**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN HỌC**



**BÁO CÁO MÔN HỌC KỸ THUẬT XỬ LÝ ẢNH**

**ĐỀ TÀI: ỨNG DỤNG CÔNG NGHỆ XỬ LÝ ẢNH ĐỂ PHÂN LOẠI**

**SẢN PHẦM THEO HÌNH DẠNG**

**Sinh viên thực hiện**: Phùng Gia Thành 181603367

Nguyễn Trường Sinh 181603672

Lại Minh Tú 181603378

Nguyễn Xuân Trường 181601694

Nguyễn Văn Quang 181602385

**Lớp**: Tự động hóa 1-K59

**Giảng viên hướng dẫn**: T.S Phí Văn Lâm

Hà Nội 15/5/2022

**MỤC LỤC**

[**LỜI MỞ ĐẦU 2**](#_Toc103505190)

[**1. Giới thiệu về đề tài 3**](#_Toc103505191)

[**1.1. Mục đích nghiên cứu của đề tài 3**](#_Toc103505192)

[**1.2. Phạm vi nghiên cứu: 3**](#_Toc103505193)

[**2. Một số khái niệm và các vấn đề cơ bản trong xử lý ảnh 3**](#_Toc103505194)

[**2.1. Phần tử ảnh (Pixel – Picture Element): 3**](#_Toc103505195)

[**2.2. Ảnh màu (Color Image): 4**](#_Toc103505196)

[**2.3. Mức xám của ảnh: 5**](#_Toc103505197)

[**2.4. Ảnh nhị phân: 6**](#_Toc103505198)

[**3. Các bước thực hiện đề tài 7**](#_Toc103505199)

[**3.1. Xây dựng lưu đồ thuật toán cho đề tài 7**](#_Toc103505200)

[**3.1.1. Chọn ảnh 8**](#_Toc103505201)

[**3.1.2 Chuyển ảnh sang ảnh xám 8**](#_Toc103505202)

[**3.2. Tách biến ảnh 8**](#_Toc103505203)

[**B1. Loại bớt nhiễu trong ảnh 8**](#_Toc103505204)

[**B2. Tính toán giá trị Gradien trong ảnh 8**](#_Toc103505205)

[**B3. Loại bỏ các giá trị không phải cực đại 9**](#_Toc103505206)

[**B4. Chọn ra biên của đối tượng trong ảnh 10**](#_Toc103505207)

[**3.3. Hiển thị kết quả nhận dạng thu được 10**](#_Toc103505208)

[**4. Kết quả đạt được và phương hướng phát triển. 14**](#_Toc103505209)

[**4.1. Một vài ví dụ 14**](#_Toc103505210)

[**4.2. Hướng phát triển 14**](#_Toc103505211)

LỜI MỞ ĐẦU

Xử lý và nhận dạng ảnh là một lĩnh vực từ lâu được nhiều người quan tâm. Nó đã được ứng dụng vào nhiều lĩnh vực như:

* Trong y học, cải thiện ảnh X- quang và nhận dạng đường biên mạch máu từ ảnh chụp bằng tia X , ứng dụng vào các xét nghiệm lâm sang như phát hiện và nhận dạng u não, nội soi cắt lớp…
* Trong thiên văn học, hệ thống chụp hình gắn trên tàu vũ trụ hạn chế về kích thước và trọng lượng, do đó chất lượng hình ảnh nhận được bị giảm chất lượng như bị mờ, méo hình học và nhiễu nền. Các hình ảnh đó được xử lý bằng máy tính.
* Trong các lĩnh vực công nghiệp, người máy ngày càng đóng vai trò quan trọng. Chúng thực hiện các công việc nguy hiểm, đòi hỏi có tốc độ và độ chính xác cao vượt quá khả năng con người. Người máy sẽ trở nên tinh vi hơn và thị giác máy tính đóng vai trò quan trọng hơn. Người ta sẽ không chỉ đòi hỏi người máy phát hiện và nhận dạng các bộ phận công nghiệp mà còn phải “hiểu ” được những gì chúng “ thấy ” và đưa ra hành động phù hợp. Xử lý ảnh sẽ tác động đến thị giác của máy tính.
* Ngoài ra, xử lý và nhận dạng còn được ứng dụng trong lĩnh vực khác ít được nói đến hơn. Công an giao thông thường hay chụp ảnh trong môi trường không thuận lợi, ảnh thường bị nhòe nên cần được xử lý và nhận dạng để có thể nhìn thấy biển số xe.

Trong bài tập lớn của môn học này em xin phép được trình bày về đề tài là: **Ứng dụng công nghệ xử lý ảnh để phân loại sản phẩm**

**theo hình dạng**

Em xin chân thành cảm ơn thầy Phí Văn Lâm đã tận tình hướng dẫn em trong suốt thời gian học tập để có thể giúp em hoàn thành bài tập lớn của môn học này.

1. Giới thiệu về đề tài
   1. Mục đích nghiên cứu của đề tài

Hệ thống nhận dạng vật thể theo hình dạng là một ứng dụng dựa trên kỹ thuật xử lí ảnh số. Mục đích của nhận dạng vật thể theo hình dạng là thực hiện các bước xử lí để từ một ảnh đầu vào, máy tính có thể nhận ra chính xác các loại hình dạng thu được. Nhận dạng vật thể theo hình dạng trở thành một ứng dụng hữu ích, được đưa vào trong ứng dụng trong thực hiện đồ án sau này.

* 1. Phạm vi nghiên cứu:

Chủ yếu dựa vào các tài liệu và chạy mô phỏng nhận dạng hình ảnh trên máy bằng phần mềm Visual studio, chưa cần dùng đến mô hình thực tế.

Việc lấy ảnh từ camera hoặc từ ảnh trong máy tính vào còn gặp nhiều khó khăn vì chất lượng ảnh đầu vào chưa tốt, cùng với đó có rất nhiều yếu tố nhiễu tác động nên sẽ gây rất nhiều khó khăn cho việc nhận dạng. Do quá trình nhận dạng dựa vào phương pháp xử lý ảnh và trích xuất hình dạng từ ảnh chụp nên độ sáng khác nhau làm tăng độ phức tạp trong quá trình nhận dạng.

1. Một số khái niệm và các vấn đề cơ bản trong xử lý ảnh
   1. Phần tử ảnh (Pixel – Picture Element):

Ảnh trong thực tế là một ảnh liên tục về không gian và giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính cần thiết phải số hóa ảnh. Trong quá trình số hóa, người ta biến đổi tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng hóa thành phần giá trị. Trong quá trình này người ta sử dụng khái niệm Pixel để biểu diễn các phần tử của bức ảnh. Ở đây, cũng cần phân biệt khái niệm pixel hay đề cập đến trong các hệ thống đồ họa máy tính. Để tránh nhầm lẫn ta tạm thời gọi khái niệm pixel này là pixel thiết bị.

Khái niệm pixel thiết bị có thể xem xét như sau: khi ta quan sát màn hình (trong chế độ đồ họa), màn hình không liên tục mà gồm nhiều điểm nhỏ, gọi là pixel. Mỗi pixel bao gồm một cặp tọa độ x, y và màu. Cặp tọa độ x, y tạo nên độ phân giải (resolution). Như màn hình máy tính có nhiều độ phân giải khác nhau, hiện tại phổ biến là màn hình VGA có độ phân giải 640x480 hay XSGA độ phân giải 1024x768.

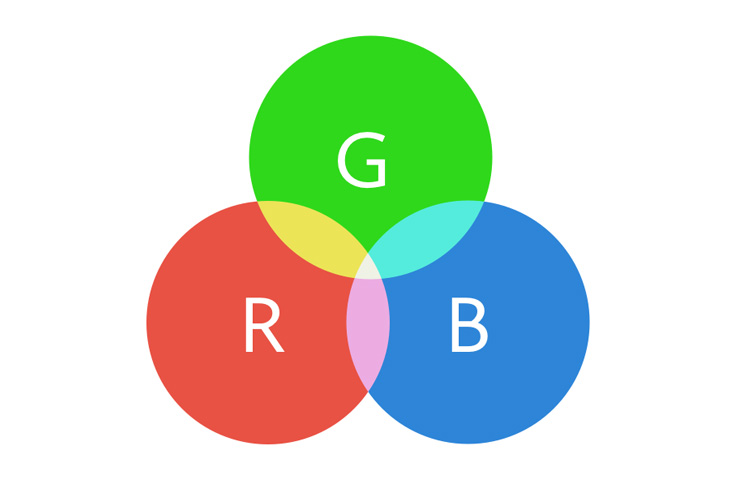
A picture containing colorful, yellow, green, different

Description automatically generated

Hình 2.1: Hình ảnh thể hiện một điểm ảnh

* 1. Ảnh màu (Color Image):

Ảnh màu chứa thông tin màu cho mỗi phần tử ảnh. Thông thường giá trị màu này dựa trên các không gian màu (color space) trong đó không gian màu thườngđược dùng là RGB tương ứng với ba kênh màu đỏ (Red) – xanh lá cây (Green) – xanh da trời (Blue). Tùy thuộc vào số bit, được sử dụng để lưư trữ màu ta có số lượng màu khác nhau, ví dụ 8 bit, 16 bit, 24 bit (True Color). Nếu ta sử dụng nhỏ hơn 24-bit để lưu trữ màu thì ta phải có 1 bảng Palette màu, nó tương tự như một bảng Lookup Table cho phép ánh xạ giữa một vị trí trong bảng với một tổ hợp của không gian màu RGB. Ví dụ như sử dụng 8 bit tương ứng với 256 màu thì ta phải có bảng ánh xạ 256 màu đó tương ứng với 256 tổ hợp Red – Green – Blue.



Hình 2.2: Ảnh màu RGB

* 1. Mức xám của ảnh:

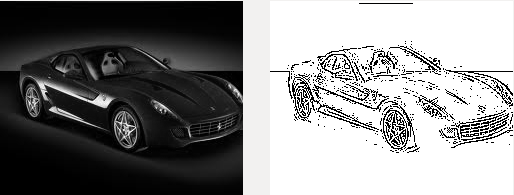
Mức xám (Gray level) là kết quả sự mã hóa tương ứng một cường độ sáng của mỗi điểm ảnh với một giá trị số - kết quả của quá trình lượng tử hóa. Ảnh đa mức xám là ảnh có sự chuyển dần mức xám từ trắng sang đen. Thực tế, một giá trị mức xám chính là sự tổ hợp của ba giá trị RGB (Red- Green – Blue). Thông thường mỗi điểm ảnh trong bức ảnh đa xám thường được mã hóa 8 bit, tương ứng với 256 mức xám.



Hình 2.3: Hình ảnh được chuyển sang mức xám

* 1. Ảnh nhị phân:

Ảnh nhị phân chỉ có 2 mức đen trắng phân biệt tức là dùng 1 bit mô tả 21 mức khác nhau. Nói cách khác: mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1.



1. Các bước thực hiện đề tài
   1. Xây dựng lưu đồ thuật toán cho đề tài

Diagram

Description automatically generated

* + 1. ****Chọn ảnh****

cv::Mat src = cv::imread("pic3.png");

if (src.empty())

return -1;\

****3.1.2 Chuyển ảnh sang ảnh xám****

cv::Mat gray;

cv::cvtColor(src, gray, cv::COLOR\_BGR2GRAY);

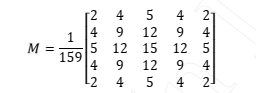
* 1. Tách biến ảnh

**Giới thiệu bộ lọc Canny**

Bộ lọc Canny là sự kết hợp của nhiều bước khác nhau để tìm và tối ưu đường biên, kết quả là cho ra một đường biên khá mảnh và chính xác. Quá trình tìm biên sử dụng phương pháp Canny có thể được thực hiện qua 4 bước sau:

### ****B1. Loại bớt nhiễu trong ảnh****

Người ta loại nhiễu trong ảnh, làm mờ ảnh đi bằng cách nhân chập ảnh với một bộ lọc Gauss, chẳng hạn bộ lọc Gauss 5x5 với hệ số δ = 1.4:

**[](https://codelearn.io/sharing/ky-thuat-loc-anh-voi-opencv-trong-cpp)**

### ****B2. Tính toán giá trị Gradien trong ảnh****

Vì đường biên trong ảnh là nơi phân cách giữa các đối tượng khác nhau, nên tại đó gradien của nó sẽ có biến đổi mạnh mẽ nhất. Để tính toán gradien trong ảnh, ta có thể sử dụng bộ lọc Sobel, hoặc trực tiếp nhân chập ma trận ảnh với các mặt nạ theo hướng x và y, chẳng hạn:

https://codelearn.io/Media/Default/Users/NTN/IMG3/02.png

Sau đó tính độ lớn gradien trong ảnh:

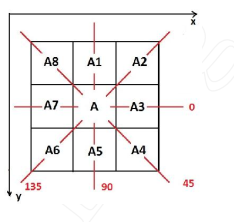
https://codelearn.io/Media/Default/Users/NTN/IMG3/03.png

Trong đó, Gx và Gy chính là đạo hàm theo hướng x, y của ảnh ta đang xét. Góc θ sẽ được làm tròn theo các hướng thẳng đứng, nằm ngang và theo hướng chéo. nghĩa là nó sẽ được làm tròn để nhận các giá trị trong 0, 45, 90 và 135 độ.

### ****B3. Loại bỏ các giá trị không phải cực đại****

Bước này sẽ tìm ra những điểm ảnh có khả năng là biên ảnh nhất bằng cách loại đi những giá trị không phải là cực đại trong bước tìm gradien ảnh ở trên. Ta thấy rằng, với giá trị của góc θ  ở trên thì biên của đối tượng có thể tuân theo bốn hướng, và ta có bốn khả năng sau:

* Nếu θ = 0 , khi đó điểm A sẽ được xem xét là một điêm trên biên độ nếu độ lớn gradien tại A lớn hơn gradien của các điểm tại A3, A7.
* Nếu θ = 45, khi đó điểm A được xem là một điểm trên biên độ nếu độ lớn gradien tại A lớn hơn độ lớn gradien của các điểm tại A4, A8
* Nếu θ = 90, khi đó điểm A sẽ được coi là một điểm nằm trên biên độ nếu độ lớn gradien tại A lớn hơn độ lớn gradien của các điểm tại A1, A5.
* Nếu θ = 135, khi đó điểm A được xem là một điểm nằm trên biên độ nếu độ lớn gradien tại A lớn hơn độ lớn gradien của các điểm tại A2, A6



### ****B4. Chọn ra biên của đối tượng trong ảnh****

 Sau bước trên, ta thu được tập hợp các điểm tương ứng trên đường biên khá mỏng. Vì những điểm có giá trị gradien lớn bao giờ cũng có xác suất biên thật sự hơn những điểm có gradien bé, do đó để xác định chính xác hơn nữa biên của các đối tượng, ta sử dụng các ngưỡng. Theo đó, bộ lọc Canny sẽ sử dụng một ngưỡng trên (upper threshold) và một ngưỡng dưới (lower threshold), nếu gradien tại một điểm trong ảnh có giá trị lớn hơn ngưỡng trên thì ta xác nhận điểm đó là một điểm biên trong ảnh, nếu giá trị này bé hơn ngưỡng dưới thì ta xác nhận điểm đó không phải là điểm biên. Trong trường hợp giá trị gradien nằm giữa ngưỡng trên và ngưỡng dưới thì nó chỉ được tính là điểm trên biên khi các điểm liên kế bên cạnh của nó có giá trị gradien lớn hơn ngưỡng trên

* 1. Hiển thị kết quả nhận dạng thu được

Shape, polygon

Description automatically generated

Code chương trình:

/\*\*

\* Simple shape detector program.

\* It loads an image and tries to find simple shapes (rectangle, triangle, circle, etc) in it.

\* This program is a modified version of `squares.cpp` found in the OpenCV sample dir.

\*/

#include <opencv2/highgui/highgui.hpp>

#include <opencv2/imgproc/imgproc.hpp>

#include <cmath>

#include <iostream>

/\*\*

\* Chức năng trợ giúp để tìm cosin của góc giữa các vectơ

\* from pt0->pt1 and pt0->pt2

\*/

static double angle(cv::Point pt1, cv::Point pt2, cv::Point pt0)

{

double dx1 = pt1.x - pt0.x;

double dy1 = pt1.y - pt0.y;

double dx2 = pt2.x - pt0.x;

double dy2 = pt2.y - pt0.y;

return (dx1\*dx2 + dy1 \* dy2) / sqrt((dx1\*dx1 + dy1 \* dy1)\*(dx2\*dx2 + dy2 \* dy2) + 1e-10);

}

/\*\*

\* Chức năng trợ giúp để hiển thị văn bản ở giữa đường viền

\*/

void setLabel(cv::Mat& im, const std::string label, std::vector<cv::Point>& contour)

{

int fontface = cv::FONT\_HERSHEY\_SIMPLEX;

double scale = 0.4;

int thickness = 1;

int baseline = 0;

cv::Size text = cv::getTextSize(label, fontface, scale, thickness, &baseline);

cv::Rect r = cv::boundingRect(contour);

cv::Point pt(r.x + ((r.width - text.width) / 2), r.y + ((r.height + text.height) / 2));

cv::rectangle(im, pt + cv::Point(0, baseline), pt + cv::Point(text.width, -text.height), CV\_RGB(255, 255, 255), cv::FILLED);

cv::putText(im, label, pt, fontface, scale, CV\_RGB(0, 0, 0), thickness, 8);

}

int main()

{

cv::Mat src = cv::imread("pic3.png");

if (src.empty())

return -1;

// Chuyển đổi sang thang độ xám

cv::Mat gray;

cv::cvtColor(src, gray, cv::COLOR\_BGR2GRAY);

// Sử dụng Canny thay vì ngưỡng để bắt các hình vuông với đổ bóng gradient

cv::Mat bw;

cv::Canny(gray, bw, 0, 50, 5);

// Tìm đường viền

std::vector<std::vector<cv::Point> > contours;

cv::findContours(bw.clone(), contours, cv::RETR\_EXTERNAL, cv::CHAIN\_APPROX\_SIMPLE);

std::vector<cv::Point> approx;

cv::Mat dst = src.clone();

for (int i = 0; i < contours.size(); i++)

{

// Đường bao gần đúng với độ chính xác tỷ lệ thuận

// đến chu vi đường bao

cv::approxPolyDP(cv::Mat(contours[i]), approx, cv::arcLength(cv::Mat(contours[i]), true)\*0.02, true);

// Bỏ qua các đối tượng nhỏ hoặc không lồi

if (std::fabs(cv::contourArea(contours[i])) < 100 || !cv::isContourConvex(approx))

continue;

if (approx.size() == 3)

{

setLabel(dst, "TRI", contours[i]); //Hình tam giác

}

else if (approx.size() >= 4 && approx.size() <= 6)

{

// Số đỉnh của đường cong đa giác

int vtc = approx.size();

// Nhận cosin của tất cả các góc

std::vector<double> cos;

for (int j = 2; j < vtc + 1; j++)

cos.push\_back(angle(approx[j%vtc], approx[j - 2], approx[j - 1]));

// Sắp xếp tăng dần các giá trị cosine

std::sort(cos.begin(), cos.end());

// Nhận cosin thấp nhất và cao nhất

double mincos = cos.front();

double maxcos = cos.back();

// Sử dụng các độ thu được ở trên và số lượng đỉnh

// để xác định hình dạng của đường viền

if (vtc == 4 && mincos >= -0.1 && maxcos <= 0.3)

setLabel(dst, "RECT", contours[i]);

else if (vtc == 5 && mincos >= -0.34 && maxcos <= -0.27)

setLabel(dst, "PENTA", contours[i]);

else if (vtc == 6 && mincos >= -0.55 && maxcos <= -0.45)

setLabel(dst, "HEXA", contours[i]);

}

else

{

//Phát hiện và gắn nhãn các vòng kết nối

double area = cv::contourArea(contours[i]);

cv::Rect r = cv::boundingRect(contours[i]);

int radius = r.width / 2;

if (std::abs(1 - ((double)r.width / r.height)) <= 0.2 &&

std::abs(1 - (area / (CV\_PI \* std::pow(radius, 2)))) <= 0.2)

setLabel(dst, "CIR", contours[i]);

}

}

cv::imshow("anh goc", src);

cv::imshow("nhan dang", dst);

cv::waitKey(0);

return 0;

1. Kết quả đạt được và phương hướng phát triển.
   1. Một vài ví dụ

Shape, square

Description automatically generated

Kết luận: Bài tập đã nhận dạng được một só hình ảnh cơ bản, đã đáp ứng đủ nhu cầu cần thiết tuy nhiên vẫn còn một số lỗi nhỏ khi nhận dạng.

* 1. Hướng phát triển

+ Có thể phát triển thêm bằng cách sử dụng camera hoặc wedcam để nhận dạng nhanh hơn so vói cách lấy ảnh từ trong máy.

+ Có thể ứng dụng vào thực tế để làm đồ án.

+ Phát triển thêm bằng cách nhận dạng nhiều hình hơn.