớn.
* Sử dụng các thuật toán về xử lý ảnh để trích xuất đặc trưng của bức ảnh đó.

## 2.4. Những thuật toán được sử dụng trong hệ thống

### *2.4.1. Giới thiệu sơ qua về xử lý ảnh*

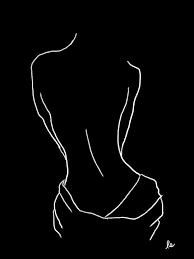
Trước khi nói về những thuật toán này thì em xin giới thiệu sơ qua những lý thuyết cần dùng trong quá trình về xử lý ảnh. Xử lý ảnh những năm gần đây có những ứng dụng hay và có tính sáng tạo cao và có thể phát triển mạnh trong thời gian sắp tới.

#### 2.4.1.1. Điểm ảnh

Điểm ảnh hay còn gọi là Pixel là một phần tử nhỏ nhất của ảnh số với một mức màu nhất định. Kích thước và khoảng cách giữa đó được chọn thích hợp sao cho mắt người cảm nhận bức ảnh số đó là liên tục. Mỗi một phần tử trong một ma trận điểm ảnh sẽ tạo ra 1 bức ảnh số.

#### 2.4.1.2. Mức xám của ảnh

* Định nghĩa : Mức xám của một điểm ảnh là cường độ sáng của điểm ảnh đó với giá trị là một số nguyên.
* Các thang mức xám được sử dụng phổ biến: 2,8,16,32,64,128,256 (mức 256 là phổ biến nhất) ứng với số bit được biểu diễn sẽ là 1,3,4,5,6,7,8 bit tương ứng.
* Ảnh nhị phân: là ảnh biểu diễn 1 bit gồm 2 màu đó là màu đen và màu trắng, với giá trị màu đen là bit giá trị là 0 và màu trắng giá trị là 1.



**Hình 2.2. Ảnh đen trắng**

* Ảnh xám: là ảnh có mức xám có giá trị từ [0,255] với mức xám 0 là tối nhất còn mức xám 255 là sáng nhất.
* Ảnh màu: Phổ biến nhất vẫn là 3 kênh màu là Red, Green, Blue. Chúng ta có thể phối hợp mức xám của từng kênh sau đó kết hợp lại để ra được màu khác. Khi đó tổng giá trị màu sẽ là triệu màu.



**Hình 2.3. Ảnh RGB**

* Ngoài ra còn có các ảnh màu khác như HSV, CMYK.
* Công thức chuyển đổi ảnh từ RGB sang ảnh Gray:

I(x,y) =

* Với I(x,y) : mức xám tại vị trí (x,y)
* Tương tự như thế với R(x,y) và G(x,y) , B(x,y)

#### 2.4.1.3. Các loại nhiễu trong ảnh

* Nhiễu Gauss là mô hình phổ biến để xấp xỉ nhiễu trong nhiều ứng dụng khác nhau. Mật độ phân bố xác suất của nhiễu là hàm Gauss, được đặc trưng bởi giá trị trung bình , và phương sai là .
* Nhiễu muối tiêu: có đặc trưng bởi một điểm ảnh có giá trị mức xám khác biệt rất lớn so với những điểm lân cận.

****

**Hình 2.4. Ảnh có nhiễu muối tiêu**

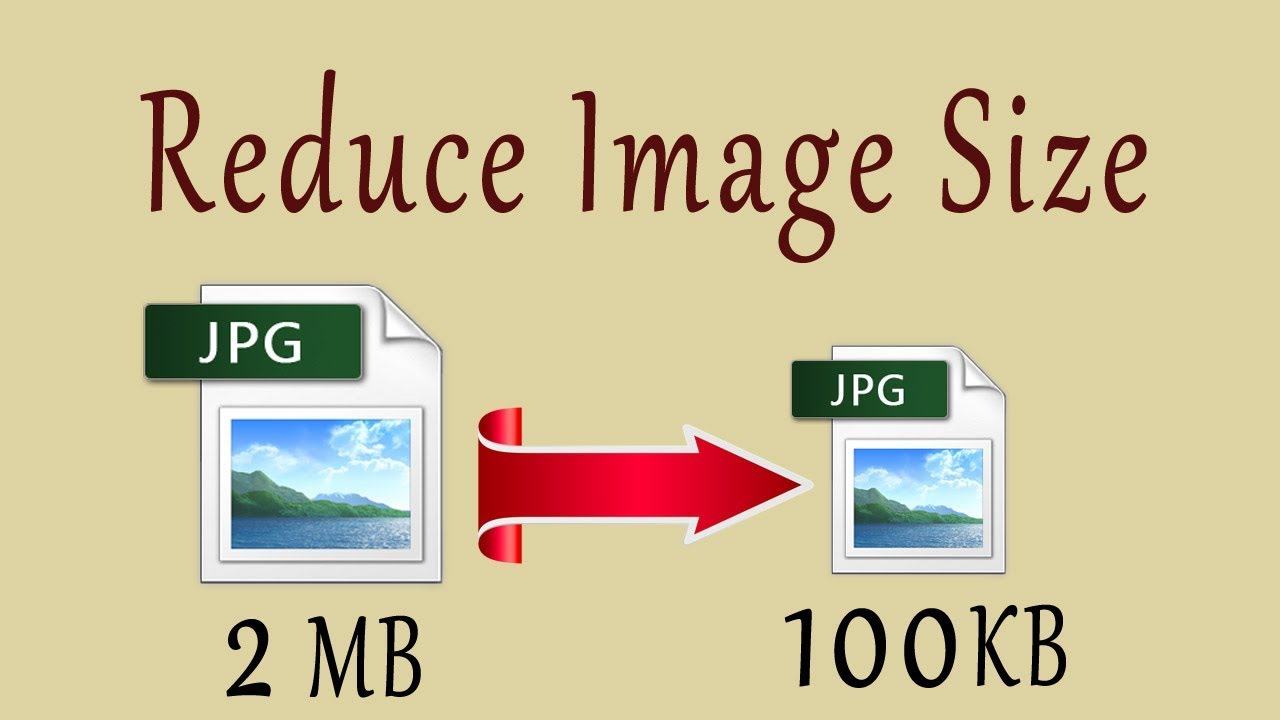
* Nhiễu trắng
* Nhiễu Poission
* Đa phần ta sẽ sử dụng một số bộ lọc để loại bỏ nhiễu ,mục đích của nó làm tăng cường chất lượng hình ảnh lên đồng thời xóa bỏ đi nhiễu làm chất lượng ảnh không tốt thông qua một số bộ lọc Median Blur (lọc trung vị), bộ lọc Gauss Blur và Mean Blur (lọc trung bình).

#### 2.4.1.4. Thay đổi kích thước của bức ảnh.

Giả sử ta có bức ảnh có kích thước ban đầu là (a,b) với a là chiều cao của bức ảnh và b là chiều rộng của bức ảnh. Ta có thể thay đổi kích thước bức ảnh nhưng thông tin trên bức ảnh vẫn giữ nguyên.

Mục đích của việc thay đổi bức ảnh nhỏ đi có những ứng dụng sau:

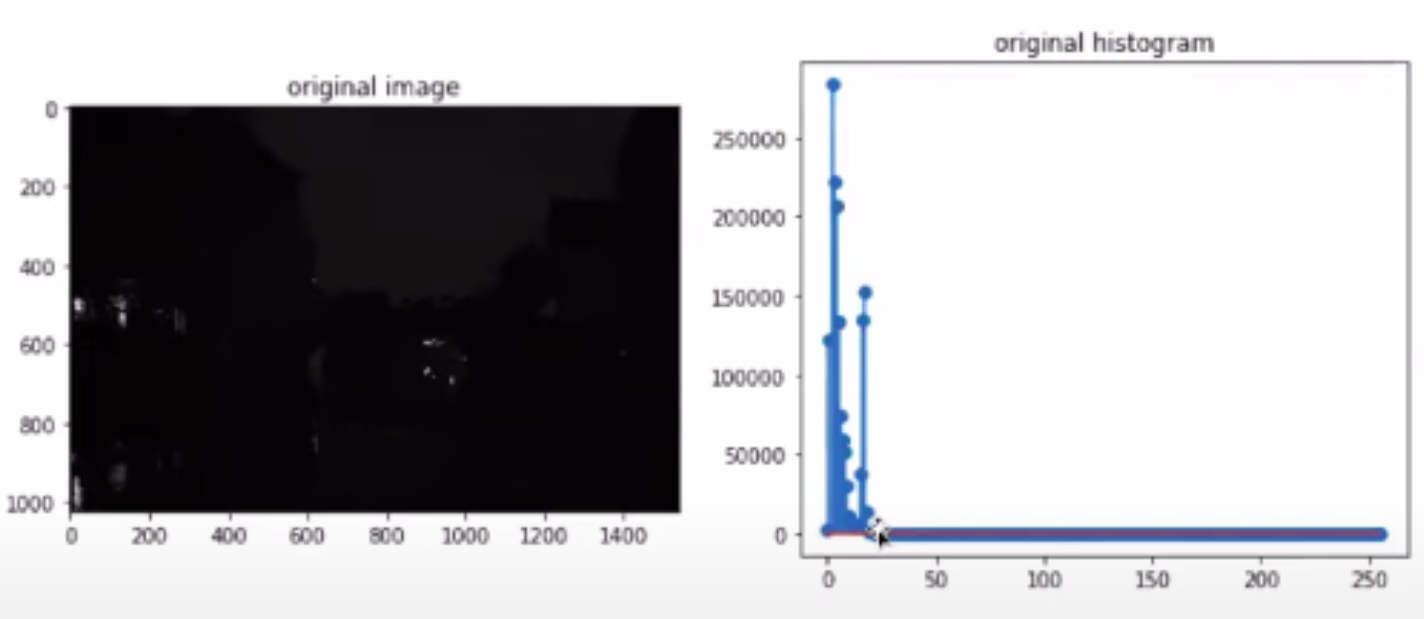
* Giảm thời gian tính toán khi đưa bức hình đó vào một thuật toán nào đó. Ví dụ trong các mạng Neural Network việc đầu vào bức ảnh có kích thước quá lớn có thể dẫn đến việc train mô hình mạng quá lâu mà cũng chưa chắc hiệu quả hơn việc đưa đầu vào bức ảnh có kích thước nhỏ.
* Đặc biệt giảm thời gian chạy trong các mô hình REAL-TIME hơn việc để bức ảnh giữ nguyên.
* Tốn ít chi phí hơn.



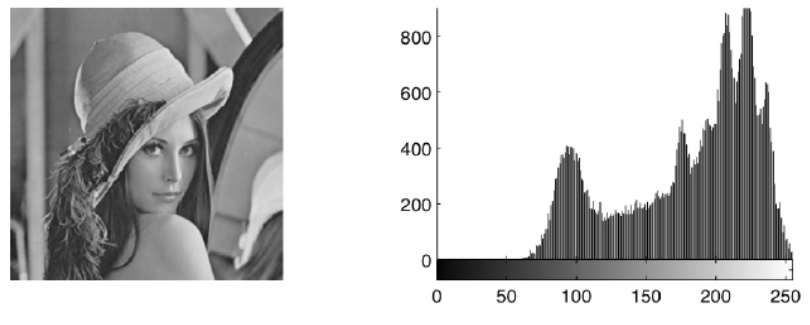
**Hình 2.5. Giảm kích thước bức ảnh**

#### 2.4.1.5. Độ tương phản của ảnh

Xét một bức ảnh xám thì thấy được rằng khi một bức ảnh khá tối, rất tối thì giá trị mức xám của những điểm đó sẽ tập trung về giá trị 0 nhiều hơn so với giá trị 255. Tương tự như vậy đối với một bức ảnh khá sáng, rõ nét.

****

**Hình 2.6. Biểu đồ mức xám của bức ảnh tối**

****

**Hình 2.7. Biểu đồ mức xám của bức ảnh sáng**

Từ đó ta có thể thấy rằng việc một bức ảnh sáng hay tối phụ thuộc rất nhiều số điểm ảnh trong bức ảnh đó. Giá trị của điểm ảnh đó gần về 0 thì bức ảnh đó càng tối, còn giá trị của điểm ảnh đó gần với 255 thì bức ảnh đó càng sáng.

* Từ những điều trên, nó có thể áp dụng tương tự đối với ảnh màu. Ảnh màu mà cả 3 kênh Red, Green, Blue mà số lượng mức xám gần 0 thì có thể đoán ra bức ảnh đó màu đen. Tương tự với những màu còn lại. Trong phần tiếp theo sẽ nói kĩ hơn về những thuật toán được sử dụng.

### *2.4.2. Thuật toán tìm đặc trưng mức xám 3 kênh R,G,B của bức ảnh.*

Việc dự đoán màu sắc dựa vào các thuật toán trong Machine Learning cần có những điều phải chú ý sau:

* Do bức ảnh gồm rất nhiều pixels gộp thành nên nó sẽ rất lớn nếu đưa vào các thuật toán trong máy học . Giả sử mỗi một đặc trưng ứng với 1 pixel thì những bức ảnh có kích thước lớn sẽ tính toán rất lâu. Mà thuật toán KNN không phù hợp với những bài toán phức tạp và không có tính REAL-TIME.
* Phải tìm ra ba đặc trưng ứng với ba kênh Red, Green, Blue để khi có được đặc trưng đó ta có thể dự đoán được bức ảnh đó sẽ có màu vàng, đỏ, xanh, vàng hay cam, v.v..

Có hai cách để đưa 3 đặc trưng:

* Tìm giá trị mức xám mà tại đó số lần xuất hiện mức xám đó [0,255] trên toàn bức ảnh là lớn nhất.
* Dựa trên giá trị trung bình mức xám của bức ảnh.

#### 2.4.2.1. Tìm giá trị mức xám mà tại đó số lần xuất hiện mức xám đó [0,255] trên toàn bức ảnh là lớn nhất

* Bước 1: Chọn bức ảnh RGB , không được phép lấy bức ảnh xám
* Bước 2: Tìm số lần xuất hiện giá trị mức xám từ [0,255] của cả 3 kênh Red, Green, Blue.
* Bước 3: Lấy giá trị mức xám mà số lần xuất hiện tại mức xám đó là giá trị lớn nhất.

#### 2.4.2.2. Dựa trên giá trị trung bình mức xám của bức ảnh tại mỗi kênh

* Hai bước trên đều giống với phần trên.
* Bước 3: Tìm giá trị mức xám trung bình của bức ảnh đó.
* Dễ dàng nhận thấy rằng thời gian để ra được kết quả của việc tìm giá trị trung bình mức xám sẽ lâu hơn so với việc tìm mức xám mà tại đó giá trị của nó trên bức ảnh là lớn nhất.

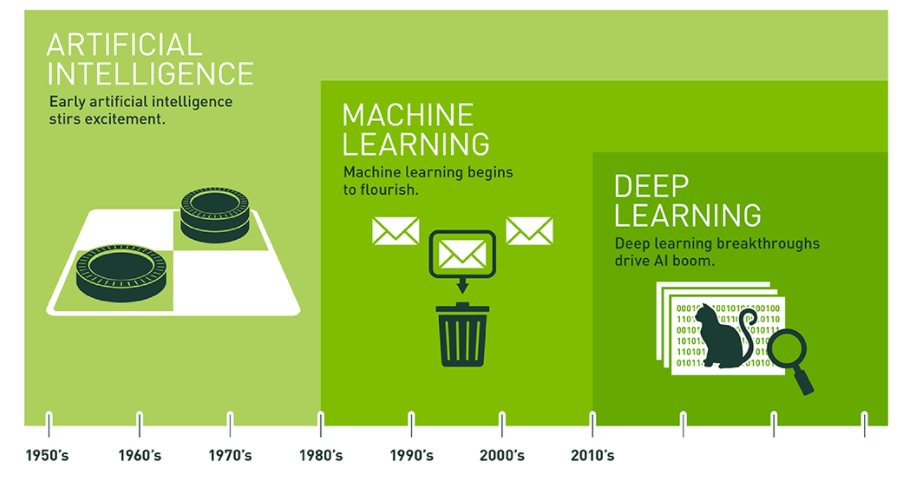
### *2.4.3. Tổng hợp lại các đặc trưng của mỗi bức ảnh để tạo data*

Việc lưu dữ liệu vào 1 file để thuận tiện hơn cho quá trình thực hiện thuật toán KNN trong Machine Learning. Và đặc trưng của nó là giá trị mức xám tại 3 kênh Red, Green, Blue và cuối cùng sẽ là màu sắc của bức ảnh đó là gì?. Dữ liệu về các đặc trưng của bức ảnh càng nhiều thì nó sẽ càng phân biệt rõ rệt màu nào với màu nào hơn so với tập đặc trưng quá ít. Ở đây em sẽ sử dụng file csv để đơn giản và Python hỗ trợ 1 thư viện về csv để dễ dàng cho việc lập trình.

### *2.4.4. Thuật toán KNN trong Machine Learning*

#### 2.4.4.1 Giới thiệu tổng quan về Machine Learning

Trí tuệ nhân tạo đang vô cùng phổ biến trong kỷ nguyên hiện nay bởi những ứng dụng của nó và nghe cái tên cũng khá hứng thú cho người nghe và đọc. Trí Tuệ Nhân Tạo đã và đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không biết. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, chỉ là một vài trong hàng ngàn những ứng dụng của AI/Machine Learning. Machine Learning là một tập con của Trí Tuệ Nhân Tạo.

****

**Hình 2.8. Mối quan hệ giữa AI, Machine Learning, và Deep Learning**

Ở đây ta sẽ phân những thuật toán dựa trên phương thức học của Machine Learning:

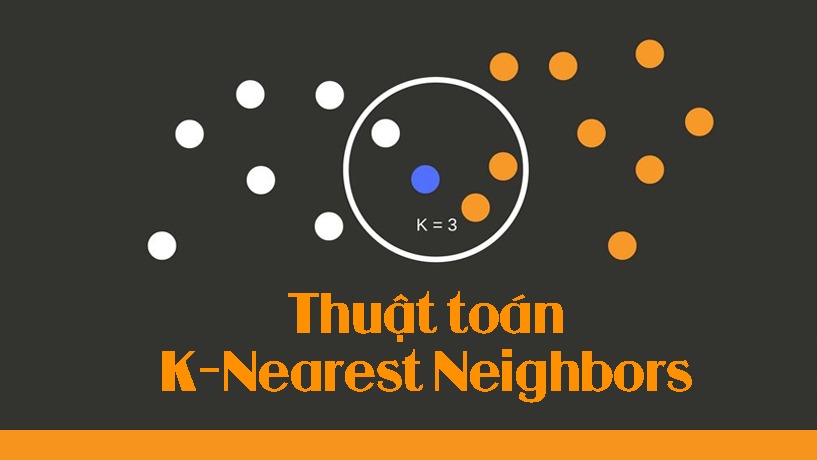
* Supervised Learning : Học có giám sát
* Regression (Bài toán hồi quy)
* Classification (Bài toán phân loại)
* Unsupervised Learning : Học không có giám sát
* Clustering (Bài toán phân nhóm)
* Association
* Semi-Supervised Learning : Học bán giám sát
* Reinforcement Learning : Học củng cố

#### 2.4.4.2. Giới thiệu qua về KNN

Sau khi hoàn thành đồ án 1, em thấy được rằng thuật toán này rất phù hợp dành cho những người muốn bắt đầu học Machine Learning. Lý do vì thuật toán này dễ hiểu và đặc biệt hơn là kiến thức về toán không nặng.

Thuật toán KNN có thể áp dụng cho bài toán hồi quy (regression) và bài toán phân loại (classification). Với những ứng dụng có thể được sử dụng như sau:

* Dự đoán giá nhà (bài toán hồi quy)
* Phân loại màu sắc (bài toán phân loại, cụ thể phân loại nhiều lớp)
* Phân loại các tài liệu văn bản thuộc tài liệu giải trí,tài liệu khoa học hay tài liệu thể thao,vv.(kết hợp giữa Bag-of-Words và thuật toán KNN)



**Hình 2.9. Thuật toán KNN**

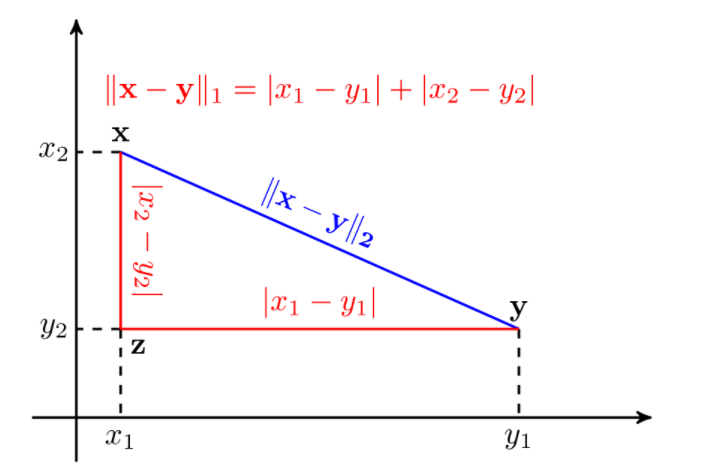
Thuật toán KNN (K-nearest-neighbor) được hiểu là K láng giếng gần nhất hay tìm K điểm gần nhất của dữ liệu cần dự đoán lớp. Đây là thuật toán “lazy learning” bởi vì nó không học được cái gì từ tập huấn luyện. KNN còn được gọi là thuật toán Memory-based learning , hiểu đơn giản thì nó học dựa trên bộ nhớ, bộ nhớ càng chứa nhiều dữ liệu thì khả năng dự đoán của KNN sẽ trở nên chính xác hơn.

#### 2.4.4.3. Những vấn đề liên quan đến KNN

Từ những điều trên em đặt ra câu hỏi dự đoán nhãn dựa trên những điểm gần nhất như thế nào?. Và chúng có thật sự tốt cho những bài toán có tập dữ liệu lớn không?

* Đầu tiên, ta sẽ tìm những điểm gần nhất dựa vào các đặc trưng của điểm cần dự đoán và các đặc trưng của tập huấn luyện thông qua các chuẩn norm. Ta có công thức tổng quát có thể áp dụng cho toàn bộ chuẩn norm được biểu diễn bởi công thức dưới đây:

|  |  |  |
| --- | --- | --- |
| Với p = 1: |  | (1) |
| Với p = 2: |  | (2) |
| Với p = |  | (3) |

****

**Hình 2.10. Norm 1 và norm 2 trong không gian hai chiều**

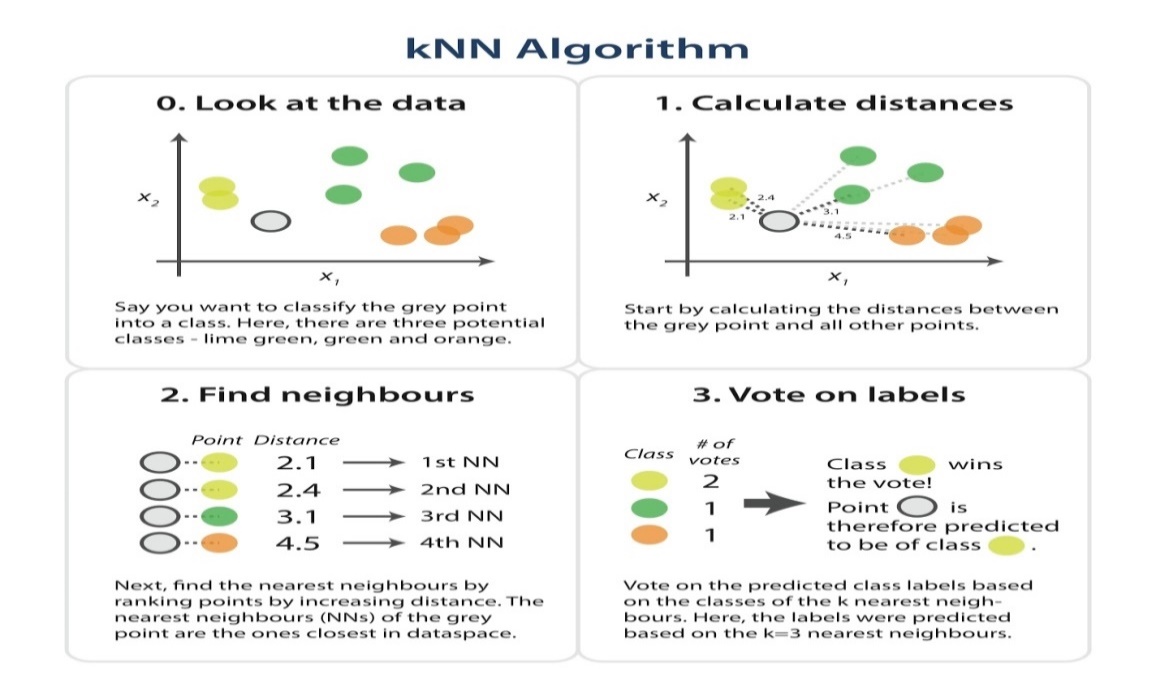
Thông thường trong các bài toán tính khoảng cách thì người ta sử dụng chuẩn norm 2. Nên em sẽ sử dụng norm 2 để sử dụng cho việc dự đoán đầu ra của dữ liệu cần dự đoán.

* Để giảm thời gian dự đoán đầu ra dựa trên thuật toán KNN có rất nhiều phương pháp, bao gồm Kd tree và Ball Tree.

#### 2.4.4.4. Mô tả hoạt động thuật toán KNN

Các bước thực hiện thuật toán KNN gồm những bước sau:

* Bước 1: Phân chia dữ liệu ( tập huấn luyện và tập kiểm tra , có thể cần đến tập kiểm thử).
* Bước 2: Tính toán khoảng cách giữa các đặc trưng dữ liệu mới với từng dữ liệu điểm huấn luyện ( 1 điểm ở đây bao gồm n features và 1 label ).
* Bước 3: Lấy K điểm gần với điểm dữ liệu mới.
* Bước 4 : Dự đoán nhãn dựa trên K điểm huấn luyện ( nhãn nào có nhiều hơn trong K điểm thì khả năng cao nhãn dự đoán sẽ là nhãn đó )**.**

****

**Hình 2.11. Cách thức hoạt động của thuật toán KNN**

Chú ý: Khi tìm K láng giềng gần nó nhất, có thể xảy ra khả năng các nhãn sẽ bằng nhau trong K điểm đó thì lúc đó ta sẽ xét xem tổng khoảng cách của nhãn đó đến nhãn cần dự đoán, nhãn cần dự đoán khả năng sẽ thuộc nhãn mà có tổng khoảng cách của nhãn đó đến nhãn cần dự đoán là giá trị nhỏ nhất.

#### 2.4.4.5. Đánh giá độ chính xác của mô hình

Do bài toán đặt ra ở đây là phân loại màu sắc dựa trên KNN nên sẽ dùng accuracy cho bài toán phân loại. Công thức sau sẽ biểu diễn độ chính xác của mô hình:

* True\_predict\_test : Số lượng nhãn dự đoán đúng trong tập test.
* Test\_data: Số lượng nhãn trong tập test.

#### 2.4.4.6. Ưu điểm và nhược điểm của KNN

* Ưu điểm:
* Thuật toán đơn giản, dễ hiểu cho người mới bắt đầu.
* Không học gì từ tập dữ liệu so với các thuật toán khác trong ML.
* Phù hợp với những bài toán không cần dữ liệu quá lớn (vừa đủ là được).
* Có thể giải quyết cả 2 bài toán là phân loại và hồi quy.
* Nhược điểm:
* Lưu trữ rất nhiều khoảng cách do tập huấn luyện lớn => Có thể dẫn đến tràn RAM.
* Tính toán nhiều lần => Thời gian để ra được dự đoán đầu ra khá lâu nếu dữ liệu huấn luyện lớn.
* Không phù hợp với bài toán REAL TIME.

# CHƯƠNG 3: THIẾT KẾ CHI TIẾT TỪNG KHỐI

## 3.1. Khối lưu trữ dữ liệu

### *3.1.1. Mục đích*

Khối này được sử dụng để biến đổi những bức ảnh có kích thước bất kỳ về những điểm dữ liệu, mỗi điểm dữ liệu có ba đặc trưng là ba kênh và nhãn của ảnh đó là màu sắc. Việc làm này giúp cho thuật toán KNN hoạt động nhanh hơn rất nhiều so với việc đưa tất cả pixels của bức ảnh vào thuật toán KNN.

Việc gom lại vào trong một file sẽ giúp chúng ta quản lý tệp dữ liệu dễ hơn và còn biết được các đặc trưng có bị lỗi trong quá trình lấy đặc trưng.

### *3.1.2. Chức năng*

* Một điểm dữ liệu gồm có ba đặc trưng là kênh Red, kênh Green, kênh Blue và giá trị cuối cùng lưu trữ màu sắc của bức hình đó.
* Lưu trữ dữ liệu dưới dạng tệp có đuôi .csv.
* Thuận tiện cho việc sử dụng thuật toán.

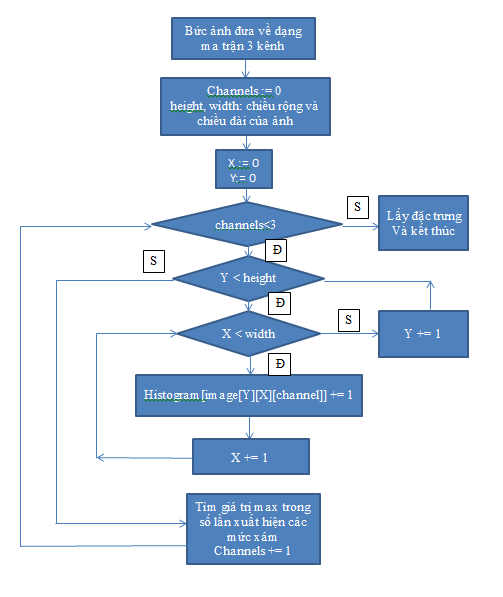
### *3.1.3. Sơ đồ khối của khối lưu trữ*

****

**Hình 3.1. Sơ đồ khối của khối lưu trữ**

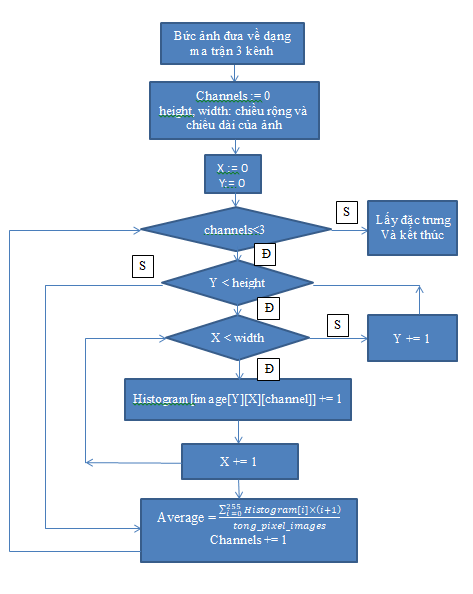
Tiếp theo lấy đặc trưng để gửi vào tập dữ liệu dạng đuôi .csv cần sử dụng một trong hai thuật toán đã được đề cập ở phần 2.4.2.1 và 2.4.2.2 đã được nói ở trong Chương 2. Qua đây em xin trình bày về cả 2 lưu đồ thuật toán đối với hai thuật toán trên.

### *3.1.4. Lưu đồ thuật toán tìm giá trị mức xám mà tại đó số lần xuất hiện mức xám đó trên bức ảnh là lớn nhất*

****

### *Hình 3.2. Lưu đồ thuật toán tìm giá trị mức xám mà tại đó số lần xuất hiện mức xám đó trên bức ảnh là lớn nhất*

### *3.1.5. Lưu đồ thuật toán tìm giá trị trung bình mức xám của bức ảnh tại mỗi kênh*



**Hình 3.3.** **Lưu đồ thuật toán tìm giá trị trung bình mức xám**

## 3.2. Khối đồ họa (UX/UI)

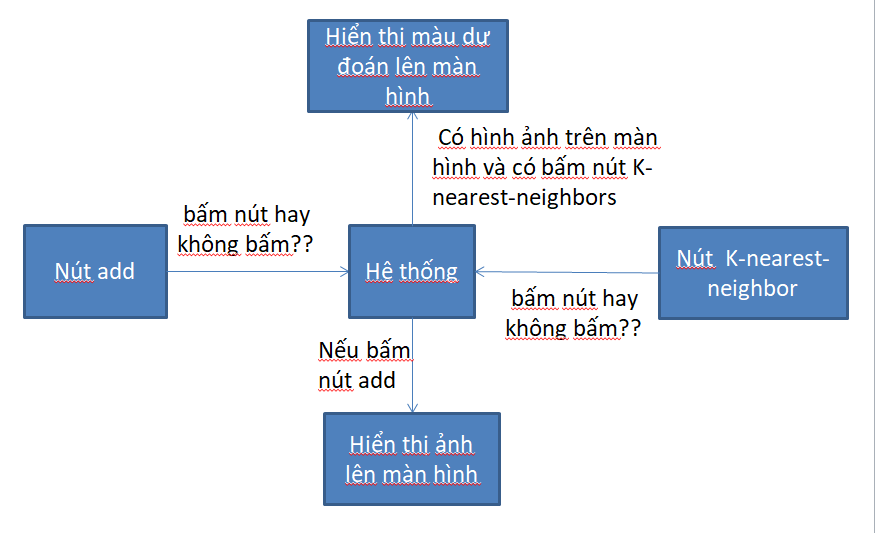
### *3.2.1. Mục đích*

Khối này để giúp cho người dùng thao tác với chương trình mà không cần biết bên trong có những cái gì và chỉ cần biết rằng hệ thống này dùng để nhận diện màu sắc.

### *3.2.2. Chức năng*

* Khi bấm nút Add trong chương trình thì nó sẽ xuất hiện màn hình xuất hiện những bức ảnh, lúc này ta sẽ chọn một bức ảnh bất kì rồi bấm nút Open và nó sẽ hiện lên bức ảnh mình đã chọn lên màn hình của chương trình.
* Sau đó bấm nút K-nearesrt-neighbors để dự đoán màu sắc của hình vừa chọn và đợi một chút thời gian để nó hiện lên kết quả .
* Người dùng dễ dàng sử dụng chương trình mà không gặp chút khó khăn.

### *3.2.3. Sơ đồ khối của khối đồ họa*

****

**Hình 3.4. Sơ đồ khối của khối đồ họa**

## 3.3. Khối thuật toán KNN

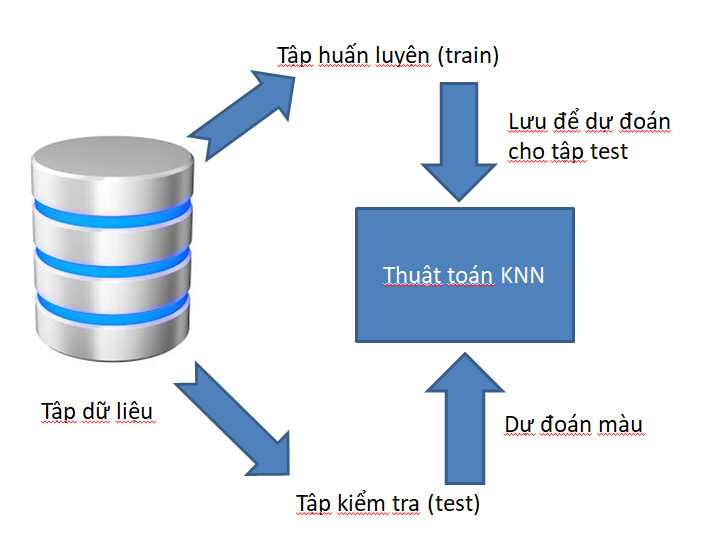
### *3.3.1. Mục đích*

Khối này có mục đích là dự đoán màu cho bức ảnh đầu với ba đặc trưng là kênh Red, kênh Green, kênh Blue và nhãn là 1 giá trị màu sắc (ví dụ : xanh, đỏ, vàng, tím, trắng, v.v).

### *3.3.2. Chức năng*

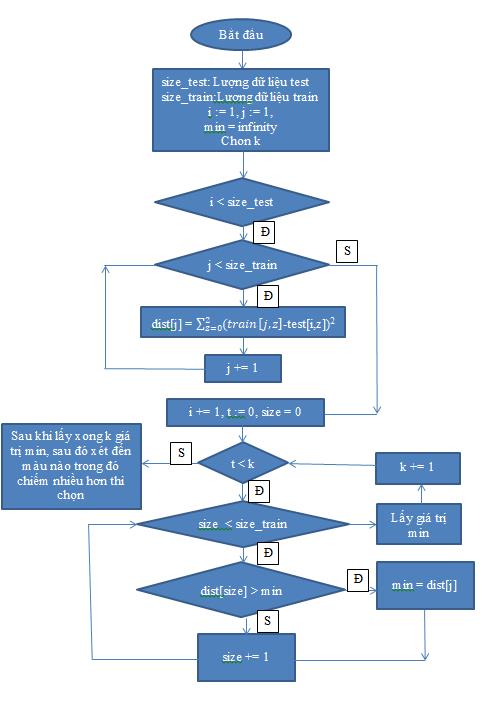
* Một phần được sử dụng làm tập huấn luyện và một phần được sử dụng làm tập kiểm tra để kiểm tra độ chính xác của mô hình trước khi đưa vào thực tế.
* Dự đoán màu.
* Cho biết tỷ lệ tập kiểm tra và tập huyến luyện để có thể chỉnh sửa một cách hợp lý.
* Biết được mô hình đó tốt hay xấu giữa trên độ chính xác.

### *3.3.3. Sơ đồ khối*

****

**Hình 3.5. Sơ đồ khối của khối thuật toán KNN**

### 3.4. Lưu đồ thuật toán của thuật toán KNN

****

**Hình 3.6. Lưu đồ thuật toán KNN**

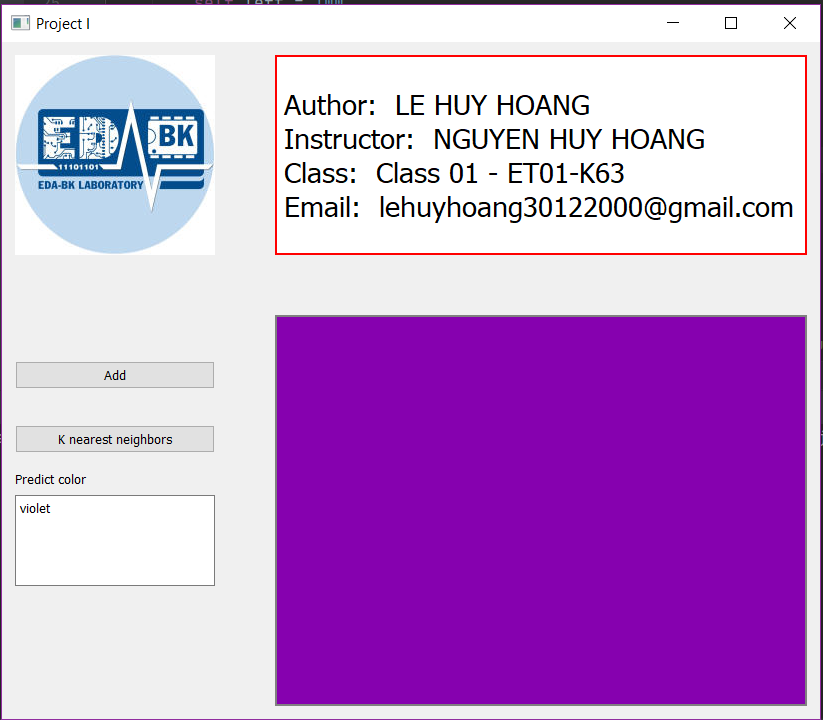
# CHƯƠNG 4: KIỂM THỬ HỆ THỐNG

Từ những lý thuyết đã được nêu lên ở chương 3 và chương 4, đến chương này chúng ta sẽ xem những lý thuyết ở trên đúng hay sai. Việc kiểm thử hệ thống là vô cùng quan trọng, quyết định đến sự thành công hay thất bại của sản phẩm. Nó cho em biết được rằng hệ thống của mình có hoạt động chính xác cho nhiều bức ảnh hay không và còn giúp em thêm một số tính năng của hệ thống.

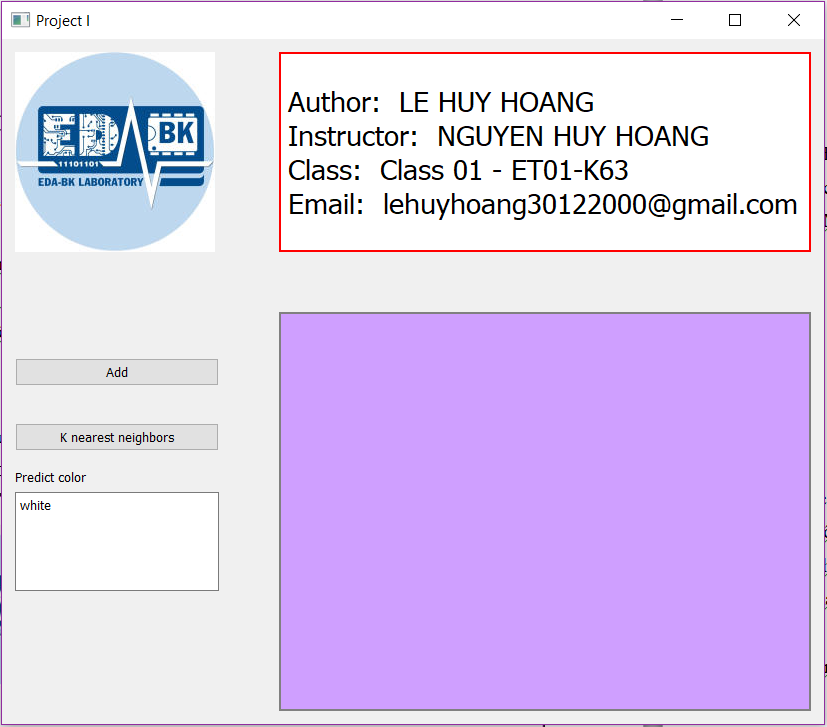
## 4.1.Một số dự đoán màu trên hệ thống UX/UI

Em sẽ lấy một số màu như tím, xanh để làm ví dụ cho việc dự đoán màu

### *4.1.1. Dự đoán màu cho ảnh màu tím*

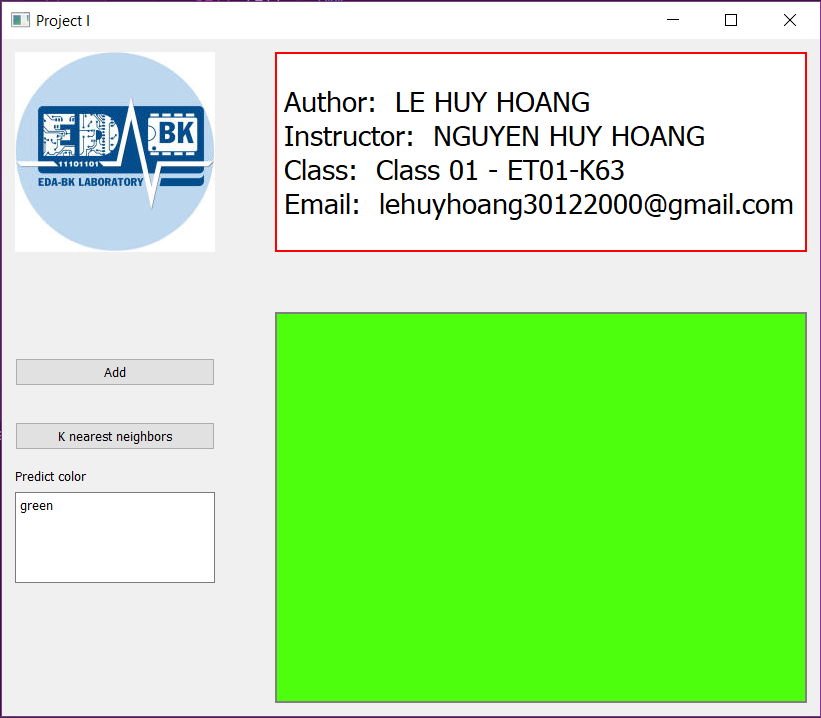
****

**Hình 4.1. Dự đoán màu cho ảnh màu tím**

****

**Hình 4.2. Dự đoán màu cho ảnh màu tím**

### *4.1.2. Dự đoán màu xanh cho ảnh màu xanh*

****

**Hình 4.3. Dự đoán màu cho ảnh màu xanh**

Từ những kết quả trên em rút ra được kết luận:

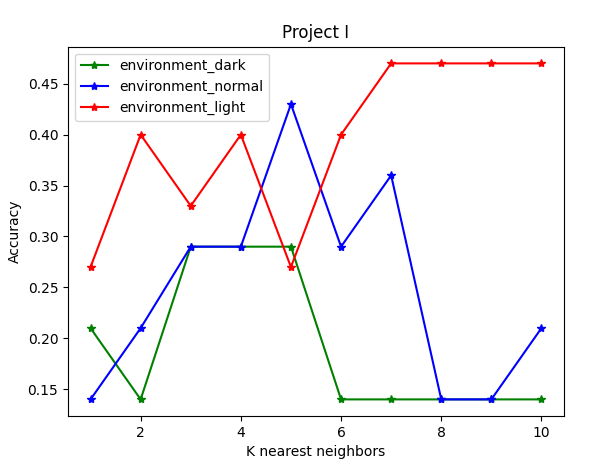
* Không nên lấy quá nhiều dữ liệu huấn luyện cho thuật toán KNN dẫn đến việc nó tính toán rất lâu
* Thuật toán KNN học dựa trên bộ nhớ nên khi bộ nhớ không có đầy đủ các tính chất của màu như đậm, nhạt, bình thường thì khả năng bị dự đoán nhầm màu là có thể xảy ra.

## 4.2. Tác động của môi trường bên ngoài đến màu sắc

### *4.2.1. Kết quả thông qua hình vẽ*

Những bức hình mà em sử dụng ở trên đều không có sự tác động của môi trường bên ngoài đến màu sắc điều đó dẫn đến độ chính xác cũng cao. Giả sử trong thực tế không bao giờ có chuyện bức hình nó chất lượng tốt như trên nên em sẽ thử kiểm tra thuật toán KNN liệu có hoạt động tốt nếu có các tác động bên ngoài như ánh sáng trong điều kiện ngoài trời, tối và bình thường. Ở đây em sẽ sử dụng bức ảnh quần áo và thêm các điều kiện trên.

Em sẽ chọn K từ 1 đến 10 để dự đoán liệu tại K bằng mấy thì độ chính xác ở K nào sẽ đạt giá trị cao nhất.

****

**Hình 4.4. So sánh độ chính xác của thuật toán KNN trên những điều kiện bên ngoài tác động**

Nhận xét :

* Ta thấy độ chính xác khi chụp ảnh ở ngoài trời có tỷ lệ chính xác cao hơn so với chụp ảnh trong điều kiện bình thường và tối.
* Độ chính xác không cao có thể do tập ảnh train còn quá ít đã dẫn đến tập ảnh train không có đầy đủ các đặc trưng để dự đoán, ngoài ra còn yếu tố tác động như môi trường.
* Nếu muốn tốt hơn về chính xác thì nên chọn màu áo, màu quần phải đồng với phong cảnh mình đang chụp.

### *4.2.2. Kết quả thông qua bảng so sánh*

#### 4.2.2.1. Trong môi trường tối

Bảng 1: Kết quả phân loại màu sắc trong môi trường tối

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **K** | **Black** | **Blue** | **Red** | **Orange** | **Yellow** | **White** | **Green** |
| **1** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **50%** |
| **2** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **0%** |
| **3** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **100%** |
| **4** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **100%** |
| **5** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **100%** |
| **6** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **0%** |
| **7** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **0%** |
| **8** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **0%** |
| **9** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **0%** |
| **10** | **100%** | **0%** | **100%** | **0%** | **0%** | **0%** | **0%** |

#### 4.2.2.2. Trong môi trường bình thường

Bảng 2: Kết quả phân loại màu sắc trong môi trường bình thường

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **K** | **Black** | **Blue** | **Red** | **Orange** | **Yellow** | **White** | **Green** |
| **1** | **100%** | **50%** | **0%** | **0%** | **0%** | **0%** | **0%** |
| **2** | **100%** | **100%** | **0%** | **0%** | **0%** | **0%** | **0%** |
| **3** | **100%** | **100%** | **0%** | **0%** | **0%** | **0%** | **50%** |
| **4** | **100%** | **100%** | **100%** | **0%** | **0%** | **0%** | **0%** |
| **5** | **100%** | **100%** | **100%** | **0%** | **0%** | **0%** | **100%** |
| **6** | **100%** | **50%** | **100%** | **0%** | **0%** | **0%** | **50%** |
| **7** | **100%** | **50%** | **100%** | **0%** | **33.3%** | **0%** | **50%** |
| **8** | **100%** | **0%** | **0%** | **0%** | **33.3%** | **0%** | **0%** |
| **9** | **100%** | **0%** | **0%** | **0%** | **33.3%** | **0%** | **0%** |
| **10** | **100%** | **50%** | **0%** | **0%** | **0%** | **0%** | **50%** |

*4.2.2.3. Trong môi trường sáng*

Bảng 3: Kết quả phân loại màu sắc trong môi trường sáng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **K** | **Black** | **Blue** | **Red** | **Orange** | **Yellow** | **White** | **Green** |
| **1** | **100%** | **0%** | **0%** | **25%** | **0%** | **0%** | **66.67%** |
| **2** | **100%** | **0%** | **0%** | **75%** | **0%** | **0%** | **66.67%** |
| **3** | **100%** | **0%** | **100%** | **25%** | **0%** | **0%** | **66.67%** |
| **4** | **100%** | **0%** | **100%** | **50%** | **0%** | **0%** | **66.67%** |
| **5** | **100%** | **0%** | **100%** | **25%** | **0%** | **0%** | **66.67%** |
| **6** | **100%** | **0%** | **0%** | **75%** | **0%** | **0%** | **66.67%** |
| **7** | **100%** | **0%** | **100%** | **75%** | **0%** | **0%** | **66.67%** |
| **8** | **100%** | **0%** | **100%** | **75%** | **0%** | **0%** | **66.67%** |
| **9** | **100%** | **0%** | **100%** | **75%** | **0%** | **0%** | **66.67%** |
| **10** | **100%** | **0%** | **100%** | **75%** | **0%** | **0%** | **66.67%** |

# KẾT LUẬN

Quá trình làm đồ án 1 đã giúp em hiểu rõ hơn về thuật toán KNN trong Machine Learning cũng như hiểu rõ hơn về lý thuyết màu sắc trong ảnh. Mặc dù đã cố gắng hết sức để tạo ra kết quả nhưng không thể tránh khỏi những sai sót. Đó là những kinh nghiệm vô cùng quý giá khiến cho em tích lũy thêm kinh nghiệm khi làm thêm những project mới.

Để hoàn thiện được đồ án 1, em xin gửi l ời cảm ơn tớ i thầ y Nguyễn Huy Hoàng đã giúp đ ỡ em trong suốt quá trình thực hiện đồ án 1.

# TÀI LIỆU THAM KHẢO

**[1]** [**https://machinelearningcoban.com/2017/01/08/knn/**](https://machinelearningcoban.com/2017/01/08/knn/)

**[2]** [**https://en.wikipedia.org/wiki/K-nearest\_neighbors\_algorithm**](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)

**[3]** [**https://build-system.fman.io/pyqt5-tutorial**](https://build-system.fman.io/pyqt5-tutorial)

**[4]** [**https://www.youtube.com/watch?v=Vde5SH8e1OQ&list=PLzMcBGfZo4-lB8MZfHPLTEHO9zJDDLpYj**](https://www.youtube.com/watch?v=Vde5SH8e1OQ&list=PLzMcBGfZo4-lB8MZfHPLTEHO9zJDDLpYj)