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**

[**CHƯƠNG I : TỔNG QUAN VỀ NGƯỠNG VÀ PHƯƠNG PHÁP PHÁT HIỆN NGƯỠNG** 3](#_Toc531300114)

[**1.1** **: Khái niệm về ngưỡng** 3](#_Toc531300115)

[**1.2** **: Kĩ thuật phát hiện ngưỡng** 3](#_Toc531300116)

[**1.3** **: Quy trình phát hiện ngưỡng tự động** 3](#_Toc531300117)

[1.3.1 : Biến đổi ảnh đầu vào thành ảnh Bit map 3](#_Toc531300118)

[1.3.2 : Biến đổi ảnh đầu vào thành ảnh xám 3](#_Toc531300119)

[1.3.3: Áp dụng thuật toán Otsu để tìm ngưỡng tối ưu cho ảnh 3](#_Toc531300120)

[**CHƯƠNG II: CHƯƠNG TRÌNH ỨNG DỤNG** 7](#_Toc531300121)

[**2.1: Giới thiệu** 7](#_Toc531300122)

[**2.2: Giao diện chương trình** 7](#_Toc531300123)

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

* 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.
  1. **: 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.

* 1. **: Quy trình phát hiện ngưỡng tự động**

1.3.1 : Biến đổi ảnh đầu vào thành ảnh Bit map

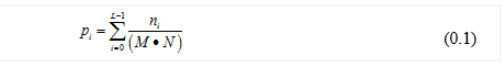
1.3.2 : Biến đổi ảnh đầu vào thành ảnh xám

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**));

1.3.3: 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.

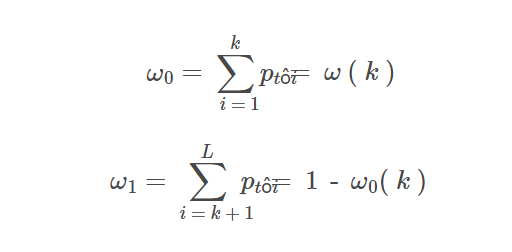
L : 0,1,…,256.

* Chia điểm ảnh thành 2 lớp :

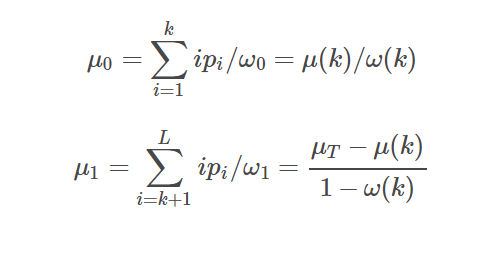
+*C0* là tập hợp các pixel có giá trị từ [1,…,k].

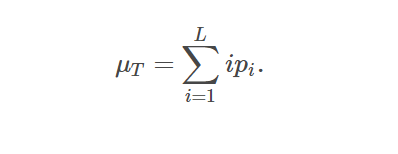
+*C1* là tập hợp các pixel có giá trị từ [k+1,…,L].

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

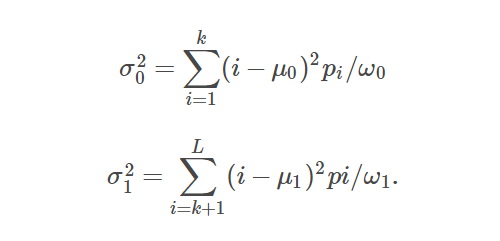


* *µ0* và *µ*1 là giá trị trung bình của các pixel trong *C0* và *C1:*
* *µ(T* ) là giá trị pixel trung bình của ảnh*:*

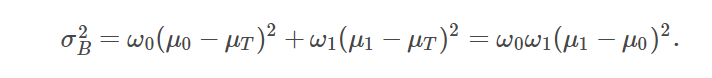




* Phương sai của 2 lớp *µ0* và *µ1* là *𝛔 20* và *𝛔12* :



* 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 II: CHƯƠNG TRÌNH ỨNG DỤNG**

**2.1: 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

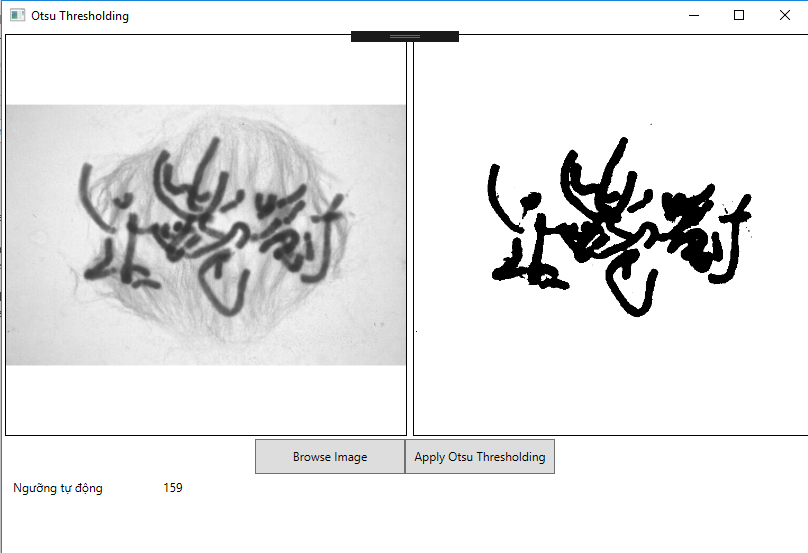
+ Hiển thị ngưỡng tối ưu

+ Hiển thị ảnh đã biến đổi

**2.2: Giao diện chương trình**



Và kết quả thu được :



**NHẬN XÉT CỦA GIÁO VIÊN**

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

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

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

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

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

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

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

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

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

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