TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

Khoa Công Nghệ Thông Tin

-----🙞🙜🕮🙞🙜-----



**ĐỒ ÁN MÔN HỌC**

XỬ LÝ ẢNH

Đề tài: Xây dựng công cụ biến đổi ảnh bằng ngưỡng tự động

Giảng viên hướng dẫn : Nguyễn Hữu Quỳnh

Sinh viên thực hiện : Đinh Văn Đông

Trần Thị Diệu Ninh

Lớp : D10CNPM

*Hà Nội, tháng 11 năm 2018*

**LỜI MỞ ĐẦU**

Qua thời gian học tập ở lớp, chúng em đã đạt được những kiến thức cơ bản về bộ môn xử lý ảnh và hoàn thành tiến độ dự kiến xây dựng công cụ biến đổi ảnh bằng ngưỡng tự động. Để đạt được kết quả này, chúng em đã nỗ lực thực hiện và đồng thời cũng nhận được rất nhiều sự giúp đỡ, quan tâm, ủng hộ của các thầy cô, bạn bè và gia đình.

Chúng em xin chân thành cảm ơn thầy giáo Nguyễn Hữu Quỳnh đã tận tình hướng dẫn, giúp đỡ chúng em trong quá trình học tập.

Vì thời gian có hạn nên không thể tránh khỏi những thiếu sót, chúng em rất mong được sự đóng góp ý kiến từ thầy cô và các bạn.

Chúng em xin chân thành cảm ơn!

Sinh viên thực hiện

Đinh Văn Đông

Trần Thị Diệu Ninh

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 1](#_Toc531335457)

[**CHƯƠNG I: GIỚI THIỆU CHUNG** 3](#_Toc531335458)

[**1.1: Giới thiệu môn học** 3](#_Toc531335459)

[**1.2: Mục đích của xử lý ảnh** 4](#_Toc531335460)

[**1.3: Các phương pháp xử lý ảnh** 5](#_Toc531335461)

[**CHƯƠNG II : TỔNG QUAN VỀ NGƯỠNG VÀ PHƯƠNG PHÁP PHÁT HIỆN NGƯỠNG** 8](#_Toc531335462)

[**2.1: Khái niệm về ngưỡng** 8](#_Toc531335463)

[**2.2: Kĩ thuật phát hiện ngưỡng** 8](#_Toc531335464)

[**2.3: Quy trình phát hiện ngưỡng tự động** 8](#_Toc531335465)

[2.3.1 : Biến đổi ảnh đầu vào thành ảnh Bit map 8](#_Toc531335466)

[2.3.2 : Biến đổi ảnh đầu vào thành ảnh xám 8](#_Toc531335467)

[2.3.3: Áp dụng thuật toán Otsu để tìm ngưỡng tối ưu cho ảnh 8](#_Toc531335468)

[**CHƯƠNG III: CHƯƠNG TRÌNH ỨNG DỤNG** 12](#_Toc531335469)

[**3.1: Giới thiệu** 12](#_Toc531335470)

[**3.2: Giao diện chương trình** 12](#_Toc531335471)

**CHƯƠNG I: GIỚI THIỆU CHUNG**

**1.1: Giới thiệu môn học**

**Xử lý ảnh** là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lựong ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người.

Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống như: photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết, xử lý ảnh thiên văn, ảnh y tế,....

Qúa tình xử lý ảnh được xem như là quá trình thao tác đầu vào nhằm cho ra kết quả mong muốn . Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh tốt hơn hoặc một kết luận .

Ảnh tốt hơn

Ảnh

Xử lý ảnh

Kết luận

Hình 1. Quy trình xử lý ảnh

Ảnh có thể xem là hợp các điểm ảnh mỗi điểm ảnh xem như là đặc trưng cường độ ánh sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian .

Tiền xử lý

Lưu trữ

Hệ quyết định

Hậu xử lý

Trích chọn đặc điểm

Đối sánh rút ra kết luận

Thu nhận ảnh (camera...)

Hình 2: Các bước cơ bản trong một hệ thống xử lý ảnh.

Khối thu nhận ảnh :có nhiệm vụ tiếp nhận ảnh

Khối tiền xử lý : có nhiệm vụ xử lý nâng cao chất lượng ,giảm nhiễu ,phân vùng ...

Khối trích chọn đặc điểm :Có nhiệm vụ trích chọn các đặc trưng quan trọng của các bức ảnh đã được tiền xử lý để sử dụng trong hệ quyết định

Khối hậu xử lý : Có nhiệm vụ xử lý các đặc điểm đã trích chọn ,có thể lư điểm bỏ hoặc biến đổi các đặc điểm này để phù hợp với các kỹ thuật cụ thể sử dụng trong hệ quyết định.

Khối hệ quyết định và lưu trữ : Có nhiệm vụ đưa ra ra quyết định dựa trên dữ liệu đã học lưu trong khối lưu trữ

Khối kết luận :Đưa ra kết luận dựa vào quyết định của khối quyết định

**1.2: Mục đích của xử lý ảnh**

- Ứng dụng: nhận dạng đối tượng, mặt, vân tay, văn bản

- Nhận dạng chữin hoặc đánh máy phục vụ cho việc tự động hóa việc đọc tài liệu,tăng nhanh tốc độvà chất lượng thu nhận thông tin từmáy tính.

- Nhận dạng chữ viết tay (với một số ràng buộc)

- Mạng nơron là một kỹthuật mới đang được áp dụng vào nhận dạng và cho kết quả khả quan

Xử lý ảnh rất quan trọng trong cuộc sống của chúng ta nó giúp rất nhiều như nhận dạng dược khuôn mặt ,dấu vân tay .... các phần mềm này rất hữa ích .

**1.3: Các phương pháp xử lý ảnh**

**a) Phân ngưỡng**

Kỹ thuật này đặt ngưỡng để hiển thị các tông màu liên tục. Các điểm trong ảnh được so sánh với ngưỡng định trước. Giá trị của ngưỡng sẽ quyết định điểm có được hiển thị hay không.

**b) Kỹ thuật chọn theo mẫu (Patterning)**

Kỹ thuật này sử dụng một nhóm các phần tử trên thiết bị ra (máy in chẳng hạn) để

biểu diễn một pixel trên ảnh nguồn. Các phần tử của nhóm quyết định độ sáng tối của cả nhóm. Các phần tử này mô phỏng các chấm đen trong kỹ thuật nửa cường độ. Nhóm thường được chọn có dạng ma trận vuông. Nhóm nn phần tử sẽ tạo nên 2+1 mức sáng.

Ma trận mẫu thường được chọn là ma trận Rylander. Ma trận Rylander cấp 4 có dạng như

Bảng 2.2.

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 8 | 2 | 10 |
| 4 | 12 | 6 | 14 |
| 3 | 11 | 1 | 9 |
| 7 | 15 | 5 | 13 |

Bảng 2.2. Ma trận Rylander cấp 4

Việc chọn kích thước của nhóm như vậy sẽ làm giảm độ mịn của ảnh. Vì vậy kỹ thuật này chỉ áp dụng trong trường hợp mà độ phân giải của thiết bị ra lớn hơn độ phân giải của ảnh nguồn.

Ví dụ: thiết bị ra có độ phân giải 640480 khi sử dụng nhóm có kích thước 44 sẽ chỉ còn 160×120.

**c) Kỹ thuật Dithering**

Dithering là việc biến đổi một ảnh đa cấp xám (nhiều mức sáng tối) sang ảnh nhị

phân (hai mức sáng tối). Kỹ thuật Dithering đựợc áp dụng để tạo ra ảnh đa cấp sáng khi độ phân giải nguồn và đích là như nhau. Kỹ thuật này sử dụng một ma trận mẫu gọi là ma trận Dither. Ma trận này gần giống như ma trận Rylander.

Để tạo ảnh, mỗi phần tử của ảnh gốc sẽ được so sánh với phần tử tương ứng của ma trận Dither. Nếu lớn hơn, phần tử ở đầu ra sẽ sáng và ngược lại.

**d) Kỹ thuật khuếch tán lỗi (Error diffusion**)

Khuếch tán lỗi cho phép giảm thiểu mức độ mất chi tiết của ảnh khi tách ngưỡng bằng cách phân tán lỗi do lượng tử hóa ra các điểm ảnh xung quanh. Bằng cách này tổng giá trị điểm ảnh của một vùng nhỏ được giữ tương đối gần với giá trị trên ảnh gốc.Khuếch tán lỗi có hai dạng, khuếch tán lỗi một chiều và khuếch tán lỗi hai chiều.

Khuếch tán lỗi một chiều chỉ phân tán lỗi sang điểm ảnh lân cận trên cùng một dòng trong khi đó khuếch tán lỗi hai chiều phân tán lỗi sang các điểm ảnh lân cận theo một tỷ lệ xác định bởi một ma trận khuếch tán lỗi.

 Kỹ thuật khuếch tán lỗi một chiều:

* Ảnh được duyệt từ trái qua phải, từ trên xuống dưới.
* Tại mỗi điểm ảnh, giá trị điểm ảnh được tách theo ngưỡng có sẵn.
* Phần dư do lượng tử hóa được được chuyến sang điểm ảnh tiếp theo trên cùng dòng.
* Các bước được lặp lại cho đến hết dòng, phần dư của điểm ảnh cuối cùng sẽ được loại bỏ.

 Kỹ thuật khuếch tán lỗi hai chiều:

* Các bước được thực hiện như khuếch tán lỗi một chiều, tuy nhiên lỗi do lượng tử hóa sẽ đươc phân tán ra các điểm xung quanh theo tỷ lệ xác định bởi ma trận khuếch tán

Trong xử lý ảnh con rất nhiều kĩ thuật được sử dụng như :phân ngưỡng, lọc trùng bình ,lọc trung vị ,phân vùng ,sử dụng biểu đồ màu ,sử dụng phương pháp moment màu ......

**CHƯƠNG II : TỔNG QUAN VỀ NGƯỠNG VÀ PHƯƠNG PHÁP PHÁT HIỆN NGƯỠNG**

**2.1: Khái niệm về ngưỡng**

* Ngưỡng là phương pháp phân đoạn ảnh đơn giản nhất, từ một ảnh đa cấp xám ta có thể tạo ảnh nhị phân.
* Phương pháp biến đổi ảnh sử dụng ngưỡng đơn giản là thay thế từng giá trị pixel một trong ảnh bằng một giá trị khác nếu cường độ Iij của nó nhỏ hơn hoặc lớn hơn một ngưỡng T cố định.

**2.2: Kĩ thuật phát hiện ngưỡng**

* Thuật toán phân ngưỡng tối ưu từ ảnh xám .
* Thuật toán phân ngưỡng tối ưu từ ảnh màu .

Ở đề tài được giao , chúng em thực hiện đề tài theo thuật toán phân ngưỡng tối ưu từ ảnh xám sử dụng thuật toán Otsu.

Thay vì chọn một ngưỡng cố định thỳ thuật toán này dựa vào sự phân bố mức xám của các pixel trong ảnh sẽ tính toán một ngưỡng tối ưu cho ảnh.

**2.3: Quy trình phát hiện ngưỡng tự động**

2.3.1 : Biến đổi ảnh đầu vào thành ảnh Bit map

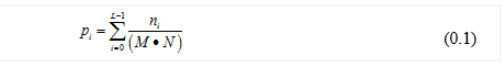
2.3.2 : Biến đổi ảnh đầu vào thành ảnh xám

Ảnh cần biến đổi sẽ được chuyển thành ảnh đa cấp xám ( grayscale) bằng công thức :

int gray = (int)((color.R \* **0.3**) + (color.G \* **0.59**) + (color.B \* **0.11**));

2.3.3: Áp dụng thuật toán Otsu để tìm ngưỡng tối ưu cho ảnh

* Đầu tiên sử dụng lược đồ Histogram để biểu diễn tần suất các mức xám trong ảnh :



+ Trong đó : ni là số lượng điểm ảnh tại giá trị i.

M\*N : tổng số lượng điểm ảnh ban đầu.

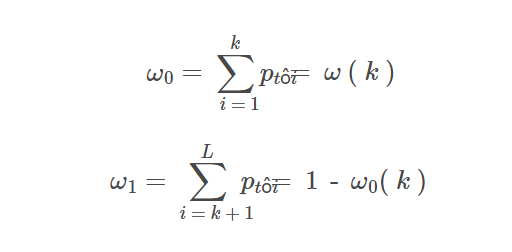
L : 0,1,…,256.

* Chia điểm ảnh thành 2 lớp :

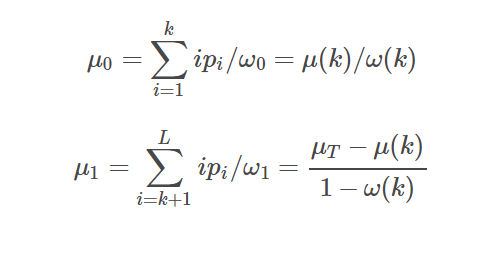
+*C0* là tập hợp các pixel có giá trị từ [1,…,k].

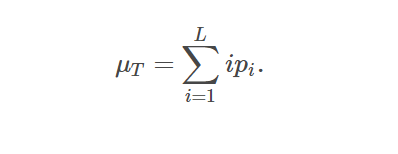
+*C1* là tập hợp các pixel có giá trị từ [k+1,…,L].

Xác suất lớp tổng thể , *w0* và *w1 ,*là:

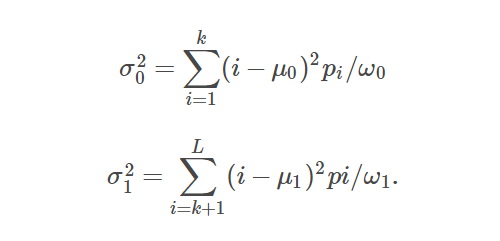


* *µ0* và *µ*1 là giá trị trung bình của các pixel trong *C0* và *C1:*
* *µ(T* ) là giá trị pixel trung bình của ảnh*:*

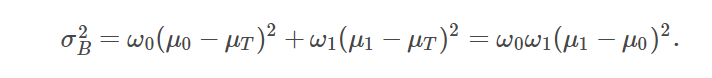




* Phương sai của 2 lớp *µ0* và *µ1* là *𝛔 20* và *𝛔12* :



* Ngưỡng k\* tốt nhất được chọn là giá trị mà tại đó nó làm cho sự chênh lệch giữa hai đoạn trên đồ thị đạt giá trị cực đại đại (tìm ra đỉnh của 𝛔*b2 ) :*



**CHƯƠNG III: CHƯƠNG TRÌNH ỨNG DỤNG**

**3.1: Giới thiệu**

* Chương trình demo được viết bằng ngôn ngữ lập trình C# trên công cụ Visual Studio 2015.
* Các chức năng chính : + Đọc ảnh vào  
   + Biến đổi ảnh vỡi ngưỡng

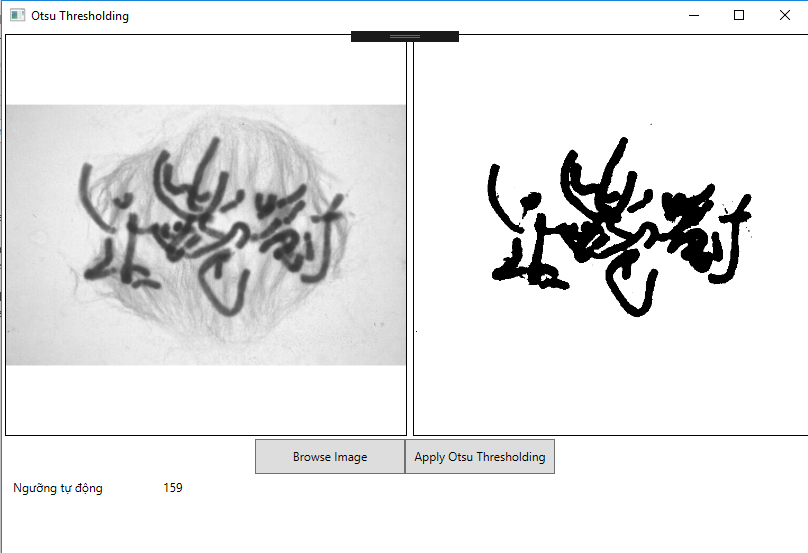
+ Hiển thị ngưỡng tối ưu

+ Hiển thị ảnh đã biến đổi

**3.2: Giao diện chương trình**



Và kết quả thu được :



**NHẬN XÉT CỦA GIÁO VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………