TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TIỂU LUẬN MÔN XÁC SUẤT VÀ THÔNG KÊ ỨNG DỤNG CHO CÔNG NGHỆ THÔNG TIN**

**MIDTERM ESSAY**

*Người hướng dẫn*: **TS TRẦN LƯƠNG QUỐC ĐẠI**

*Người thực hiện*: **VÕ NHẬT HÀO – 522H0090**

Lớp **: 22H50202**

Khoá  **: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TIỂU LUẬN MÔN XÁC SUẤT VÀ THÔNG KÊ ỨNG DỤNG CHO CÔNG NGHỆ THÔNG TIN**

**MIDTERM ESSAY**

Người hướng dẫn: **TS TRẦN LƯƠNG QUỐC ĐẠI**

Người thực hiện: **VÕ NHẬT HÀO – 522H0090**

Lớp **: 22H50202**

Khoá  **: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

* Lời đầu tiên em cảm ơn thầy Trần Lương Quốc Đại đã đồng hành cùng em. Cảm ơn thầy đã dạy và hướng dẫn cho em hoàn thành bài tiểu luận giữa kì môn Thực Hành Xác Suất và Thống Kê Ứng Dụng Công Nghệ Thông Tin.
* Cảm ơn khoa Công Nghệ Thông Tin đã cho em tiếp xúc bài tiểu luận này và em tin chắc rằng bài tiểu luận này sẽ cho em tiếp cận với những kĩ năng làm việc về các bài báo cáo và kĩ năng trình bày nhằm tiếp cận cho công việc, khi đi làm có thể thích nghi với môi trường mới hơn.
* Ngoài ra, em chưa được tiếp cận đến nhiều bài tập về tiểu luận nên có thể còn nhiều lỗi và một số sai sót. Hy vọng thầy/cô chấm điểm và góp ý để em có thể càng phát triển, tốt hơn trên lĩnh vực này.
* Lời cuối cùng, em xin cảm ơn thầy/cô đã đọc và chấm điểm, đánh giá. Chúc cho thầy/cô thật nhiều sức khỏe để có thể truyền tải những kiến thức mới cho em và các bạn sinh viên

**BÀI TIỂU LUẬN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của Thầy Trần Lương Quốc Đại. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

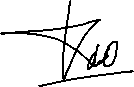
Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 8 tháng 11 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*



*Võ Nhật Hào*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

* Bài tiểu luận nghiên cứu về các chức năng của thư viện Thống kê và khảo sát cân bằng lược đồ và phù hợp lược đồ trong xử lý ảnh bằng ngôn ngữ Python.
* Được thực hiện bởi sinh viên khoa Công Nghệ Thông Tin của Đại học Tôn Đức Thắng.
* Võ Nhật Hào – MSSV: 522H0090
* Bài tiểu luận được tham khảo từ nguồn website của khoa đã đề cập và tham khảo thêm nhiều khác trên internet.
* Hiểu thêm về các hàm thống kê và cách sử dụng của các hàm và biết thêm về các khái niệm về thuật toán cân bằng lược đồ và thuật toán phù hợp lược đồ. Có thể áp dụng để xử lý ảnh cho những ảnh bị tương phản (quá tối hoặc quá sáng).

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc150445519)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc150445520)

[TÓM TẮT iv](#_Toc150445521)

[MỤC LỤC 1](#_Toc150445522)

[CHƯƠNG 1 – THƯ VIỆN THỐNG KÊ BẰNG PYTHON 2](#_Toc150445523)

[1.1 Tổng quan về thư viện thống kê bằng Python. 2](#_Toc150445524)

[1.2 Giải thích về các hàm, ví dụ code và màn hình chạy. 3](#_Toc150445525)

[1.3 NormalDist. 19](#_Toc150445527)

[CHƯƠNG 2 – THUẬT TOÁN CÂN BẰNG LƯỢC ĐỒ 24](#_Toc150445528)

[2.1 Các vấn đề, ràng buộc/điều kiện, phương pháp/thuật toán. 24](#_Toc150445529)

[2.2 Các ví dụ. 24](#_Toc150445539)

[2.3 Nhận xét, phân tích, đánh giá. 25](#_Toc150445542)

[CHƯƠNG 3 – THUẬT TOÁN PHÙ HỢP LƯỢC ĐỒ 26](#_Toc150445546)

[3.1 Các vấn đề, ràng buộc/điều kiện, phương pháp/thuật toán. 26](#_Toc150445547)

[3.2 Các ví dụ. 26](#_Toc150445559)

[3.3 Nhận xét, phân tích, đánh giá. 28](#_Toc150445564)

[CHƯƠNG 4 – THỰC HIỆN 28](#_Toc150445567)

[4.1 Giải thích các dòng code trong chương 2 và 3. 28](#_Toc150445568)

[4.2 Các bước xây dựng và chạy sourcecode. 31](#_Toc150445586)

[4.3 Màng hình chạy. 32](#_Toc150445587)

[TÀI LIỆU THAM KHẢO 33](#_Toc150445593)

CHƯƠNG 1 – THƯ VIỆN THỐNG KÊ BẰNG PYTHON

* 1. Tổng quan về thư viện thống kê bằng Python.
* **statistics.mean**(): Sử dụng để tính giá trị trung bình của tập dữ liệu.
* **statistics.fmean**(): Sử dụng để tính giá trị trung bình của tập dữ liệu chứa các số thực và tính nhanh hơn hàm **mean()**.
* **statistics.geometric\_mean**(): Sử dụng để tính giá trị trung bình hình học của tập dữ liệu.
* **statistics.harmonic\_mean**(): Sử dụng để tính giá trị trung bình điều hòa của tập dữ liệu.
* **statistics.median**(): Sử dụng để tính giá trị trung vị của tập dữ liệu.
* **statistics.median\_low**(): Sử dụng để tính giá trị trung vị thấp của tập dữ liệu.
* **statistics.median\_high**(): Sử dụng để tính giá trị trung vị cao của tập dữ liệu.
* **statistics.median\_grouped**(): Sử dụng để tính giá trị trung vị của tập dữ liệu được phân theo nhóm.
* **statistics.mode**(): Sử dụng để tìm giá trị mà xuất hiện nhiều lần nhất của tập dữ liệu và trả về một giá trị.
* **statistics.multimode**(): Sử dụng để tìm giá trị mà xuất hiện nhiều lần nhất của tập dữ liệu và trả về một danh sách.
* **statistics.quantiles**(): Sử dụng để tìm các giá trị phân vị của tập dữ liệu.
* **statistics.pstdev**(): Sử dụng để tính độ lệch chuẩn của toàn bộ dữ liệu.
* **statistics.pvariance**(): Sử dụng để tính phương sai của toàn bộ tập dữ liệu.
* **statistics.stdev**(): Sử dụng để tính độ lệch chuẩn của mẫu tập dữ liệu.
* **statistics.variance**(): Sử dụng để tính phương sai của mẫu tập dữ liệu.
* **statistics.covariance**(): Sử dụng để tính hiệp phương sai của hai tập dữ liệu.
* **statistics.correlation**(): Sử dụng để tính hệ số tương quan của hai tập dữ liệu.
* **statistics.linear\_regression**(): Sử dụng để tính hồi quy tuyến tính của hai tập dữ liệu.
  1. Giải thích về các hàm, ví dụ code và màn hình chạy.
* statistics.mean():

- Hàm sử dụng để tính giá trị trung bình của một tập dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng). Nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**mean**(data).

- Tham số truyền vào (data) của hàm **mean** là một tập dữ liệu, có thể là một danh sách hay một mảng.

- Ví dụ code và màn hình chạy:

* Tính giá trị trung bình của dữ liệu [2, 4, 6, 8, 10]
* Cho ra kết quả là 6.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.fmean():**

- Hàm sử dụng để tính giá trị trung bình của một tập dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng). Hàm này chạy nhanh hơn hàm mean() và luôn trả về một số thực. Nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**fmean**(data, weights).

- Tham số thứ 1 (data) truyền vào của hàm **fmean** là một dữ liệu có thể là một danh sách hay một mảng.

- Tham số thứ 2 (weights) truyền vào của hàm **fmean** là một tập trọng số tương ứng

với mỗi phần tử trong tham số thứ 1 (data).

- Ví dụ code và màn hình chạy:

* Tính giá trị trung bình của môn Xác suất thống kê:
* Điểm quá trình 1(10%): 9.0
* Điểm quá trình 2(10%): 8.0
* Điểm giữa kì (20%): 8.5
* Điểm cuối kì (50%): 7.0
* Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

  Mô tả được tạo tự độngKết quả là 7.7
* Màn hình chạy:
* **statistics.geometric\_mean():**

- Hàm sử dụng để tính giá trị trung bình hình học của một tập dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng). Nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError. Giá trị trung bình hình học được tính bằng cách lấy tích của các số trong tập dữ liệu, sau đó lấy căn bậc n của tích đó (n là số lượng phần tử trong tập dữ liệu).

- Cú pháp để gọi hàm như sau: statistics.**geometric\_mean**(data).

- Tham số truyền vào (data) của hàm **geometric\_mean** là một dữ liệu có thể là một

danh sách hay một mảng. Nếu dữ liệu truyền vào rỗng, chứa số 0 hoặc số âm sẽ ném ra

ngoại lệ StatisticsError.

- Ví dụ code và màn hình chạy:

* Tính giá trị trung bình hình học của các số [6, 12, 24]
* Cho ra kết quả là 12.

Ảnh có chứa văn bản, ảnh chụp màn hình, danh thiếp, Phông chữ

Mô tả được tạo tự động

*  Màn hình chạy:
* **statistics.harmonic\_mean():**

- Hàm sử dụng để tính giá trị trung bình điều hòa của một tập dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng). Nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError. Giá trị trung bình điều hòa được tính bằng cách lấy nghịch đảo của giá trị trung bình của nghịch đảo các số trong tập dữ liệu.

- Cú pháp để gọi hàm như sau: statistics.**harmonic\_mean**(data).

- Tham số truyền vào (data) của hàm **harmonic\_mean** là một dữ liệu có thể là một danh sách hay một mảng. Nếu một trong các giá trị bằng 0 thì kết quả sẽ bằng 0.

- Ví dụ code và màn hình chạy:

* Tính giá trị trung bình điều hòa của các số [20, 30]
* Cho ra kết quả là 24.0.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.median():**

- Hàm sử dụng để tính giá trị trung vị của một tập dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng). Nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError. Giá trị trung vị là giá trị ở giữa của tập dữ liệu sau khi được sắp xếp (sắp xếp theo thứ tự tăng dần).

- Cú pháp để gọi hàm như sau: statistics.**median**(data).

- Tham số truyền vào (data) của hàm **median** là một dữ liệu có thể là một danh sách hay một mảng. Khi trong tập dữ liệu là số chẵn thì giá trị trung vị được xác định bằng cách lấy trung bình cộng của hai giá trị ở giữa. Khi trong tập dữ liệu là số lẻ thì giá trị trung vị là phần tử ở giữa sau khi được sắp xếp.

- Ví dụ code và màn hình chạy:

* Tính giá trị trung bình thấp của các số [1, 3, 6, 4, 9]
* Cho ra kết quả 4

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.median\_low():**

- Hàm sử dụng để tính giá trị trung vị thấp của một tập dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng). Nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError. Giá trị trung vị thấp là giá trị ở giữa của tập dữ liệu sau khi được sắp xếp (sắp xếp theo thứ tự tăng dần).

- Cú pháp để gọi hàm như sau: statistics.**median\_low**(data).

- Tham số truyền vào (data) của hàm **median\_low** là một dữ liệu, có thể là một danh sách hay một mảng. Khi trong tập dữ liệu là số chẵn thì giá trị trung vị thấp được trả về giá trị trung vị của phần tử thấp hơn sau khi sắp xếp. Khi trong tập dữ liệu là số lẻ thì giá trị trung vị thấp là phần tử ở giữa sau khi được sắp xếp.

- Ví dụ code và màn hình chạy:

* Tính giá trị trung bình điều hòa của các số [2, 3, 5, 4, 6]
* Cho kết quả: 4

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, danh thiếp

Mô tả được tạo tự động

* Màn hình chạy:



* **statistics.median\_high():**

- Hàm sử dụng để tính giá trị trung vị cao của một tập dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng). Nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError. Giá trị trung vị cao là giá trị ở giữa của tập dữ liệu sau khi được sắp xếp (sắp xếp theo thứ tự tăng dần).

- Cú pháp để gọi hàm như sau: statistics.**median\_high**(data).

- Tham số truyền vào (data) của hàm **median\_high** là một dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng). Khi trong tập dữ liệu là số chẵn thì giá trị trung vị cao được

trả về giá trị trung vị của phần tử cao hơn sau khi sắp xếp. Khi trong tập dữ liệu là số lẻ

thì giá trị trung vị cao là phần tử ở giữa sau khi được sắp xếp.

- Ví dụ code và màn hình chạy:

* Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

  Mô tả được tạo tự độngTính giá trị trung vị cao của các số [2, 3, 5, 4, 6] và cho ra kết quả 5
* Màn hình chạy:
* **statistics.median\_grouped():**

- Hàm sử dụng để tìm giá trị trung vị của một tập dữ liệu đã được phân nhóm (tập dữ liệu có thể là danh sách hoặc một mảng). Nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**median\_grouped**(data, interval).

- Tham số thứ 1 (data) truyền vào của hàm **median\_grouped** là một dữ liệu (tập dữ liệu có thể là danh sách hoặc một mảng).

- Tham số thứ 2 (interval) truyền vào của hàm **median\_grouped** là khoảng giá trị của

nhóm, với giá trị truyền vào là số nguyên hoặc số thực dương.

- Ví dụ code và màn hình chạy:

* Tính giá trị trung vị phân nhóm của các số [1.5, 2.5, 3.5, 4.5, 5.5] đã được nhóm thành các khoảng 2
* Cho ra kết quả là 3.5

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.mode():**

- Hàm sử dụng để tìm giá xuất hiện nhiều nhất của một tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Nếu có nhiều giá trị xuất hiện nhiều lần thì sẽ trả về giá trị gặp đầu tiên trong danh sách và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**mode**(data).

- Tham số truyền vào (data) của hàm **mode** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Ví dụ code và màn hình chạy:

* Tìm giá trị xuất hiện của các số [2, 2, 4, 5, 8, 9, 4, 3]
* Cho ra kết quả là 2

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.multimode():**

- Hàm sử dụng để tìm giá xuất hiện nhiều nhất của một tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Hàm sẽ trả về một danh sách chứa tất cả các giá trị xuất hiện nhiều nhất trong tập dữ liệu. Nếu có nhiều giá trị xuất hiện nhiều lần thì sẽ trả về danh sách các giá trị xuất hiện nhiều trong danh sách và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**multimode**(data).

- Tham số truyền vào (data) của hàm **multimode** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Ví dụ code và màn hình chạy:

* Tìm giá trị xuất hiện của các số [1, 3, 2, 3, 3, 4, 5, 4, 4]
* Cho ra kết quả [3, 4]

Ảnh có chứa văn bản, ảnh chụp màn hình, danh thiếp, Phông chữ

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.quantiles():**

- Hàm sử dụng để tính phân vị của một tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Các giá trị được tính dựa trên số lượng phân vị được chỉ định bởi tham số n và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**quantiles**(data, n = 4, method = ‘exclusive’).

- Tham số thứ 1 (data) truyền vào của hàm **quantiles** là một dữ liệu có thể là danh sách

hoặc là một mảng chứa nhiều giá trị.

- Tham số thứ 2 (n) truyền vào của hàm **quantiles** là số lượng phân vị muốn tính. Mặc định n = 4, hàm sẽ trả về 3 giá trị và phân chia thành 4 phần bằng nhau lần lượt các phân vị là 25%, 50%, 75%.

- Tham số thứ 3 (method) truyền vào của hàm **quantiles** là phương pháp để chia phân vị, có 2 phương pháp là ‘inclusive’ và ‘exclusive’. Đối với phương pháp ‘inclusive’ là bao gồm cả giá trị tối thiểu và tối đa trong tập dữ liệu để tính phân vị và nó chia tập dữ liệu thành các phần bằng nhau dựa trên số lượng phân vị được yêu cầu. Đối với phương pháp ‘exclusive’ là bỏ qua giá trị tối thiểu và giá trị tối đa trong tập dữ liệu để tính phân vị và nó chia tập dữ liệu thành các phần bằng nhau dựa trên số lượng phân vị được yêu cầu.

- Ví dụ code và màn hình chạy:

* Tìm giá trị xuất hiện của các số [15, 25, 35, 45, 55, 65]
* Ảnh có chứa văn bản, ảnh chụp màn hình, danh thiếp, Phông chữ

  Mô tả được tạo tự độngCho ra kết quả là [22.5, 40.0, 57.5].
* Màn hình chạy:
* **statistics.pstdev():**

- Hàm sử dụng để tính độ lệch chuẩn của toàn bộ dân số (population standard deviation) của một tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Độ lệch

chuẩn cho biết mức độ biến đổi hoặc mức độ phân tán trong tập dữ liệu và nếu dữ liệu

truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**pstdev**(data, mu = None).

- Tham số thứ 1 (data) truyền vào của hàm **pstdev** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Tham số thứ 2 (mu) truyền vào của hàm **pstdev** là giá trị trung bình của dữ liệu. Nếu không chỉ định (mu) hàm sẽ tính giá trị trung bình của dữ liệu.

- Ví dụ code và màn hình chạy:

* Tính độ lệch chuẩn của các số [2, 4, 6, 8] và cho ra kết quả: 2.23606797749979

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Mà hình chạy:
* **statistics.pvariance():**

- Hàm sử dụng để tính phương sai (population variance) của một tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Phương sai cho biết mức độ biến đổi hoặc mức độ phân tán trong tập dữ liệu và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**pvariance**(data, mu = None).

- Tham số thứ 1 (data) truyền vào của hàm **pvariance** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Tham số thứ 2 (mu) truyền vào của hàm **pvariance** là giá trị trung bình của dữ liệu. Nếu không chỉ định (mu) hàm sẽ tính giá trị trung bình của dữ liệu.

- Ví dụ code và màn hình chạy:

* Tính phương sai của các số [2, 4, 6, 8]
* Cho ra kết quả: 5

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* Màn hình chạy:



* **statistics.stdev():**

- Hàm sử dụng để tính độ lệch chuẩn của một mẫu cụ thể (sample standard deviation) của một tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Độ lệch chuẩn cho biết mức độ biến đổi hoặc mức độ phân tán trong tập dữ liệu và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**stdev**(data, xbar = None).

- Tham số thứ 1 (data) truyền vào của hàm **stdev** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Tham số thứ 2 (xbar) truyền vào của hàm **stdev** là giá trị trung bình của dữ liệu. Nếu không chỉ định (xbar) hàm sẽ tính giá trị trung bình của dữ liệu.

- Ví dụ code và màn hình chạy:

* Tính độ lệch chuẩn của các số [2, 4, 6, 8]
* Cho kết quả: 2.581988897471611

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.variance():**

- Hàm sử dụng để tính phương sai (sample variance) của một tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Phương sai cho biết mức độ biến đổi hoặc mức độ phân tán trong tập dữ liệu và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**variance**(data, xbar= None).

- Tham số thứ 1 (data) truyền vào của hàm **variance** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Tham số thứ 2 (xbar) truyền vào của hàm **variance** là giá trị trung bình của dữ liệu. Nếu không chỉ định (xbar) hàm sẽ tính giá trị trung bình của dữ liệu.

- Ví dụ code và màn hình chạy:

* Tính phương sai của các số [2, 4, 6, 8]
* Cho ra kết quả: 6.666666666666667

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.covariance():**

- Hàm sử dụng để tính hiệp phương sai của 2 tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Hiệp phương sai cho biết mức độ thay đổi đồng nhất giữa hai

biến số. Nó phản ánh xu hướng của hai biến cùng tăng lên hoặc giảm xuống đồng thời

và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**covariance** (x, y).

- Tham số thứ 1 (x) truyền vào của hàm **covariance** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Tham số thứ 2 (y) truyền vào của hàm **covariance** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Lưu ý: Cả hai đầu vào phải có cùng độ dài, mỗi tham số đầu vào có độ dài không ít hơn 2, nếu không thì sẽ xảy ra StatisticsError.

- Ví dụ code và màn hình chạy:

* Tính hiệp phương sai của các số [2, 4, 6, 8, 10] và [1, 3, 5, 7, 9]
* Cho ra kết quả: 10.0

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.correlation():**

- Hàm sử dụng để tính hệ số tương quan của 2 tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng. Hệ số tương quan cho biết mối quan hệ tuyến tính giữa 2 biến số,

có giá trị trong khoảng [-1, 1] và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ

StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**correlation**(x, y).

- Tham số thứ 1 (x) truyền vào của hàm **correlation** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Tham số thứ 2 (y) truyền vào của hàm **correlation** là một dữ liệu có thể là danh sách hoặc là một mảng chứa nhiều giá trị.

- Lưu ý: Cả hai đầu vào phải có cùng độ dài, mỗi tham số đầu vào có độ dài không ít hơn 2 và không cần phải liên tục, nếu không thì sẽ xảy ra StatisticsError.

- Nếu hệ số tương quan được tính ra gần bằng 1: tương quan thuận hoàn toàn.

- Nếu hệ số tương quan được tính ra gần bằng -1: tương quan nghịch hoàn toàn.

- Nếu hệ số tương quan được tính ra gần bằng 0: không tương quan.

- Ví dụ code và màn hình chạy:

* Tính hệ số tương quan của 2 tập dữ liệu [2, 4, 6, 8, 10] và [1, 3, 5, 7, 9]
* Cho ra kết quả: 1.0

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* Màn hình chạy:
* **statistics.linear\_regression():**

- Hàm sử dụng để tính hồi quy tuyến tính của 2 tập dữ liệu, tập dữ liệu có thể là danh sách hoặc một mảng và nếu dữ liệu truyền vào rỗng sẽ ném ra ngoại lệ StatisticsError.

- Cú pháp để gọi hàm như sau: statistics.**linear\_regression**(x, y, proportional = False).

- Tham số thứ 1 (x) truyền vào của hàm **linear\_regression** là một dữ liệu độc lập.

- Tham số thứ 2 (y) truyền vào của hàm **linear\_regression** là một dữ liệu phụ thuộc.

- Tham số thứ 3 (proportional) truyền vào của hàm **linear\_regression** là để chỉ định kiểu hồi quy tuyến tính. Nếu proportional = False (mặc định): hàm sẽ tìm phương trình đường thẳng tốt nhất đi qua tập dữ liệu. Nếu proportional = True: hàm sẽ thực hiện hồi quy tuyến tính tỷ lệ thuận và tìm phương trình đường thẳng đi qua gốc tọa độ (0, 0).

- Lưu ý: Cả hai đầu vào phải có cùng độ dài, mỗi tham số đầu vào có độ dài không ít hơn 2 và biến x không được liên tục, nếu không thì sẽ xảy ra StatisticsError.

- Ví dụ code và màn hình chạy:

* Tính hồi quy tuyến tính của 2 tập dữ liệu [2, 4, 6, 8, 10] và [1, 3, 5, 7, 9]
* Cho ra kết quả:
* Hệ số góc (slope): 1.0
* Hệ số góc (intercept): -1.0

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Màn hình chạy:
  1. NormalDist.
* **Khái niệm**

- Phân phối chuẩn là công cụ dùng để tạo và điều chỉnh phân phối chuẩn của một biến ngẫu nhiên. Lớp này dùng để xử lý giá trị trung bình và độ lệch chuẩn của các phép đo lường dữ liệu như một thực thể duy nhất.

- Phân phối chuẩn được rút ra từ Định lý Trung tâm và có rất nhiều ứng dụng trong thống kê.

- Cú pháp để gọi hàm như sau: statistics.**NormalDist**(mu = 0.0, sigma = 1.0)

+ Trả về một đối tượng NormalDist mới với giá trị trung bình là mu, và độ lệch chuẩn là sigma. Nếu sigma là số âm thì lỗi StatisticsError sẽ xuất hiện.

* **Các thuộc tính của lớp statistics.NormalDist:**
* **mean**: Trả về giá trị trung bình của phân phối chuẩn.
* **median**: Trả về giá trị trung vị của phân phối chuẩn.
* **mode**: Trả về giá trị xuất hiện nhiều nhất của phân phối chuẩn.
* **stdev**: Trả về giá trị độ lệch chuẩn của phân phối chuẩn.
* **variance**: Trả về giá trị phương sai của phân phối chuẩn, bằng bình phương của độ lệch chuẩn.
* **from\_samples**(data): Tạo một đối tượng phân phối chuẩn với các tham số mu và sigma được ước tính từ dữ liệu sử dụng hàm fmean() và stdev().

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* Màn hình chạy:

Ảnh có chứa văn bản, Phông chữ, màu trắng, thuật in máy

Mô tả được tạo tự động

* **samples**(n, \*, seed = None): Tạo n mẫu ngẫu nhiên cho giá trị trung bình và độ lệch chuẩn đã cho. Trả về danh sách các giá trị float.

Ảnh có chứa văn bản, ảnh chụp màn hình, danh thiếp, Phông chữ

Mô tả được tạo tự động

* + Màn hình chạy:



* **pdf**(x): Sử dụng hàm mật độ xác suất (pdf), tính xác suất có liên quan đến giá trị x của biến ngẫu nhiên X. Xác suất liên quan là tỷ lệ giữa xác suất một mẫu xuất hiện trong khoảng nhỏ và chiều rộng của khoảng đó.

Ảnh có chứa văn bản, ảnh chụp màn hình, danh thiếp, Phông chữ

Mô tả được tạo tự động

* + Màn hình chạy:
* **cdf**(x): Sử dụng hàm phân phối tích lũy (cdf), tính xác suất mà một biến ngẫu nhiên X sẽ nhỏ hơn hoặc bằng giá trị x.

Ảnh có chứa văn bản, ảnh chụp màn hình, danh thiếp, Phông chữ

Mô tả được tạo tự động

* + Màn hình chạy:
* **inv\_cdf**(p): Tính hàm phân phối tích lũy nghịch đảo, còn được gọi là hàm phân vị hoặc hàm điểm phần trăm. Tìm giá trị x của biến ngẫu nhiên X sao cho xác suất của biến đó nhỏ hơn hoặc bằng giá trị p.

Ảnh có chứa văn bản, ảnh chụp màn hình, danh thiếp, Phông chữ

Mô tả được tạo tự động

* + Màn hình chạy:



* **overlap**(other): Đo lường mức độ trùng lắp giữa hai phân phối xác suất chuẩn. Trả về một giá trị trong khoảng từ 0.0 đến 1.0 thể hiện diện tích trùng lắp cho hai hàm mật độ xác suất.

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động**

* + Màn hình chạy:



* **quantiles**(n = 4): Chia phân phối chuẩn thành n khoảng liên tục với xác suất bằng nhau. Trả về danh sách gồm (n - 1) điểm chia cách nhau.

Ảnh có chứa văn bản, ảnh chụp màn hình, danh thiếp, Phông chữ

Mô tả được tạo tự động

* + Màn hình chạy:



* **zscore**(x): Tính Điểm chuẩn mô tả giá trị x dựa trên số độ lệch chuẩn so với giá trị trung bình của phân phối chuẩn: (x - mean) / stdev.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

* + Màn hình chạy:



CHƯƠNG 2 – THUẬT TOÁN CÂN BẰNG LƯỢC ĐỒ

2.1 Các vấn đề, ràng buộc/điều kiện, phương pháp/thuật toán.

* Các vấn đề về thuật toán cân bằng lược đồ.

- Cân bằng lược đồ (Histogram Equalization) là một phương pháp trong xử lý ảnh được sử dụng để cải thiện độ tương phản và sự phân bố độ sáng của ảnh. Phương pháp này áp dụng kỹ thuật cân bằng lược đồ vào ảnh đầu vào để cải thiện chất lượng hình ảnh và làm nổi bật các chi tiết bằng cách phân bổ đều độ chói trên hình ảnh.

* Ràng buộc/điều kiện.

- Phương pháp cân bằng lược đồ yêu cầu ảnh đầu vào phải là ảnh thang độ xám và không thể áp dụng trực tiếp lên ảnh màu. Đặc biệt, phương pháp này có hiệu quả là phạm vi độ sáng hình ảnh rộng bao gồm các mức độ sáng khác nhau từ đen đến trắng.

* Phương pháp/thuật toán.

- Tạo lược đồ của ảnh đầu vào và đếm số pixel có cùng giá trị thang độ xám.

- Tính xác suất mỗi mức độ sáng xuất hiện trong lược đồ bằng cách chia số pixel có cùng giá trị màu xám cho tổng số pixel trong ảnh.

- Xác định hàm chuyển đổi bằng cách tích lũy xác suất từ ​​mức độ sáng nhỏ đến lớn.

- Áp dụng chức năng chuyển đổi này cho ảnh gốc để cân bằng lược đồ và tạo ra ảnh cải tiến.

2.2 Các ví dụ.

Ảnh có chứa văn bản, ảnh chụp màn hình, Sơ đồ, biểu đồ

Mô tả được tạo tự độngẢnh có chứa mây, đen và trắng, phong cảnh, bầu trời

Mô tả được tạo tự động- Ảnh gốc và lược đồ:

Ảnh có chứa văn bản, ảnh chụp màn hình, Sơ đồ, hàng

Mô tả được tạo tự độngẢnh có chứa mây, phong cảnh, đen và trắng, bầu trời

Mô tả được tạo tự động- Ảnh đã được cân bằng và lược đồ:

2.3 Nhận xét, phân tích, đánh giá.

- Phương pháp cân bằng lược đồ là phương pháp đơn giản và hiệu quả để cải thiện chất lượng của hình ảnh.

- Tuy nhiên, phương pháp này cũng có các nhược điểm sau: phạm vi độ sáng quá lớn hoặc không phù hợp làm cho thông tin trong hình ảnh có thể bị mất.

- Trong một số trường hợp, phương pháp này có thể làm nổi bật nhiễu trong ảnh đã cân bằng lược đồ. Cần cân nhắc và lựa chọn kỹ lưỡng các phương pháp phù hợp cho từng loại hình ảnh và mục đích xử lý khác nhau.

CHƯƠNG 3 – THUẬT TOÁN PHÙ HỢP LƯỢC ĐỒ

3.1 Các vấn đề, ràng buộc/điều kiện, phương pháp/thuật toán.

* Các vấn đề về thuật toán cân bằng lược đồ.

- Phù hợp lược đồ (Histogram Matching) là một phương pháp trong xử lý ảnh để phân phối lược đồ của ảnh nguồn sao cho giống với phân phối lược đồ mong muốn. Phương pháp này cải thiện mức đồ tương phản cục bộ của ảnh, giảm hiện tượng mờ của các vùng tối, làm nổi bật lên các chi tiết của ảnh.

* Ràng buộc/điều kiện.

- Phương pháp cân bằng lược đồ yêu cầu ảnh đầu vào phải là ảnh thang độ xám và không thể áp dụng trực tiếp lên ảnh màu và chia ảnh thành các vùng nhỏ để xử lý riêng biệt. Ngoài ra, cần phải xác định ngưỡng độ lớn cửa sổ tính lược đồ.

* Phương pháp/thuật toán.

- Chia ảnh thành các khối nhỏ, thường là hình vuông.

- Tính lược độ cho từng khối, tính lược đồ riêng biệt chỉ trên các pixel thuộc khối đó.

- Áp dụng cân bằng lược đồ đề cải thiện mức độ tương phản cho khối đó.

- Ghép cái khối lại với nhau để tạo thành ảnh hoàn chỉnh.

- Lặp lại đối với nhiều kích cỡ khối khác nhau để xử lý nhiều cấp độ chi tiết.

- Kết hợp các kết quả từ các cấp độ khác nhau để tạo ra kết quả hoàn chỉnh nhất.

* 1. Các ví dụ.

Ảnh có chứa ngoài trời, người, đơn sắc, tòa nhà

Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Sơ đồ

Mô tả được tạo tự động- Ảnh gốc và lược đồ:

Ảnh có chứa động vật có vú, ngoài trời, động vật hoang dã, cỏ

Mô tả được tạo tự độngẢnh có chứa văn bản, biểu đồ, Sơ đồ, ảnh chụp màn hình

Mô tả được tạo tự động- Ảnh tham chiếu và lược đồ:

Ảnh có chứa văn bản, ảnh chụp màn hình, Sơ đồ, biểu đồ

Mô tả được tạo tự động- Ảnh đã được cân bằng và lược đồ:

Ảnh có chứa ngoài trời, trang phục, đen và trắng, tòa nhà

Mô tả được tạo tự động

3.3 Nhận xét, phân tích, đánh giá.

- Phương pháp cân bằng lược đồ là phương pháp đơn giản và hiệu quả để cải thiện chất lượng của hình ảnh, cải thiện được độ tương phạn của ảnh, làm nổi bật chi tiết, xử lý ảnh linh hoạt, tích ứng với nội dung ảnh, có thể kết hợp với nhiều kỹ thuật xử lý ảnh khác.

- Nếu tìm được ảnh mẫu tốt thì cho kết quả sẽ rất tốt. Tuy nhiên, nếu ảnh mẫu không được tốt lắm, kết quả chỉ như của biến đổi lược đồ xám thông thường, thậm chí không tốt bằng. Ngoài ra, tốn thời gian xử lý bởi vì phải xử lý nhiều lần, khó xác định kích thước khối, đôi khi hình ảnh khó xử lý dẫn đến là mất chi tiết ảnh.

CHƯƠNG 4 – THỰC HIỆN

4.1 Giải thích các dòng code trong chương 2 và 3.

* Thuật toán cân bằng lược đồ:
* Trong sourcecode có những phần sau:
* Đọc ảnh.
* Chuyển ảnh sang màu xám.
* Chuẩn hóa lược đồ.
* Tính hàm phân phối tích lũy.
* Hiển thị ảnh và lược đồ.
* Giải thích code trong file 522H0090\_Part2.py.

- Dòng 1-3: Import thư viện OpenCV, thư viện NumPy và thư viện Matplotlib.pyplot.

- Dòng 8: Đọc ảnh từ file “image.jpg” và lưu vào biến “img”.

- Dòng 11: Chuyển đổi ảnh từ không gian màu BGR sang ảnh xám và lưu vào biến “hinh\_xam”.

- Dòng 14: Lấy kích thước của ảnh bằng cách truy cập thuộc tính “shape” của ảnh. Biến “h\_cao” lưu chiều cao, “w\_rong” lưu chiều rộng của ảnh.

- Dòng 17: Tạo một mảng “temp” gồm 256 phần tử để lưu giá trị của ảnh. Mảng này sẽ được dùng để đếm tần số xuất hiện của mỗi giá trị cường độ trong ảnh xám.

- Dòng 20-22: Duyệt qua từng pixel của ảnh xám và đếm tần số của mỗi giá trị cường độ trong mảng.

- Dòng 25: Chuẩn hóa lược đồ bằng cách thực hiện chia từng giá trị trong mảng lược đồ cho tổng số pixel trong ảnh.

- Dòng 28-31: Tính hàm phân phối tích lũy của lược đồ bằng cách tích lũy các giá trị

trong mảng.

- Dòng 34: Nhân các giá trị trong mảng CDF với 255 để có các giá trị cường độ mới, từ 0 đến 255.

- Dòng 37: Tạo một mảng ảnh mới có kích thước giống ảnh gốc để lưu trữ ảnh đã được cân bằng.

- Dòng 38-40: Duyệt qua từng pixel của ảnh gốc, lấy giá trị cường độ của pixel và gán cho giá trị mới tương ứng từ mảng “GTM” vào mảng ảnh đã cân bằng.

- Dòng 43-45: Hiển thị ảnh gốc và ảnh đã cân bằng, bằng cửa sổ hiển thị của thư viện OpenCV.

- Dòng 48: Chuyển đổi ảnh gốc về dạng một chiều.

- Dòng 49: Đặt tên cho tiêu đề cho lược đồ của ảnh gốc.

- Dòng 50: Hiện thị lược đồ của ảnh gốc.

- Dòng 53: Chuyển đổi ảnh đã cân bằng về dạng một chiều.

- Dòng 54: Đặt tên cho tiêu đề cho lược đồ của ảnh đã cân bằng.

- Dòng 55: Hiện thị lược đồ của ảnh đã cân bằng.

* Thuật toán phù hợp lược đồ:
* Trong sourcecode có những phần sau:
* Đọc ảnh.
* Chuyển ảnh sang màu xám.
* Chuẩn hóa lược đồ.
* Tính hàm phân phối tích lũy.
* Ánh xạ ảnh.
* Hiển thị ảnh và lược đồ.
* Giải thích code trong file 522H0090\_Part3.py.

- Dòng 1-3: Import thư viện OpenCV, thư viện NumPy và thư viện Matplotlib.pyplot.

- Dòng 8-9: Đọc ảnh từ file “img1.jpg”, “img2.jpg” và lưu lần lượt vào biến “img1”, “img2”.

- Dòng 12-13: Chuyển đổi ảnh từ không gian màu BGR sang ảnh xám và lưu vào biến “hinh1\_xam” và “hinh2\_xam”.

- Dòng 16-17: Lấy kích thước của ảnh bằng cách truy cập thuộc tính “shape” của ảnh. Biến “h1\_cao” và “h2\_cao” lưu chiều cao, “w1\_rong” và “w2\_rong” lưu chiều rộng của ảnh.

- Dòng 20-21: Tạo một mảng “temp1 và temp2” gồm 256 phần tử để lưu giá trị của ảnh. Mảng này sẽ được dùng để đếm tần số xuất hiện của mỗi giá trị cường độ trong ảnh xám.

- Dòng 24-30: Duyệt qua từng pixel của ảnh xám và đếm tần số của mỗi giá trị cường

độ trong mảng.

- Dòng 33-34: Chuẩn hóa lược đồ bằng cách thực hiện chia từng giá trị trong mảng lược đồ cho tổng số pixel trong ảnh.

- Dòng 37-46: Tính hàm phân phối tích lũy của lược đồ bằng cách tích lũy các giá trị trong mảng.

- Dòng 49: Tạo một mảng “frame” gồm 256 phần tử và tất cả được khởi tạo bằng số 0.

- Dòng 50: Duyệt qua các giá trị từ 0 đến 255, lần lược xét phần tử “i” của cdf1.

- Dòng 51: Tạo biến “min” với giá trị dương vô cùng. Biến này sẽ theo dỗi sự chênh lệch tương đối nhỏ nhất của các phần tử cdf1 và cdf2.

- Dòng 52: Tạo biến “min\_val” với giá trị bằng 0. Biến này sẽ lưu trữ giá trị trong cdf2 khi có sự chênh lệch nhỏ nhất.

- Dòng 53: Duyệt qua các giá trị từ 0 đến 255, lần lược xét từng phần tử của “j” của cdf2.

- Dòng 54: Tính toán sự chênh lệch tuyệt đối giữa phần tử “i” trong cdf1 và phần tử “j” trong cdf2 sau đó lưu vào biến “valu”.

- Dòng 55-57: Kiểm tra sự chênh lệch tuyệt đối và nếu sự chênh lệch tuyệt đối nhỏ hơn giá trị tối thiểu thì sẽ gán giá trị tối thiểu thành sự chênh lệch tuyệt đối hiện tại.

- Dòng 58: Sau khi duyệt qua các phần tử trong cdf2 và gán giá trị vừa tìm được min\_val cho frame.

- Dòng 61: Tạo một mảng “m\_img” với cùng hình dạng như ảnh xám “hinh1\_xam” để lưu ảnh đã được cân bằng, tất cả các phần tử bằng 0. Kiểu dữ liệu là số nguyên không dấu 8 bits.

- Dòng 63: Duyệt qua từng pixel của ảnh gốc, lấy giá trị cường độ của pixel và gán cho giá trị mới tương ứng từ mảng “frame” vào mảng ảnh đã cân bằng.

- Dòng 69-71: Hiển thị ảnh gốc, ảnh tham chiếu và ảnh đã cân bằng, bằng cửa sổ hiển thị của thư viện OpenCV.

- Dòng 75: Chuyển đổi ảnh gốc về dạng một chiều.

- Dòng 76: Đặt tên cho tiêu đề cho lược đồ của ảnh gốc.

- Dòng 77: Hiện thị lược đồ của ảnh gốc.

- Dòng 80: Chuyển đổi ảnh tham chiếu về dạng một chiều.

- Dòng 81: Đặt tên cho tiêu đề cho lược đồ của ảnh tham chiếu.

- Dòng 82: Hiện thị lược đồ của ảnh ảnh tham chiếu.

- Dòng 85: Chuyển đổi ảnh đã cân bằng về dạng một chiều.

- Dòng 86: Đặt tên cho tiêu đề cho lược đồ của ảnh đã cân bằng.

- Dòng 87: Hiện thị lược đồ của ảnh đã cân bằng.

4.2 Các bước xây dựng và chạy sourcecode.

- Đảm bảo và chắc chắn đã cài đặt thư viện OpenCV và NumPy trên máy tính.

- Đổi tên file ảnh thành “image.jpg” , “img1.jpg”, “img2.jpg” và đặt file ảnh này cùng thư mục với file Python.

- Lưu và chạy chương trình Python.

- Đầu tiên hai ảnh hiển thị sẽ xuất hiện, một cửa sổ cho ảnh gốc và một cửa sổ cho ảnh đã cân bằng. Nhấn nút bất kì sẽ hiện ra lược đồ của ảnh gốc , tắt lược đồ của ảnh gốc sẽ hiện ra lược đồ của ảnh đã cân bằng.

- Tắt cửa sổ của lược đồ đã cân bằng sẽ xuất hiện ra ba ảnh (ảnh gốc, ảnh tham chiếu, ảnh đã cân bằng). Nhấn nút bất kì sẽ xuất hiện lược đồ của ảnh gốc, sau đó tắt cửa sổ đi sẽ xuất hiện lược đồ của tham chiếu và tương tự như vậy tắt cửa cửa sổ sẽ xuất hiện lược đồ của ảnh đã cân bằng.

4.3 Màng hình chạy.

* Thuật toán cân bằng lược đồ:

- Ảnh gốc: - Ảnh đã cân bằng

Ảnh có chứa mây, ngoài trời, đen và trắng, bầu trời

Mô tả được tạo tự độngẢnh có chứa mây, ngoài trời, đen và trắng, tòa nhà

Mô tả được tạo tự động

* Thuật toán phù hợp lược đồ:

Ảnh có chứa động vật có vú, ngoài trời, động vật hoang dã, cỏ

Mô tả được tạo tự độngẢnh có chứa ngoài trời, trang phục, đen và trắng, tòa nhà

Mô tả được tạo tự độngẢnh có chứa ngoài trời, người, đơn sắc, tòa nhà

Mô tả được tạo tự động- Ảnh gốc - Ảnh tham chiếu - Ảnh đã cân bằng

TÀI LIỆU THAM KHẢO

- Mathematical statistics functions:

<https://docs.python.org/3/library/statistics.html>

- Xử lí ảnh: thuật toán cân bằng histogram ảnh:

<https://viblo.asia/p/xu-li-anh-thuat-toan-can-bang-histogram-anh-GrLZDOogKk0>

**-** Tuần 3: Histogram - Histogram equalization:

<https://viblo.asia/p/tuan-3-histogram-histogram-equalization-3P0lPnxmKox>

- Histogram matching: <https://paulbourke.net/miscellaneous/equalisation/?fbclid=IwAR3W4quDV86oc-hktvWGPtaBQYHM3QdL_lxJgYxIaVvDAXsTAVLtcVpFxOc>

- Wilipedia- Histogram matching:

<https://en.wikipedia.org/wiki/Histogram_matching?fbclid=IwAR0T-EhaBEPXcnFB0OT13THGarFLIBuysmDY_qHgwEEpqkmm0qXbbwZeEuI>

- Thư viện OpenCV:

<https://www.mygreatlearning.com/blog/opencv-tutorial-in-python/>

- Thư viện NumPy:

<https://viblo.asia/p/tim-hieu-ve-thu-vien-numpy-trong-pythonphan-1-Do7542QXZM6>

<https://viblo.asia/p/tim-hieu-ve-thu-vien-numpy-trong-pythonphan-2-ORNZqernZ0n>

- Thư viện Matplotlib:

<https://www.geeksforgeeks.org/matplotlib-pyplot-hist-in-python/>  
<https://matplotlib.org/2.0.2/users/pyplot_tutorial.html>