TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



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

**KHẢO SÁT THƯ VIỆN STATISTICS TRONG PYTHON VÀ THUẬT TOÁN HISTOGRAM EQUALIZATION, HISTOGRAM MATCHING TRONG XỬ LÝ ẢNH**

*Người hướng dẫn***: ThS. Nguyễn Huỳnh Minh Duy**

*Người thực hiện*: **Võ Kim Long - 52200226**

Lớp: **22050301**

Khoá: **K26**

**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 GIỮA KÌ MÔN XÁC SUẤT VÀ THỐNG KÊ ỨNG DỤNG CHO CÔNG NGHỆ THÔNG TIN**

**KHẢO SÁT THƯ VIỆN STATISTICS TRONG PYTHON VÀ THUẬT TOÁN HISTOGRAM EQUALIZATION, HISTOGRAM MATCHING TRONG XỬ LÝ ẢNH**

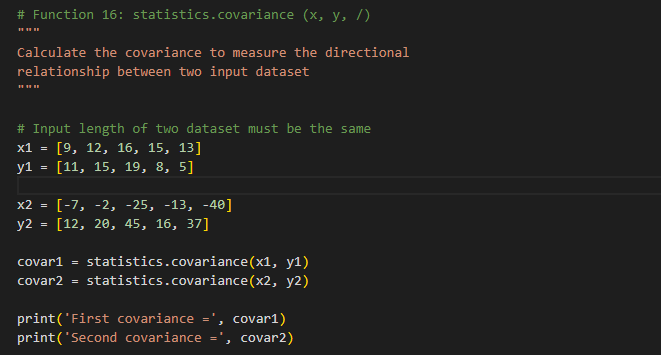
*Người hướng dẫn***: ThS. Nguyễn Huỳnh Minh Duy**

*Người thực hiện*: **Võ Kim Long - 52200226**

Lớp: **22050301**

Khoá: **K26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**



# LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn thầy Nguyễn Huỳnh Minh Duy đã tận tình giảng dạy và hướng dẫn em trong suốt thời gian môn học. Nhờ vào những kiến thức bổ ích và ví dụ thực tiễn mà thầy truyền đạt đã giúp em hiểu rõ hơn nội dung lí thuyết trên lớp và ứng dụng vào việc lập trình để kiểm nghiệm các bài toán. Ngoài ra, thầy cũng rất tận tâm trong việc hướng dẫn và trả lời thắc mắc để em có thể dễ dàng nắm bắt được bài học và sự nhiệt huyết của thầy trong giảng dạy chính là động lực để em hoàn thành bài tiểu luận này một cách tốt nhất. Em xin chân thành cảm ơn thầy.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của ThS. Nguyễn Huỳnh Minh Duy. 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.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng 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 chúng tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Võ Kim Long*

# 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 luận nhằm khảo sát các hàm, công thức trong xác suất thống kê thông qua thư viện Statistics của ngôn ngữ Python để hiểu rõ hơn về khái niệm và đặc trưng tìm ra giá trị đại diện cho bộ dữ liệu khác nhau. Bên cạnh đó, các công thức còn được ứng dụng vào thuật toán xử lí ảnh có độ tương phản thấp để cân bằng hoặc kết nối với biểu đồ khác để cho ra kết quả cân bằng hơn về màu sáng.

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc150462586)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc150462587)

[TÓM TẮT iv](#_Toc150462588)

[MỤC LỤC 1](#_Toc150462589)

[DANH MỤC HÌNH ẢNH, BẢNG BIỂU VÀ ĐỒ THỊ 3](#_Toc150462590)

[CHƯƠNG 1 - THƯ VIỆN STATISTICS NGÔN NGỮ PYTHON 6](#_Toc150462591)

[1.1 Hàm statistics.mean (data) 6](#_Toc150462592)

[1.2 Hàm statistics.fmean (data, weights = None) 7](#_Toc150462593)

[1.3 Hàm statistics.geometric\_mean (data) 8](#_Toc150462594)

[1.4 Hàm statistics.harmonic\_mean (data, weights = None) 9](#_Toc150462595)

[1.5 Hàm statistics.median (data) 10](#_Toc150462596)

[1.6 Hàm statistics.median\_low (data) 10](#_Toc150462597)

[1.7 Hàm statistics.median\_high (data) 11](#_Toc150462598)

[1.8 Hàm statistics.median\_grouped (data, interval = 1) 12](#_Toc150462599)

[1.9 Hàm statistics.mode (data) 13](#_Toc150462600)

[1.10 Hàm statistics.multimode (data) 13](#_Toc150462601)

[1.11 Hàm statistics.quantiles (data, \*, n = 4, method = ‘exclusive’) 14](#_Toc150462602)

[1.12 Hàm statistics.pstdev (data, mu = None) 15](#_Toc150462603)

[1.13 Hàm statistics.pvariance (data, mu = None) 16](#_Toc150462604)

[1.14 Hàm statistics.stdev (data, xbar = None) 17](#_Toc150462605)

[1.15 Hàm statistics.variance (data, xbar = None) 18](#_Toc150462606)

[1.16 Hàm statistics.covariance (x, y, /) 18](#_Toc150462607)

[1.17 Hàm statistics.correlation (x, y, /, \*, method = ‘linear’) 19](#_Toc150462608)

[1.18 Hàm statistics.linear\_regression (x, y, /, \*, proportional = False) 20](#_Toc150462609)

[CHƯƠNG 2 - THUẬT TOÁN HISTOGRAM EQUALIZATION 22](#_Toc150462610)

[2.1 Tổng quan về Histogram 22](#_Toc150462611)

[2.2 Đặt vấn đề: 22](#_Toc150462612)

[2.3 Về Histogram equalization 22](#_Toc150462613)

[2.4 Phương pháp tiếp cận và thuật toán 23](#_Toc150462614)

[2.5 Các công thức toán học 23](#_Toc150462615)

[2.6 Ví dụ mẫu: 24](#_Toc150462616)

[2.7 Kết luận, phân tích và đánh giá 26](#_Toc150462617)

[CHƯƠNG 3 - THUẬT TOÁN HISTOGRAM MATCHING 27](#_Toc150462618)

[3.1 Tổng quan về Histogram 27](#_Toc150462619)

[3.2 Đặt vấn đề 27](#_Toc150462620)

[3.3 Về Histogram matching 27](#_Toc150462621)

[3.4 Phương pháp tiếp cận và thuật toán 27](#_Toc150462622)

[3.5 Các công thức toán học 28](#_Toc150462623)

[3.6 Ví dụ mẫu 29](#_Toc150462624)

[3.7 Kết luận, phân tích và đánh giá 31](#_Toc150462625)

[CHƯƠNG 4 - CÀI ĐẶT THUẬT TOÁN 32](#_Toc150462626)

[4.1 Cài đặt thuật toán Histogram equalization 32](#_Toc150462627)

[4.2 Cài đặt thuật toán Histogram matching 37](#_Toc150462628)

[TÀI LIỆU THAM KHẢO 43](#_Toc150462629)

# DANH MỤC HÌNH ẢNH, BẢNG BIỂU VÀ ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1.1: Khai báo các thư viện cần thiết 6](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462630)

[Hình 1.2: Code mẫu hàm statistics.mean (data) 6](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462631)

[Hình 1.3: Lệnh in ra màn hình kết quả 7](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462632)

[Hình 1.4: Kết quả chạy 7](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462633)

[Hình 1.5: Code mẫu hàm statistics.fmean (data, weights = None) 7](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462634)

[Hình 1.6: Kết quả chạy 8](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462635)

[Hình 1.7: Code mẫu hàm statistics.geometric\_mean (data) 8](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462636)

[Hình 1.8: Kết quả chạy 8](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462637)

[Hình 1.9: Code mẫu hàm statistics.harmonic\_mean (data, weights = None) 9](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462638)

[Hình 1.10: Kết quả chạy 9](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462639)

[Hình 1.11: Code mẫu hàm statistics.median (data) 10](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462640)

[Hình 1.12: Kết quả chạy 10](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462641)

[Hình 1.13: Code mẫu hàm statistics.median\_low (data) 11](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462642)

[Hình 1.14: Kết quả chạy 11](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462643)

[Hình 1.15: Code mẫu hàm statistics.median\_high (data) 11](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462644)

[Hình 1.16: Kết quả chạy 12](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462645)

[Hình 1.17: Code mẫu hàm statistics.median\_grouped (data, interval = 1) 12](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462646)

[Hình 1.18: Kết quả chạy 12](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462647)

[Hình 1.19: Code mẫu hàm statistics.mode (data) 13](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462648)

[Hình 1.20: Kết quả chạy 13](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462649)

[Hình 1.21: Code mẫu hàm statistics.multimode (data) 14](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462650)

[Hình 1.22: Kết quả chạy 14](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462651)

[Hình 1.23: Code mẫu hàm statistics.quantiles (data, \*, n, method = ‘exclusive’) 15](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462652)

[Hình 1.24: Kết quả chạy 15](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462653)

[Hình 1.25: Code mẫu hàm statistics.pstdev (data, mu = None) 16](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462654)

[Hình 1.26: Kết quả chạy 16](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462655)

[Hình 1.27: Code mẫu hàm statistics.pvariance (data, mu = None) 17](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462656)

[Hình 1.28: Kết quả chạy 17](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462657)

[Hình 1.29: Code mẫu hàm statistics.stdev (data, mu = None) 17](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462658)

[Hình 1.30: Kết quả chạy 18](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462659)

[Hình 1.31: Code mẫu hàm statistics.variance (data, xbar = None) 18](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462660)

[Hình 1.32: Kết quả chạy 18](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462661)

[Hình 1.33: Code mẫu hàm statistics.covariance (x, y, /) 19](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462662)

[Hình 1.34: Kết quả chạy 19](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462663)

[Hình 1.35: Code mẫu hàm statistics.correlation (x, y, /, \*, method = ‘linear’) 20](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462664)

[Hình 1.36: Kết quả chạy 20](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462665)

[Hình 1.37: Code mẫu hàm statistics.linear\_regression(x, y, /, \*, protional = False) 21](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462666)

[Hình 1.38: Kết quả chạy 21](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462667)

[Hình 4.1: Gọi thư viện cần thiết 32](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462668)

[Hình 4.2: Đọc ảnh, tìm số lượng dòng và cột của mảng 32](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462669)

[Hình 4.3: Hàm tìm tần số của các giá trị pixel 33](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462670)

[Hình 4.4: Hàm tính phân phối xác suất 33](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462671)

[Hình 4.5: Hàm tính phân phối xác suất tích lũy 34](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462672)

[Hình 4.6: Hàm chuẩn hóa các giá trị pixel 34](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462673)

[Hình 4.7: Hàm ánh xạ pixel mới vào ảnh cũ 35](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462674)

[Hình 4.8: Gọi hàm thực hiện chuẩn hóa 35](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462675)

[Hình 4.9: Vẽ hình và biểu đồ bằng thư viện matplotlib 35](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462676)

[Hình 4.10: Trường hợp ảnh cho ra kết quả tốt 36](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462677)

[Hình 4.11: Trường hợp ảnh cho ra kết quả xấu 36](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462678)

[Hình 4.12: Gọi thư viện cần thiết 37](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462679)

[Hình 4.13: Đọc hình gốc và hình được tham chiếu 37](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462680)

[Hình 4.14: Hàm tìm tần số của các giá trị pixel 38](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462681)

[Hình 4.15: Hàm tính phân phối xác suất 38](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462682)

[Hình 4.16: Hàm tính phân phối xác suất tích lũy 39](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462683)

[Hình 4.17: Hàm chuẩn hóa các giá trị pixel 39](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462684)

[Hình 4.18: Hàm tìm ra giá trị pixel mới cho ảnh gốc 40](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462685)

[Hình 4.19: Hàm ánh xạ giá trị pixel mới vào ảnh gốc 41](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462686)

[Hình 4.20: Gọi hàm tính toán để thực hiện match 41](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462687)

[Hình 4.21: Trường hợp cho kết quả tốt 42](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462688)

[Hình 4.22: Trường hợp cho kết quả không tốt 42](file:///D:\Bo\Python%20Lab\Essay\52200226_VoKimLong.docx#_Toc150462689)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1: Bảng thống kê tần số thang màu 3-bit 23](#_Toc150426869)

[Bảng 2.2: Bảng tính xác suất phân bổ của các thang màu 23](#_Toc150426870)

[Bảng 2.3: Bảng phân bổ xác suất tích lũy của thang màu 24](#_Toc150426871)

[Bảng 2.4: Bảng tính giá trị thang màu mới 24](#_Toc150426872)

[Bảng 2.5: Ánh xạ thang màu mới vào bảng 25](#_Toc150426873)

[Bảng 3.1: Giá trị pixel hình 1st 28](#_Toc150426874)

[Bảng 3.2: Giá trị pixel hình 2nd 28](#_Toc150426875)

[Bảng 3.3: Phân phối xác suất hình 1st 28](#_Toc150426876)

[Bảng 3.4: Phân phối xác suất hình 2nd 28](#_Toc150426877)

[Bảng 3.5: Xác suất tích lũy hình 1st 29](#_Toc150426878)

[Bảng 3.6: Xác suất tích lũy hình 2nd 29](#_Toc150426879)

[Bảng 3.7: Giá trị pixel mới hình 1st 29](#_Toc150426880)

[Bảng 3.8: Giá trị pixel mới hình 2nd 30](#_Toc150426881)

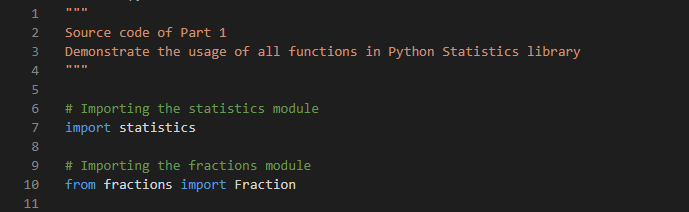
[Bảng 3.9: Hình 1st đã chuẩn hóa 30](#_Toc150426882)

[Bảng 3.10: Hình 2nd đã chuẩn hóa 30](#_Toc150426883)

[Bảng 3.11: Kết quả Pixel mới sau khi matching 30](#_Toc150426884)

# THƯ VIỆN STATISTICS NGÔN NGỮ PYTHON

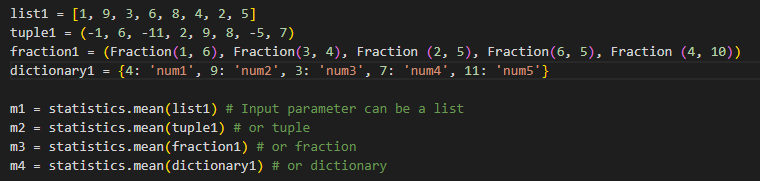
*Tổng quan:* Thư viện Statistics là một module được xây dựng sẵn và cho ra mắt vào phiên bản Python 3.4, nhằm cung cấp các hàm tính toán thống kê cho dữ liệu là số thực, đặc biệt là trong thống kê mô tả như trung bình, trung vị, phương sai, độ lệch chuẩn, số mode.. Dưới đây là khai báo các thư viện cần thiết nhằm cho việc mô tả chi tiết của các hàm trong thư viện trên.



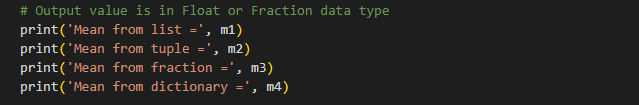
Hình .: Khai báo các thư viện cần thiết

## 1.1 Hàm statistics.mean (data)

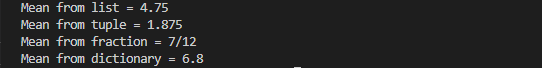
* Công dụng: Là hàm dùng để tính giá trị trung bình của bộ số liệu
* Tham số đầu vào:
  + Data: Danh sách số thực chứa bộ số liệu cần tính
* Ý nghĩa giá trị trả về: Là giá trị trung bình dùng để xem xét giá trị đại diện của bộ dữ liệu
* Ngoại lệ: Lỗi StatisticsError khi truyền tham số là danh sách rỗng
* Công thức: (1.1)



Hình .: Code mẫu hàm statistics.mean (data)



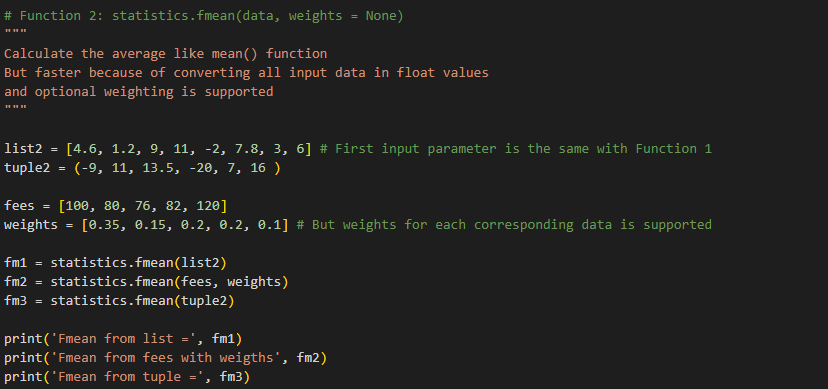
Hình .: Lệnh in ra màn hình kết quả



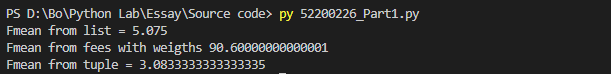
Hình .: Kết quả chạy

## 1.2 Hàm statistics.fmean (data, weights = None)

* Công dụng, cách tính và ý nghĩa giá trị trả về tương tự hàm ở 1.1
* Điểm khác biệt:
  + Thời gian chạy nhanh hơn, danh sách truyền vào được chuyển thành kiểu float và giá trị trả về cũng luôn là float
  + Tham số đầu vào:
    - Data: Danh sách số thực chứa bộ số liệu cần tính
    - Weights: Trọng số tương ứng với mỗi giá trị trong danh sách
  + Ngoại lệ: Lỗi StatisticsError khi truyền danh sách rỗng và lỗi ValueError khi số lượng giá trị và số lượng trọng số không bằng nhau
  + Công thức: (1.2)



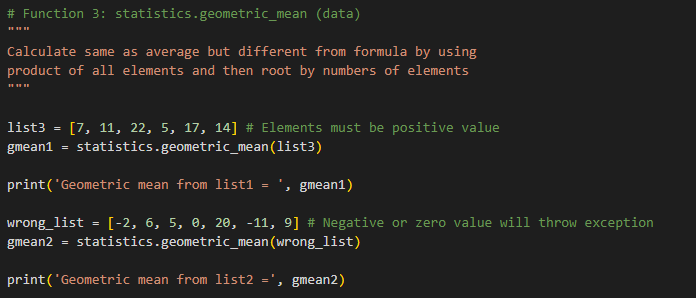
Hình .: Code mẫu hàm statistics.fmean (data, weights = None)



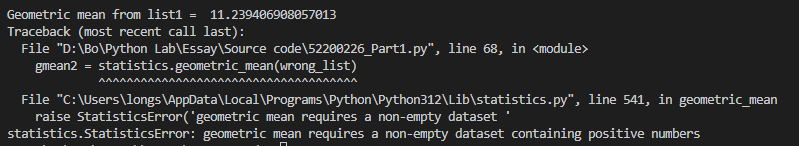
Hình .: Kết quả chạy

## 1.3 Hàm statistics.geometric\_mean (data)

* Công dụng: Là hàm dùng để tính số trung bình nhân, thường được sử dụng trong trường hợp số liệu tăng trưởng qua từng giai đoạn như lãi suất
* Tham số đầu vào:
  + Data: Danh sách số thực chứa bộ số liệu cần tính
* Ý nghĩa giá trị trả về: Là giá trị đại diện cho bộ số liệu mà có tính chất cấp số nhân hoặc hàm mũ
* Ngoại lệ: Lỗi StatisticsError khi truyền danh sách rỗng, hoặc có chứa số 0, hoặc có chứa số âm (Hình 1.8)
* Công thức: (1.3)



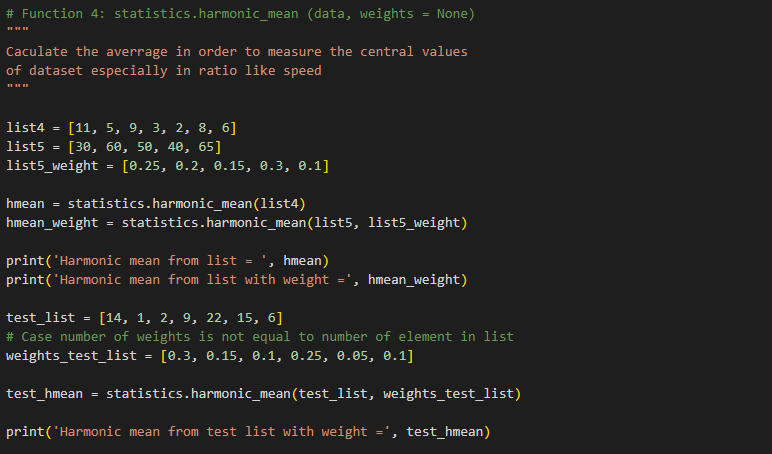
Hình .: Code mẫu hàm statistics.geometric\_mean (data)



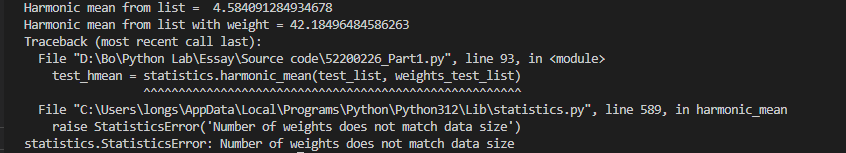
Hình .: Kết quả chạy

## 1.4 Hàm statistics.harmonic\_mean (data, weights = None)

* Công dụng: Là hàm dùng để tính số trung bình điều hòa, thường được sử dụng khi bộ giá trị là tỉ số của hai đơn vị khác nhau (vận tốc km/h, m/s, ..)
* Tham số đầu vào:
  + Data: Danh sách số thực chứa bộ số liệu cần tính
  + Weights: Trọng số tương ứng với mỗi giá trị trong danh sách
* Ý nghĩa giá trị trả về: Để đo lường giá trị đại diện cho bộ dữ liệu có giá trị thuộc dạng tỉ số hai đơn vị khác nhau
* Ngoại lệ: Lỗi StatisticsError truyền danh sách rỗng, phần tử giá trị bé hơn 0, hoặc số lượng giá trị không bằng số lượng trọng số truyền vào (Hình 1.10)
* Công thức: (1.4)



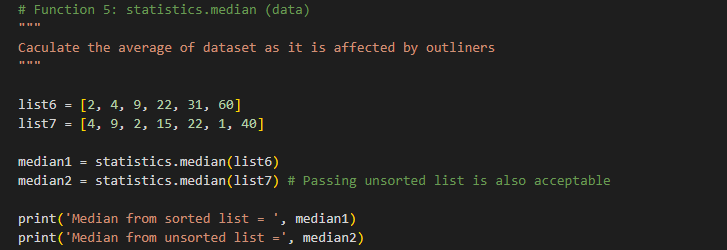
Hình .: Code mẫu hàm statistics.harmonic\_mean (data, weights = None)



Hình .: Kết quả chạy

## 1.5 Hàm statistics.median (data)

* Công dụng: Là hàm để tính giá trị trung vị của bộ số liệu khi có giá trị ngoại lai ảnh hưởng (giá trị quá lớn hoặc quá bé) làm lệch đi giá trị trung bình
* Tham số đầu vào:
  + Data: Danh sách số thực chứa bộ số liệu cần tính
* Ý nghĩa giá trị trả về: Là thước đo dùng để tìm giá trị đại diện cho bộ dữ liệu khi có giá trị ngoại lai ảnh hưởng
* Ngoại lệ: Lỗi StatisticsError khi truyền tham số là danh sách rỗng
* Công thức: (1.5)



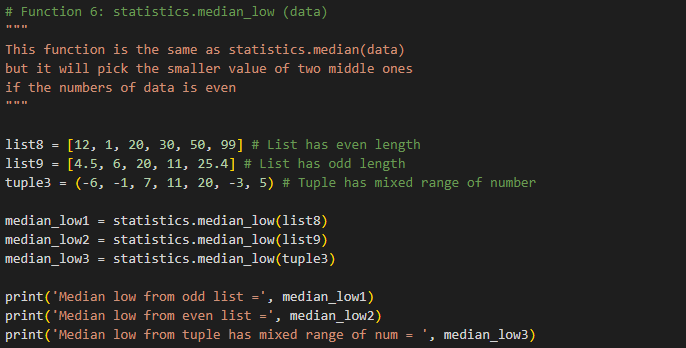
Hình .: Code mẫu hàm statistics.median (data)



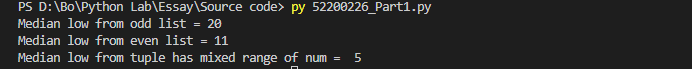
Hình .: Kết quả chạy

## 1.6 Hàm statistics.median\_low (data)

* Công dụng, tham số đầu vào, ý nghĩa giá trị trả về và ngoại lệ tương tự 1.5
* Điểm khác biệt:
  + Công thức: Nếu n là lẻ thì trung vị là số nằm giữa, ngược lại thì trung vị là số nhỏ hơn giữa hai số nằm giữa



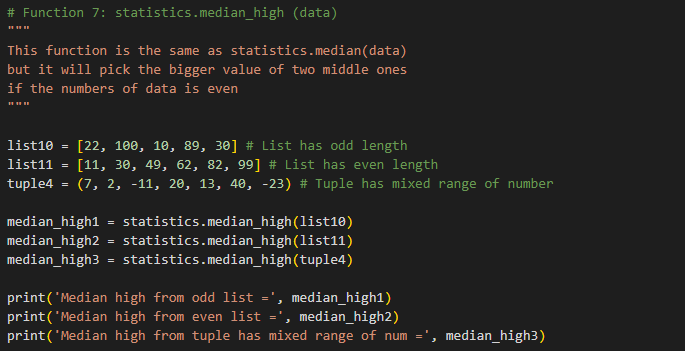
Hình .: Code mẫu hàm statistics.median\_low (data)



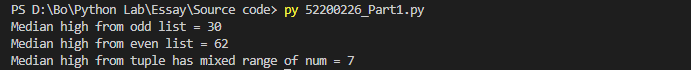
Hình .: Kết quả chạy

## 1.7 Hàm statistics.median\_high (data)

* Công dụng, tham số đầu vào, ý nghĩa giá trị trả về và ngoại lệ tương tự 1.5
* Điểm khác biệt:
  + Công thức: Nếu n là lẻ thì trung vị là số nằm giữa, ngược lại thì trung vị là số lớn hơn giữa hai số nằm giữa



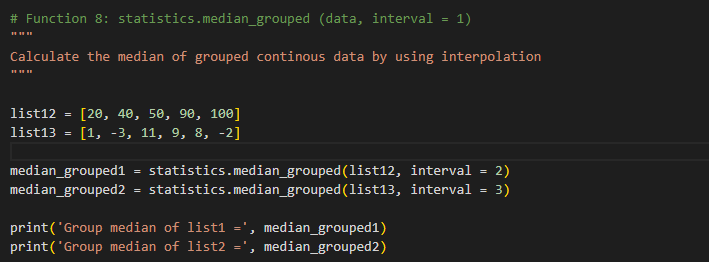
Hình .: Code mẫu hàm statistics.median\_high (data)



Hình .: Kết quả chạy

## 1.8 Hàm statistics.median\_grouped (data, interval = 1)

* Công dụng: Là hàm dùng để tính trung vị của bộ dữ liệu liên tục được nhóm
* Tham số đầu vào:
  + Data: Danh sách số thực chứa bộ số liệu cần tính
  + Interval: Xác định độ rộng của dữ liệu được nhóm
* Ý nghĩa giá trị trả về: Trung vị của nhóm dữ liệu liên tục bằng cách tính nội suy
* Ngoại lệ: Lỗi StatisticsError khi truyền vào danh sách rỗng
* Công thức: (1.6)



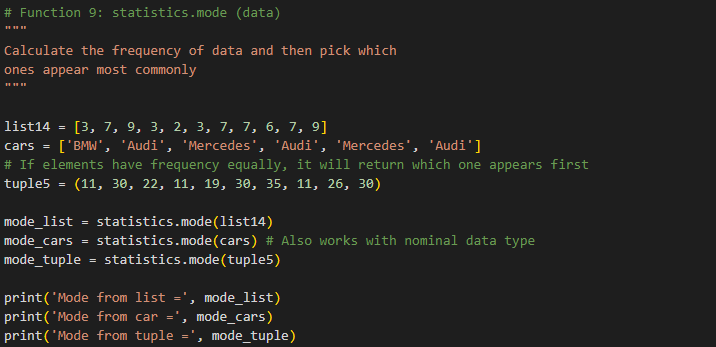
Hình .: Code mẫu hàm statistics.median\_grouped (data, interval = 1)



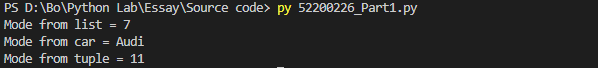
Hình .: Kết quả chạy

## 1.9 Hàm statistics.mode (data)

* Công dụng: Tính số mode của bộ giá trị
* Tham số đầu vào:
  + Data: Danh sách chứa số thực hoặc không phải số (non – numeric)
* Ý nghĩa giá trị trả về: Để đo lường giá trị đại diện cho bộ số liệu
* Ngoại lệ: Lỗi StatisticsError khi danh sách truyền vào là rỗng
* Công thức: Đếm số lần xuất hiện của các giá trị và chọn ra giá trị đầu tiên có số lần xuất hiện nhiều nhất



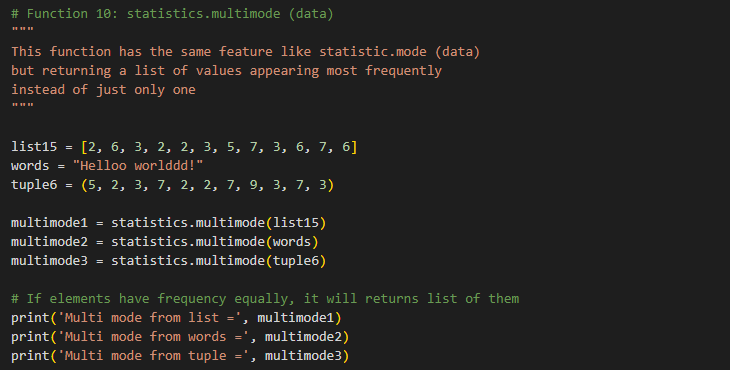
Hình .: Code mẫu hàm statistics.mode (data)



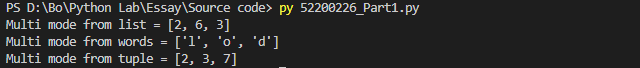
Hình .: Kết quả chạy

## 1.10 Hàm statistics.multimode (data)

* Công dụng, tham số đầu vào, ý nghĩa giá trị trả về và ngoại lệ tương tự 1.9
* Điểm khác biệt:
  + Công thức: Nếu có nhiều giá trị cùng tần số xuất hiện thì sẽ trả về một mảng chứa các giá trị đó thay vì chỉ giá trị đầu tiên



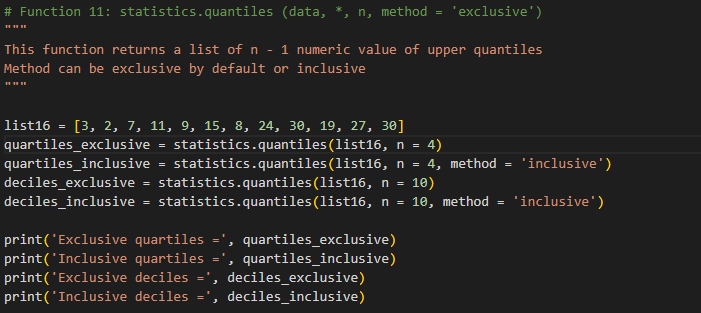
Hình .: Code mẫu hàm statistics.multimode (data)



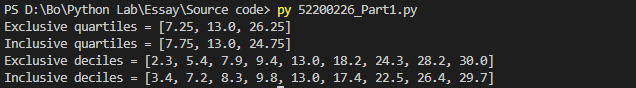
Hình .: Kết quả chạy

## 1.11 Hàm statistics.quantiles (data, \*, n = 4, method = ‘exclusive’)

* Công dụng: Là hàm dùng để tính số phân vị của bộ dữ liệu
* Tham số đầu vào:
  + Data: Danh sách chứa bộ số liệu cần tính
  + n: Số lượng phân vị mong muốn
  + method: phương pháp ‘exlusive’ cho dữ liệu lấy tập mẫu từ tổng thể mà có cực trị nhiều hơn tập mẫu được lấy hoặc phương pháp ‘inclusive’ cho tổng thể hoặc tập mẫu mà có nhiều cực trị nhất
* Ý nghĩa giá trị trả về: Dùng để xác định vị trí tương đối của một giá trị so với giá trị khác
* Ngoại lệ: Lỗi StatisticsError khi n bé hơn 1, hoặc số lượng phần tử trong data bé hơn 2



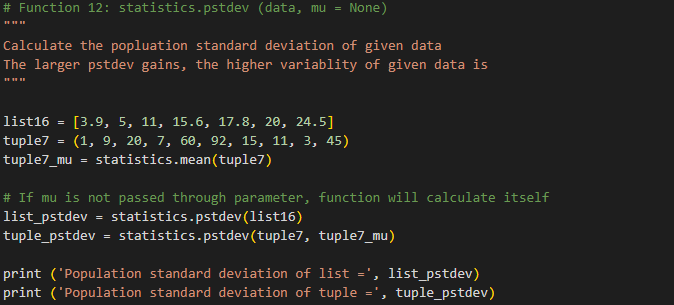
Hình .: Code mẫu hàm statistics.quantiles (data, \*, n, method = ‘exclusive’)



Hình .: Kết quả chạy

## 1.12 Hàm statistics.pstdev (data, mu = None)

* Công dụng: Là hàm dùng để tính độ lệch chuẩn của tổng thể
* Tham số đầu vào:
  + Data: Danh sách chứa bộ số liệu cần tính
  + Mu: Giá trị trung bình mean (ở 1.1)
  + Nếu không truyền Mu thì hàm sẽ tự động tính và truyền vào
* Ý nghĩa giá trị trả về: Thể hiện độ biến thiên của giá trị, độ lệch chuẩn càng lớn thì độ biến thiên càng cao
* Ngoại lệ: Lỗi StatisticsError khi truyền danh sách rỗng
* Công thức: (1.7)



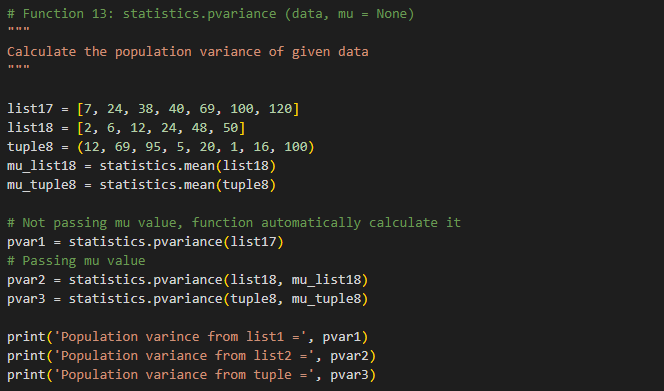
Hình .: Code mẫu hàm statistics.pstdev (data, mu = None)



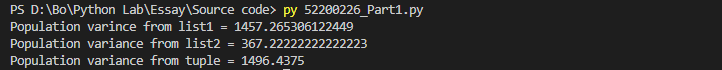
Hình .: Kết quả chạy

## 1.13 Hàm statistics.pvariance (data, mu = None)

* Công dụng: Là hàm dùng để tính phương sai của tổng thể
* Tham số đầu vào:
  + Data: Danh sách chứa bộ số liệu cần tính
  + Mu: Giá trị trung bình mean (ở 1.1)
  + Nếu không truyền Mu thì hàm sẽ tự động tính và truyền vào
* Ý nghĩa giá trị trả về: Dùng để đo lường độ biến thiên của dữ liệu, phương sai càng lớn độ phân tán dữ liệu càng cao, ngược lại dữ liệu càng gần giá trị trung bình (mean)
* Ngoại lệ: Lỗi StatisticsError khi truyền danh sách rỗng
* Công thức: : (1.8)



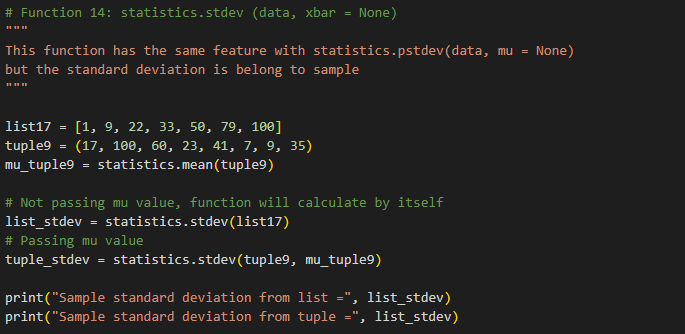
Hình .: Code mẫu hàm statistics.pvariance (data, mu = None)



Hình .: Kết quả chạy

## 1.14 Hàm statistics.stdev (data, xbar = None)

* Tham số đầu vào, ý nghĩa giá trị trả về và ngoại lệ tương tự hàm ở 1.12
* Điểm khác biệt: Tính độ lệch chuẩn cho tập mẫu thay vì tổng thể
  + Công thức: (1.9)

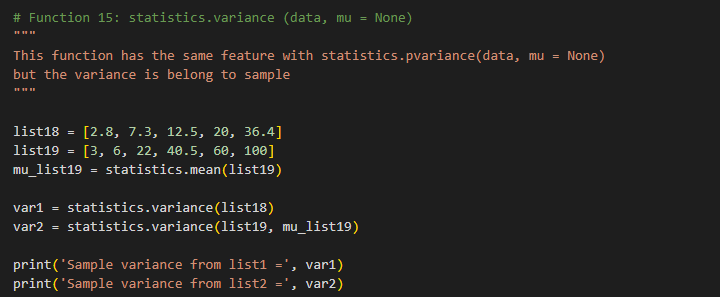


Hình .: Code mẫu hàm statistics.stdev (data, mu = None)



Hình .: Kết quả chạy

## 1.15 Hàm statistics.variance (data, xbar = None)

* Tham số đầu vào, ý nghĩa giá trị trả về và ngoại lệ tương tự hàm ở 1.13
* Điểm khác biệt: Tính phương sai cho dữ liệu là tập mẫu thay vì tổng thể
  + Công thức: (1.9)

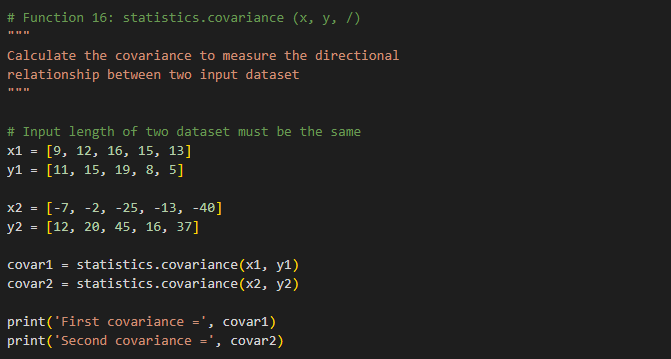
Hình .: Code mẫu hàm statistics.variance (data, xbar = None)



Hình .: Kết quả chạy

## 1.16 Hàm statistics.covariance (x, y, /)

* Công dụng: Là hàm dùng để tính hiệp phương sai của hai bộ dữ liệu
* Tham số đầu vào:
  + x và y: Tương ứng với hai bộ dữ liệu cần tính
* Ý nghĩa giá trị trả về: Đo lường mối quan hệ giữa hai bộ dữ liệu, nếu giá trị trả về là dương thì hai bộ thuận chiều, nếu là âm thì hai bộ theo chiều nghịch, nếu bằng 0 thì cả hai bộ đều độc lập, không bị tác động lẫn nhau
* Ngoại lệ: Lỗi StatisticsError khi số phần tử hai bộ dữ liệu không bằng nhau
* Công thức: (1.10)



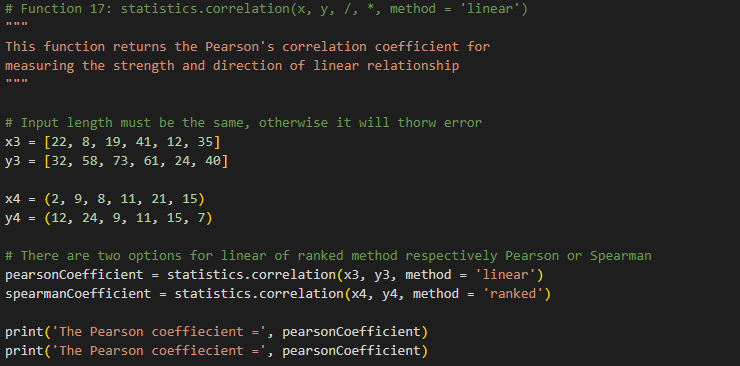
Hình .: Code mẫu hàm statistics.covariance (x, y, /)



Hình .: Kết quả chạy

## 1.17 Hàm statistics.correlation (x, y, /, \*, method = ‘linear’)

* Công dụng: Là hàm dùng để trả về hệ số tương quan của hai bộ số liệu
* Tham số đầu vào:
  + x và y: Tương ứng với hai bộ số liệu cần tính
  + method: Phương pháp ‘linear’ với cách tính của hệ số tương đối Pearson, còn ‘ranked’ tương ứng với cách tính của hệ số tương đối Spearman
* Ý nghĩa giá trị trả về: Dùng để đo lường cường độ và hướng của mối quan hệ tuyến tính, giá trị nhận được sẽ trong đoạn [-1, 1]
* Ngoại lệ: Lỗi StatisticsError khi số phần tử hai bộ dữ liệu không bằng nhau hoặc là hằng số



Hình .: Code mẫu hàm statistics.correlation (x, y, /, \*, method = ‘linear’)

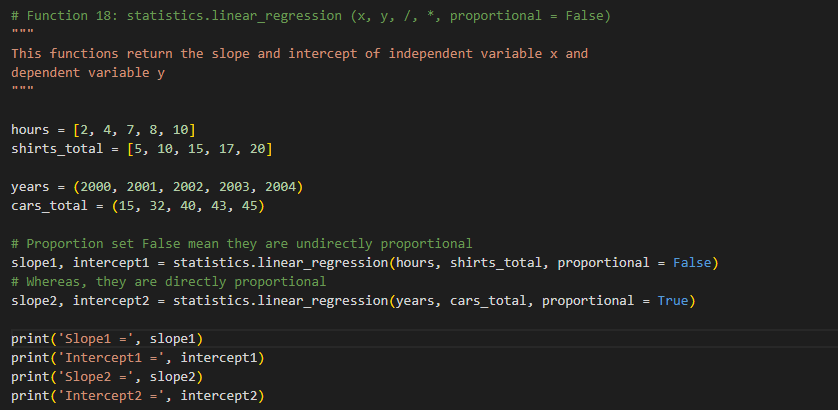


Hình .: Kết quả chạy

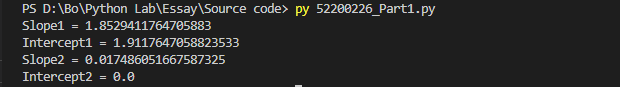
## 1.18 Hàm statistics.linear\_regression (x, y, /, \*, proportional = False)

* Công dụng: Là hàm dùng để tính độ dốc và hệ số chặn của hồi quy tuyến tính x và y
* Tham số đầu vào:
  + x : là danh sách chứa biến độc lập
  + y : là danh sách chứa biến phụ thuộc
  + proportional = True biến x và y tỉ lệ thuận, False thì x và y tỉ lệ nghịch
* Ý nghĩa giá trị trả về: Dùng để miêu tả mối quan hệ giữa biến độc lập x và biến phụ thuộc y dưới dạng hàm tuyến tính
* Ngoại lệ: Lỗi StatisticsError khi số phần tử hai bộ dữ liệu không bằng nhau hoặc biến độc lập x là hằng số
* Công thức: (1.11)

Intercept = b = (1.12)



Hình .: Code mẫu hàm statistics.linear\_regression(x, y, /, \*, protional = False)



Hình .: Kết quả chạy

# THUẬT TOÁN HISTOGRAM EQUALIZATION

## 2.1 Tổng quan về Histogram

- Một bức ảnh kỹ thuật số bản chất là một mảng 2 chiều trong đó tại mỗi điểm x và y (x tương ứng với trục ngang hay là chiều dài ảnh và y tương ứng với trục đứng hay là chiều rộng ảnh) sẽ cho ra một pixel

- Mỗi pixel đều có giá trị pixel riêng của nó để phản ánh cường độ ánh sáng tại vị trí đó

- Như vậy, nếu đưa các giá trị pixel đó vào Histogram (biểu đồ) thì ta sẽ được một biểu đồ tần số với trục x là các giá trị pixel và trục y là tần số của pixel đó thế nên Histogram sẽ cho biết độ phủ của các cường độ ánh sáng

## 2.2 Đặt vấn đề:

- Hình ảnh chụp được đôi khi có độ tương phản thấp dẫn đến nhiều chi tiết trong ảnh không được sắc nét và khó nhận diện vật thể

- Hình ảnh bị quá sáng hoặc quá tối làm mất đi màu sắc và tính tự nhiên của ảnh

- Hình ảnh chụp trong y tế như chụp CT, X quang quá tối và không đủ rõ để chẩn đoán bệnh

## 2.3 Về Histogram equalization

- Là kỹ thuật được sử dụng rộng rãi trong việc cải thiện độ tương phản của ảnh bằng cách sử dụng Histogram để cân bằng lại các cường độ điểm ảnh, nhờ vậy sẽ không còn xuất hiện điểm ảnh có cường độ quá cao hoặc quá thấp

- Điểm mạnh: Dễ thực hiện và tính toán hiệu quả, tăng độ tương phản trên toàn bộ ảnh, thích hợp xử lí những ảnh có độ tương phản thấp

- Điểm yếu: Có thể xuất hiện nhiễu và làm mất đi những chi tiết nhỏ nếu ảnh có độ tương phản quá cao (quá sáng hoặc quá tối)

## 2.4 Phương pháp tiếp cận và thuật toán

- Bước 1: Chuyển ảnh về các giá trị pixel

- Bước 2: Đếm số lần xuất hiện của các giá trị pixel

- Bước 3: Xác định L (với L là số lượng giá trị pixel khác nhau)

- Bước 4: Tính phân bổ xác suất (PMF) của các giá trị pixel

- Bước 5: Tính phân bổ xác suất tích lũy (CDF)

- Bước 6: Nhân các giá trị ở bước 4 với (L – 1) và làm tròn số

- Bước 7: Gán lại các giá trị pixel mới (ở bước 6) tương ứng lại bảng

## 2.5 Các công thức toán học

- Công thức tính phân bổ xác suất (PMF):

(.)

Trong đó: là xác suất phân bổ của giá trị i

là tần số của giá trị i

là tổng số lượng giá trị được thống kê

là số lượng giá trị khác nhau

- Công thức tính phân bổ xác suất tích lũy (CDF):

(.)

Trong đó: là xác suất phân bổ tích lũy của giá trị i

là xác suất phân bổ của giá trị j

- Công thức tính giá trị pixel mới cho từng điểm ảnh:

(.)

Trong đó: là giá trị pixel mới

là tần số tích lũy của giá trị i

là số lượng giá trị khác nhau

là làm tròn số lên 1 đơn vị

## 2.6 Ví dụ mẫu:

- Giả sử một bức ảnh 3-bit có thể biểu diễn được 8 thang màu (2 mũ 3) và được liệt kê tần số tương ứng như sau:

Bảng .: Bảng thống kê tần số thang màu 3-bit

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thang màu | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Số pixel | 0 | 100 | 400 | 50 | 200 | 50 | 200 | 0 |

- Tổng số lượng giá trị n = 0 + 100 + 400 + 50 + 200 + 50 + 200 + 0 = 1000

- Số lượng giá trị khác nhau L = 9 (0 đến 7)

- Tính xác suất phân bổ của từng giá trị:

Bảng .: Bảng tính xác suất phân bổ của các thang màu

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thang màu | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Số pixel | 0 | 100 | 400 | 50 | 200 | 50 | 200 | 0 |
|  | 0 | 0.1 | 0.4 | 0.05 | 0.2 | 0.05 | 0.2 | 0 |

- Tính xác suất phân bổ tích lũy của từng giá trị dựa vào bảng 2.2:

+ = 0

+ = + = 0.1

+ = + + = 0.5

+ = + + + = 0.55

+ = + + + + = 0.75

+ = + + + + + = 0.8

+ = + + + + + + = 1

+ tương tự = 1

Bảng .: Bảng phân bổ xác suất tích lũy của thang màu

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thang màu | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Số pixel | 0 | 100 | 400 | 50 | 200 | 50 | 200 | 0 |
|  | 0 | 0.1 | 0.4 | 0.05 | 0.2 | 0.05 | 0.2 | 0 |
|  | 0 | 0.1 | 0.5 | 0.55 | 0.75 | 0.8 | 1 | 1 |

- Tính giá trị thang màu mới và làm tròn:

+ = = 0 \* 7 = 0

+ = = 0.1 \* 7 = 0.7 = 1

+ = = 0.5 \* 7 = 3.5 = 4

+ = = 0.55 \* 7 = 3.85 = 4

+ = = 0.75 \* 7 = 5.25 = 5

+ = = 0.8 \* 7 = 5.6 = 6

+ = = 1 \* 7 = 7

+ = = 1 \* 7 = 7

Bảng .: Bảng tính giá trị thang màu mới

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thang màu | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Số pixel | 0 | 100 | 400 | 50 | 200 | 50 | 200 | 0 |
|  | 0 | 0.1 | 0.4 | 0.05 | 0.2 | 0.05 | 0.2 | 0 |
|  | 0 | 0.1 | 0.5 | 0.55 | 0.75 | 0.8 | 1 | 1 |
|  | 0 | 0.7 | 3.5 | 3.85 | 5.25 | 5.6 | 7 | 7 |
|  | 0 | 1 | 4 | 4 | 5 | 6 | 7 | 7 |

- Ánh xạ giá trị thang màu mới vào bảng (nếu trong mảng 2 chiều thì ánh xạ vào điểm ảnh pixel)

Bảng .: Ánh xạ thang màu mới vào bảng

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thang màu | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Số pixel | 0 | 100 | 400 | 50 | 200 | 50 | 200 | 0 |
| Thang màu mới | 0 | 1 | 4 | 4 | 5 | 6 | 7 | 7 |

- Như vậy thang màu 0 có tần số là 0, thang màu 1 có tần số là 100, thang màu 4 có tần số là 400 + 50 = 450, thang màu 5 có tần số là 200, thang màu 6 có tần số là 50 và thang màu 7 có tần số là 200 + 0 = 200

## 2.7 Kết luận, phân tích và đánh giá

- Nếu Histogram của ảnh lệch trái thì ảnh sẽ có nhiều điểm tối

- Nếu Histogram của ảnh lệch phải thì ảnh sẽ có nhiều điểm sáng

- Nếu Histogram dạng hình chuông tập trung ở giữa thì ảnh sẽ có độ tương phản thấp

- Nếu Histogram trải đều các giá trị thì ảnh sẽ có độ tương phản cao

- Nhìn chung, sau khi áp dụng thuật toán Histogram equalization thì ảnh sẽ được cải thiện độ tương phản và hình dạng của Histogram được cân bằng

- Tuy nhiên ở những vùng tối giá trị điểm ảnh sẽ không thay đổi nhiều nhưng vùng sáng thì giá trị điểm sáng sẽ được tích lũy tăng dần

- Do đó Histogram equalization sẽ phù hợp để xử lí những ảnh có cường độ điểm sáng giới hạn tại một vùng cụ thể, thay vì phủ rộng các vùng (cả sáng cả tối)

# THUẬT TOÁN HISTOGRAM MATCHING

## 3.1 Tổng quan về Histogram

- Tương tự ở mục 2.1

## 3.2 Đặt vấn đề

- Hình ảnh không được thống nhất về mặt tương phản

- Cần so sánh về sự phân phối về màu sắc của nhiều ảnh khác nhau

- Phát hiện biến đổi của khu vực thông qua ảnh chụp từ vệ tinh bằng cách so sánh Histogram chụp được từ nhiều thời điểm khác nhau

- Đảm bảo các hình ảnh giống nhau sẽ có cùng sự phân bố về màu sắc khi được chụp từ bộ cảm biến khác nhau

## 3.3 Về Histogram matching

- Histogram matching là quá trình thay đổi độ tương phản của ảnh bằng cách dựa vào Histogram của ảnh đó và chuyển đổi phù hợp với Histogram của một ảnh khác

- Trong đó, Histogram equalization (ở chương 2) là một trường hợp đặc biệt của Histogram matching vì Histogram của ảnh sẽ mặc định chuẩn hóa về phân bố đều thay vì dựa vào Histogram của ảnh khác

- Nhưng kỹ thuật này có tính chủ động hơn, giả sử muốn một hình A có thêm sự phân phối về màu sắc dựa vào hình B thay vì phân phối đều như Histogram equalization

## 3.4 Phương pháp tiếp cận và thuật toán

- Bước 1: Chuyển hình ảnh về các giá trị pixel

- Bước 2: Tính tần số của các giá trị pixel

- Bước 3: Tính xác suất phân bổ (PMF)

- Bước 4: Tính xác suất phân bổ tích lũy (CDF)

- Bước 5: Thực hiện chuẩn hóa Histogram (như ở chương 2) để có giá trị pixel mới

- Bước 6: Lặp lại tương tự các bước trên với hình được tham chiếu

- Bước 7: Dựa vào giá trị pixel tìm được của cả hai hình (Bước 5) để ánh xạ cho ra giá trị pixel mới cho hình ban đầu

## 3.5 Các công thức toán học

- Công thức tính phân bổ xác suất (PMF):

(3.)

Trong đó: là xác suất phân bổ của giá trị i

là tần số của giá trị i

là tổng số lượng giá trị được thống kê

là số lượng giá trị khác nhau

- Công thức tính phân bổ xác suất tích lũy (CDF) của hình ban đầu:

(3.)

Trong đó: là xác suất phân bổ tích lũy của giá trị i

là xác suất phân bổ của giá trị j trong hình ban đầu

- Công thức tính phân bổ xác suất tích lũy (CDF) của hình được tham chiếu:

(3.3)

Trong đó: là xác suất phân bổ tích lũy của giá trị i

là xác suất phân bổ giá trị j trong hình được tham chiếu

- Công thức tìm ra giá trị pixel mới:

(3.4)

Trong đó: là xác suất phân bổ tích lũy giá trị i trong hình ban đầu

là xác suất phân bổ giá trị j trong hình được tham chiếu

- Trong trường hợp không tìm thấy được giá trị bằng nhau, có thể lấy giá trị gần bằng nhất, nếu có cả hai giá trị cùng gần bằng nhất thì lấy giá trị lớn hơn

## 3.6 Ví dụ mẫu

- Giả sử có hai bức hình 5x5 với các giá trị pixel từ 0 đến 4 như sau:

Bảng .: Giá trị pixel hình 1st

Bảng .: Giá trị pixel hình 2nd

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 4 | 1 | 3 | 2 |
| 4 | 1 | 3 | 4 | 3 |
| 0 | 1 | 3 | 1 | 4 |
| 0 | 2 | 4 | 1 | 3 |
| 0 | 4 | 2 | 4 | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 1 | 2 |
| 2 | 4 | 3 | 3 | 4 |
| 0 | 3 | 2 | 1 | 0 |
| 1 | 3 | 2 | 4 | 4 |
| 1 | 3 | 0 | 4 | 1 |

- Tính phân bổ xác suất (PMF) của hình 1st và 2nd

|  |  |  |
| --- | --- | --- |
| Giá trị pixel | Tần số | PMF |
| 0 | 4 | 0.16 |
| 1 | 5 | 0.2 |
| 2 | 3 | 0.12 |
| 3 | 5 | 0.2 |
| 4 | 8 | 0.32 |

Bảng .: Phân phối xác suất hình 1st

|  |  |  |
| --- | --- | --- |
| Giá trị pixel | Tần số | PMF |
| 0 | 4 | 0.16 |
| 1 | 6 | 0.24 |
| 2 | 5 | 0.2 |
| 3 | 5 | 0.2 |
| 4 | 5 | 0.2 |

Bảng .: Phân phối xác suất hình 2nd

- Tính phân phối xác suất tích lũy (CDF) từng hình dựa vào phân phối xác suất:

+ = 0.16

+ = + = 0.16 + 0.2 = 0.36

+ = + = 0.36 + 0.12 = 0.48

+ = + = 0.48 + 0.2 = 0.68

+ = + = 0.68 + 0.32 = 1

- Tính tương tự với hình 2nd ta sẽ có bảng thống kê như bên dưới

Bảng .: Xác suất tích lũy hình 1st

Bảng .: Xác suất tích lũy hình 2nd

|  |  |  |  |
| --- | --- | --- | --- |
| Giá trị pixel | Tần số | PMF | CDF |
| 0 | 4 | 0.16 | 0.16 |
| 1 | 5 | 0.2 | 0.36 |
| 2 | 3 | 0.12 | 0.48 |
| 3 | 5 | 0.2 | 0.68 |
| 4 | 8 | 0.32 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Giá trị pixel | Tần số | PMF | CDF |
| 0 | 4 | 0.16 | 0.16 |
| 1 | 6 | 0.24 | 0.4 |
| 2 | 5 | 0.2 | 0.6 |
| 3 | 5 | 0.2 | 0.8 |
| 4 | 5 | 0.2 | 1 |

- Tính giá trị thang màu mới và làm tròn (như ở chương 2):

+ = = 0.16 \* 4 = 0.64 = 1

+ = = 0.36 \* 4 = 1.44 = 1

+ = = 0.48 \* 4 = 1.92 = 2

+ = = 0.68 \* 4 = 2.72 = 3

+ = = 1 \* 4 = 4

Bảng .7: Giá trị pixel mới hình 1st

Bảng .8: Giá trị pixel mới hình 2nd

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Giá trị pixel | Tần số | PMF | CDF | Round |
| 0 | 4 | 0.16 | 0.16 | 1 |
| 1 | 5 | 0.2 | 0.36 | 1 |
| 2 | 3 | 0.12 | 0.48 | 2 |
| 3 | 5 | 0.2 | 0.68 | 3 |
| 4 | 8 | 0.32 | 1 | 4 |

- Lặp lại tương tự bước trên với hình 2d ta sẽ có bảng như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Giá trị pixel | Tần số | PMF | CDF | Round |
| 0 | 4 | 0.16 | 0.16 | 1 |
| 1 | 6 | 0.24 | 0.4 | 2 |
| 2 | 5 | 0.2 | 0.6 | 2 |
| 3 | 5 | 0.2 | 0.8 | 3 |
| 4 | 5 | 0.2 | 1 | 4 |

- Thực hiện matching bằng cách chọn Round 2 sao cho bằng với Round 1 hoặc gần bằng. Nếu có nhiều giá trị đều gần bằng thì chọn cái lớn hơn (như hình dưới).

Bảng .: Hình 1st đã chuẩn hóa

Bảng .: Hình 2nd đã chuẩn hóa

|  |  |
| --- | --- |
| Giá trị pixel  (Hình 1st) | Round (Hình 1st) |
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |

|  |  |
| --- | --- |
| Round (Hình 2nd) | Giá trị pixel  (Hình 2nd) |
| 1 | 0 |
| 2 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |

- Thực hiện matching (như bảng trên). Như vậy, ta sẽ có giá trị mới pixel mới:

+ Pixel 0 giữ nguyên

+ Pixel 1 ánh xạ với Pixel 0

+ Pixel 2 ánh xạ với Pixel 1

+ Pixel 3 giữ nguyên

Bảng .: Kết quả Pixel mới sau khi matching

+ Pixel 4 giữ nguyên

|  |  |
| --- | --- |
| Giá trị pixel  (Hình 1st cũ) | Giá trị pixel  (Hình 1st đã ánh xạ) |
| 0 | 0 |
| 1 | 0 |
| 2 | 1 |
| 3 | 3 |
| 4 | 4 |

## 3.7 Kết luận, phân tích và đánh giá

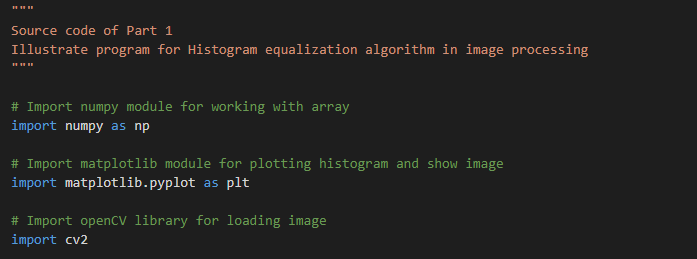
- Phương pháp Histogram matching phụ thuộc vào ảnh được tham chiếu, nếu ảnh được tham chiếu có độ phân phối màu sắc tốt thì kết quả sẽ tốt, ngược lại sẽ làm ảnh ban đầu tệ hơn. Nếu mục đích nhằm tăng độ tương phản của một bức ảnh có độ tương phản thấp thì Histogram equalization sẽ tốt hơn.

# CÀI ĐẶT THUẬT TOÁN

Tổng quan: Thực hiện code thuật toán Histogram equalization và matching bằng ngôn ngữ Python với các thư viện hỗ trợ như openCV, numpy, matplotlib..

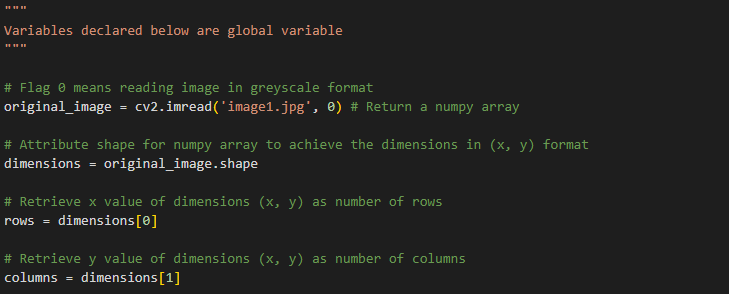
## 4.1 Cài đặt thuật toán Histogram equalization

* Gọi thư viện numpy để xử lý mảng, thư viện matplotlib để vẽ biểu đồ Histogram và hiện ảnh lên màn hình, thư viện openCV để đọc ảnh



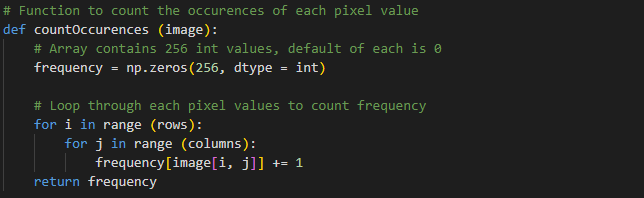
Hình .: Gọi thư viện cần thiết

* Đọc ảnh bằng hàm imread() tham số là 0 để đọc theo thang màu xám, kết quả trả về sẽ là mảng 2 chiều chứa các giá trị pixel tương ứng tại mỗi điểm sáng
* Tìm số lượng dòng và cột (x, y) của mảng 2 chiều bằng np.shape và lấy tại index [0] để lấy số dòng, index [1] để lấy số cột



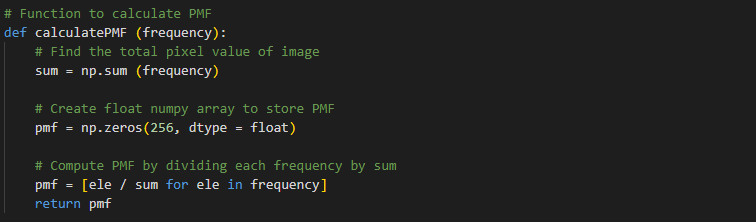
Hình .: Đọc ảnh, tìm số lượng dòng và cột của mảng

* Định nghĩa hàm tìm tần số của các giá trị pixel:
  + Tham số đầu vào là mảng image chứa các giá trị pixel của ảnh
  + Tạo mảng chứa số nguyên frequency có độ dài là 256 để lưu tần số của các giá trị pixel (Do giá trị pixel thuộc [0,255] )
  + Duyệt mảng image, giá trị pixel bằng bao nhiêu thì mảng frequency tại vị trí đó sẽ cộng 1



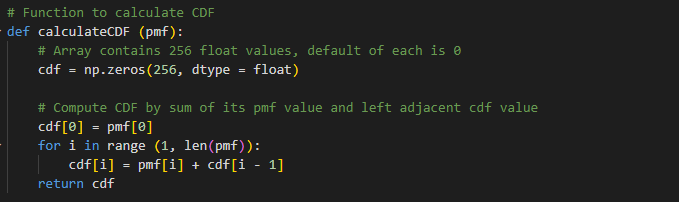
Hình .: Hàm tìm tần số của các giá trị pixel

* Định nghĩa hàm tìm phân phối xác suất:
  + Tham số đầu vào là mảng frequency chứa tần số của các giá trị pixel
  + Tìm số lượng giá trị pixel n bằng hàm np.sum()
  + Tạo mảng chứa số thực pmf có độ dài là 256 để lưu xác suất phân phối của các giá trị pixel (Do giá trị pixel thuộc [0,255] )
  + Lấy các giá trị tần số chia cho tổng để ra xác suất phân phối của giá trị pixel tương ứng



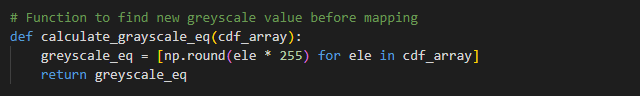
Hình .: Hàm tính phân phối xác suất

* Định nghĩa hàm tìm phân phối xác suất tích lũy:
  + Tham số đầu vào là mảng pmf chứa phân phối xác suất
  + Tạo mảng chứa số thực pmf có độ dài là 256 để lưu xác suất phân phối của các giá trị pixel (Do giá trị pixel thuộc [0,255] )
  + Duyệt qua mảng pmf để tính bằng cách lấy xác suất phân phối của giá trị pixel đó (pmf [i]) cộng với xác suất phân phối tích lũy của giá trị trước đó (cdf [i – 1])



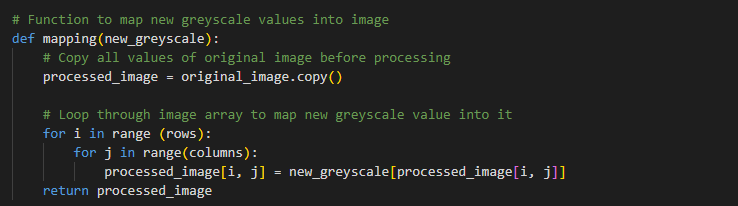
Hình .: Hàm tính phân phối xác suất tích lũy

* Định nghĩa hàm để chuẩn hóa giá trị pixel:
  + Tham số đầu vào là mảng cdf chứa xác suất phân phối tích lũy
  + Lấy từng cdf của giá trị pixel (cdf [i]) \* 255 (do L - 1) và làm tròn sẽ cho ra giá trị pixel mới của pixel thứ i



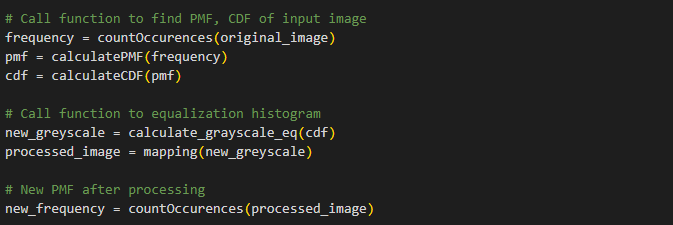
Hình .: Hàm chuẩn hóa các giá trị pixel

* Định nghĩa hàm ánh xạ pixel mới vào mảng của ảnh chứa giá trị pixel cũ:
  + Tham số truyền vào là mảng new\_grey\_scale chứa giá trị pixel mới
  + Tạo mảng processed\_image là bản sao của mảng chứa giá trị pixel ảnh gốc bằng np.copy() để chứa giá trị pixel mới
  + Duyệt qua mảng processed\_image, lấy pixel thứ i và xem giá trị của mảng new\_grayscale tại vị trí thứ i là bao nhiêu để ánh xạ



Hình .: Hàm ánh xạ pixel mới vào ảnh cũ

* Gọi hàm để thực hiện các câu lệnh chuẩn hóa trên



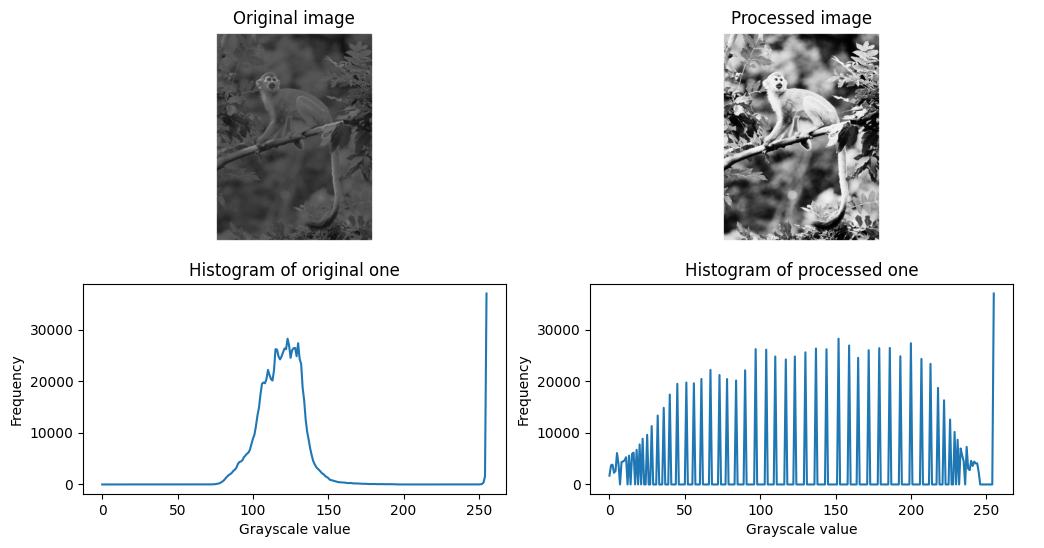
Hình .: Gọi hàm thực hiện chuẩn hóa

* Vẽ biểu đồ và hiện ảnh trước và sau khi xử lí với các hàm tiêu biểu:
  + plt.subplot( ) để chia thành các vùng nhỏ trong cửa sổ
  + plt.imshow( ) để hiện ảnh lên màn hình
  + plt.plot( ) để vẽ biểu đồ Histogram

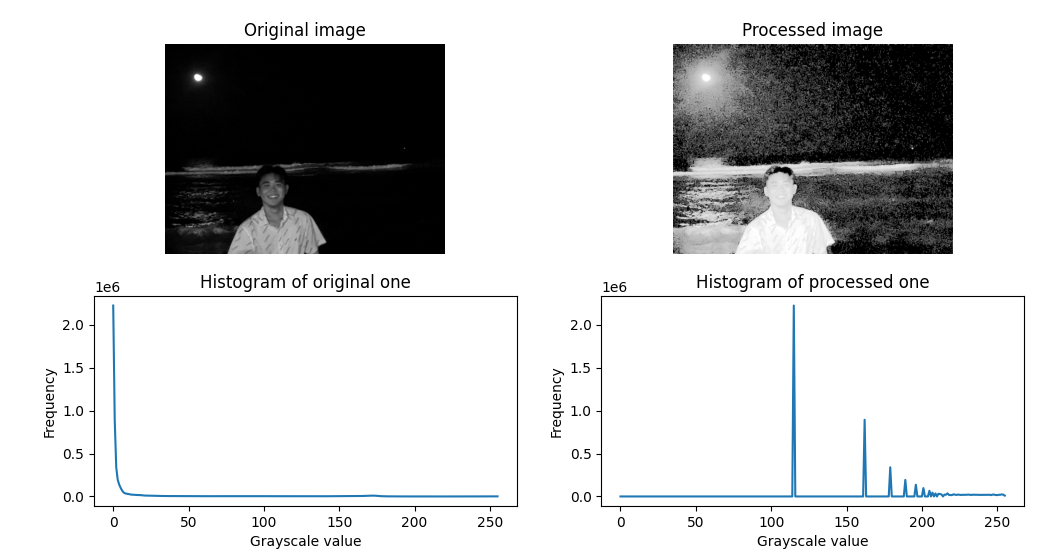


Hình .: Vẽ hình và biểu đồ bằng thư viện matplotlib

* Kết quả sau khi xử lí:



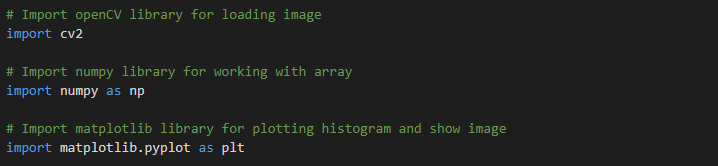
Hình .: Trường hợp ảnh cho ra kết quả tốt



Hình .: Trường hợp ảnh cho ra kết quả xấu

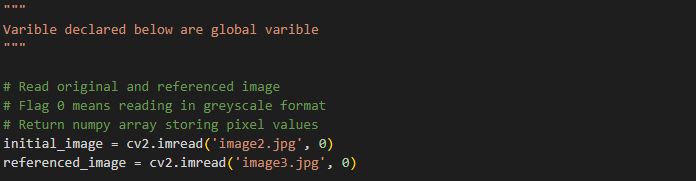
## 4.2 Cài đặt thuật toán Histogram matching

* Gọi thư viện numpy để xử lý mảng, thư viện matplotlib để vẽ biểu đồ Histogram và hiện ảnh lên màn hình, thư viện openCV để đọc ảnh



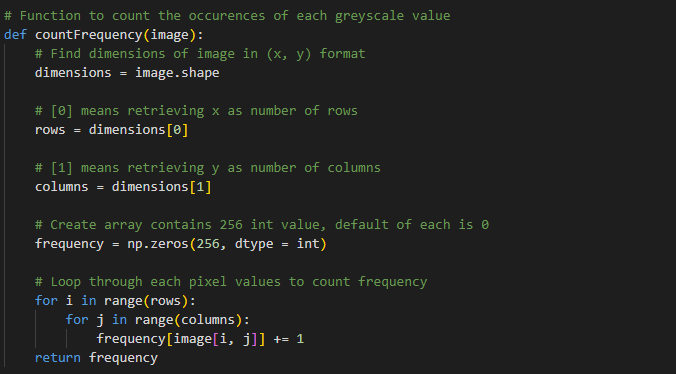
Hình .: Gọi thư viện cần thiết

* Đọc ảnh bằng hàm imread() tham số là 0 để đọc theo thang màu xám, kết quả trả về sẽ là mảng 2 chiều chứa các giá trị pixel tương ứng tại mỗi điểm sáng



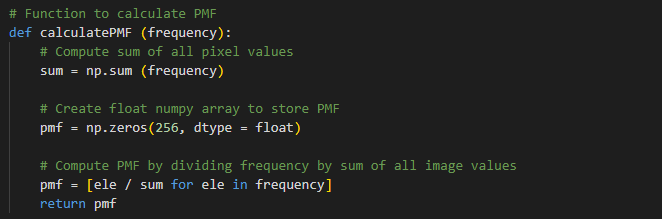
Hình .: Đọc hình gốc và hình được tham chiếu

* Định nghĩa hàm tìm tần số của các giá trị pixel:
  + Tham số đầu vào là mảng image chứa các giá trị pixel của ảnh
  + Tìm số lượng dòng và cột (x, y) của mảng 2 chiều bằng np.shape và lấy tại index [0] để lấy số dòng, index [1] để lấy số cột
  + Tạo mảng chứa số nguyên frequency có độ dài là 256 để lưu tần số của các giá trị pixel (Do giá trị pixel thuộc [0,255] )
  + Duyệt mảng image, giá trị pixel bằng bao nhiêu thì mảng frequency tại vị trí đó sẽ cộng 1



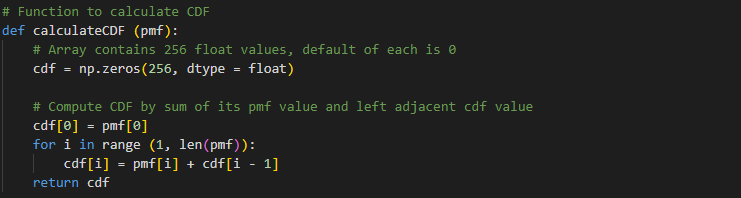
Hình .: Hàm tìm tần số của các giá trị pixel

* Định nghĩa hàm tìm phân phối xác suất:
  + Tham số đầu vào là mảng frequency chứa tần số của các giá trị pixel
  + Tìm số lượng giá trị pixel n bằng hàm np.sum()
  + Tạo mảng chứa số thực pmf có độ dài là 256 để lưu xác suất phân phối của các giá trị pixel (Do giá trị pixel thuộc [0,255] )
  + Lấy các giá trị tần số chia cho tổng để ra xác suất phân phối của giá trị pixel tương ứng



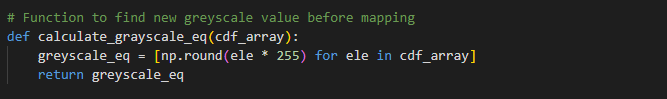
Hình .: Hàm tính phân phối xác suất

* Định nghĩa hàm tìm phân phối xác suất tích lũy:
  + Tham số đầu vào là mảng pmf chứa phân phối xác suất
  + Tạo mảng chứa số thực pmf có độ dài là 256 để lưu xác suất phân phối của các giá trị pixel (Do giá trị pixel thuộc [0,255] )
  + Duyệt qua mảng pmf để tính bằng cách lấy xác suất phân phối của giá trị pixel đó (pmf [i]) cộng với xác suất phân phối tích lũy của giá trị trước đó (cdf [i – 1])



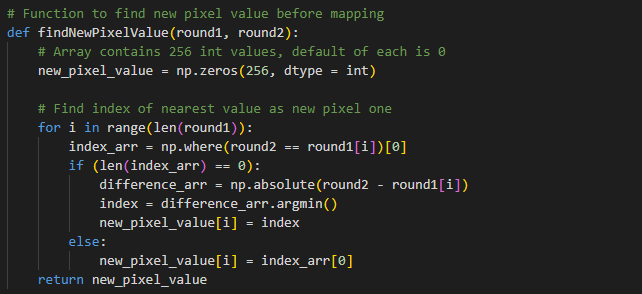
Hình .: Hàm tính phân phối xác suất tích lũy

* Định nghĩa hàm để chuẩn hóa giá trị pixel:
  + Tham số đầu vào là mảng cdf chứa xác suất phân phối tích lũy
  + Lấy từng cdf của giá trị pixel (cdf [i]) \* 255 (do L - 1) và làm tròn sẽ cho ra giá trị pixel mới của pixel thứ i



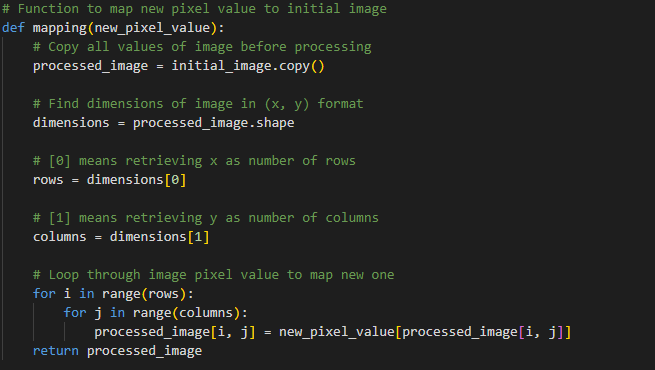
Hình .: Hàm chuẩn hóa các giá trị pixel

* Định nghĩa hàm tìm giá trị pixel mới:
  + Tham số đầu vào là các giá trị pixel đã chuẩn hóa (round) của hai hình
  + Tạo mảng new\_pixel\_value chứa số nguyên có độ dài là 256 để lưu các giá trị pixel mới của ảnh gốc (Do giá trị pixel thuộc [0,255] )
  + Duyệt qua từng round của của hình gốc, và chọn round2 của hình được tham chiếu có giá trị bằng, hoặc gần bằng nhất và pixel của round2 được tham chiếu tới sẽ là giá trị pixel mới cho hình gốc
  + Hàm np.where() để trả về mảng các index (tức giá trị pixel) có round2 bằng với giá trị cụ thể round1
  + Nếu có index, tức có giá trị bằng thì lấy index nhỏ nhất (index\_arr[0]), nếu không có thì chọn ra index có giá trị pixel gần bằng nhất với hàm np.absolute() và np.argmin()
  + Hàm np.absolute() để lấy trị tuyệt để sau khi tìm khoảng cách
  + Hàm np.argmin() để lấy index giá trị đầu tiên nhỏ nhất trong mảng



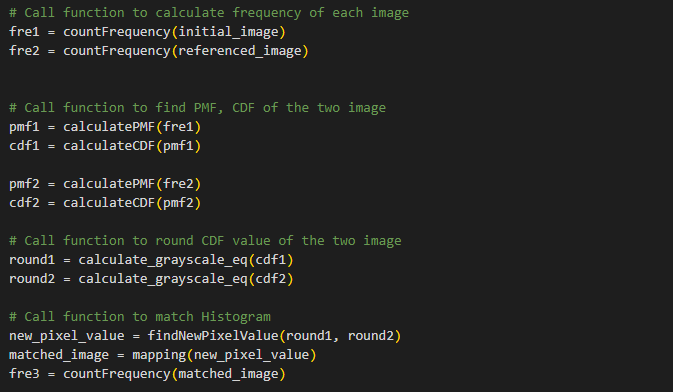
Hình .: Hàm tìm ra giá trị pixel mới cho ảnh gốc

* Định nghĩa hàm ánh xạ pixel mới vào mảng của ảnh gốc:
  + Tạo mảng processed\_image là bản sao của mảng chứa giá trị pixel ảnh gốc bằng np.copy() để chứa giá trị pixel mới
  + Duyệt qua mảng processed\_image, lấy pixel thứ i và xem giá trị của mảng new\_grayscale tại vị trí thứ i là bao nhiêu để ánh xạ



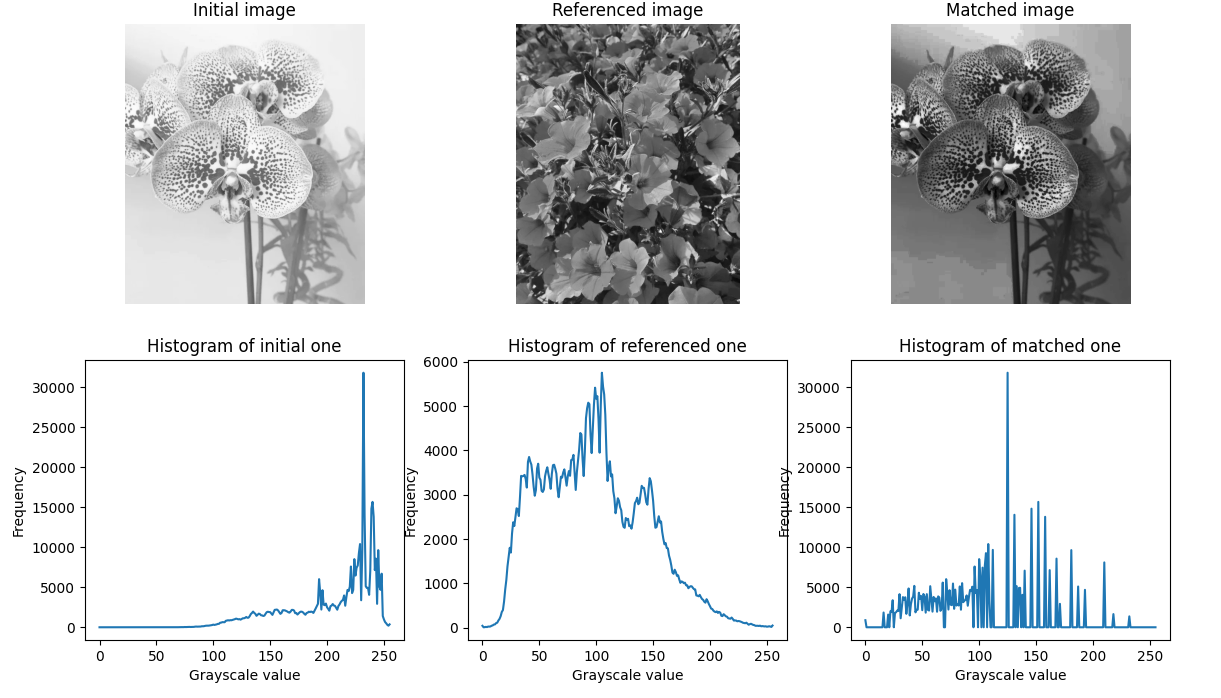
Hình .: Hàm ánh xạ giá trị pixel mới vào ảnh gốc

* Gọi hàm để thực hiện các câu lệnh chuẩn hóa trên

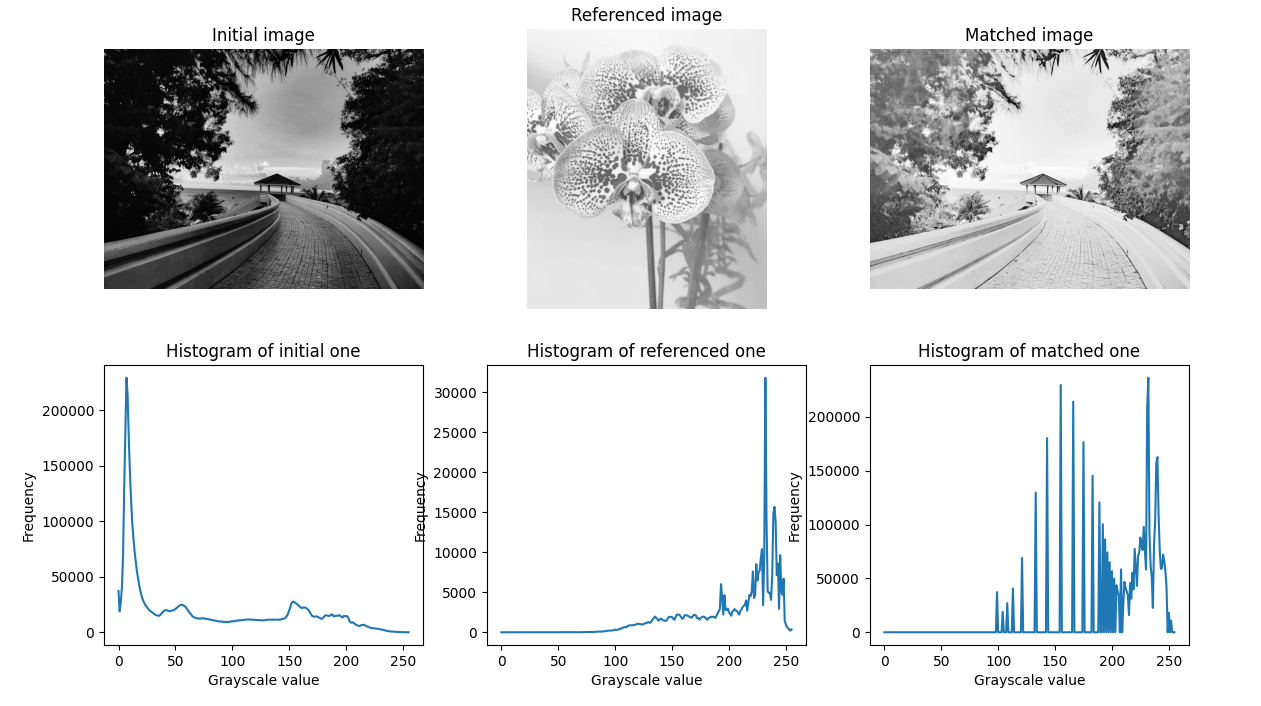


Hình .: Gọi hàm tính toán để thực hiện match

* Kết quả chạy



Hình .: Trường hợp cho kết quả tốt



Hình .: Trường hợp cho kết quả không tốt

# TÀI LIỆU THAM KHẢO

**Tiếng Anh**

1. *Histogram Matching for 8 \* 8 images.* (n.d.). Retrieved November 9, 2023, from https://www.ques10.com/p/66012/histogram-matching-for-8-8-images/
2. Joshi, N. (2022, January 28). Histogram Equalization. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2022/01/histogram-equalization/
3. Kang & Atul. (2019, April 10). Histogram Matching (Specification). *TheAILearner*. https://theailearner.com/2019/04/10/histogram-matching-specification/
4. *Python statistics Module*. (n.d.). Retrieved November 9, 2023, from https://www.w3schools.com/python/module\_statistics.asp
5. Statistics with Python. (2021, July 22). *GeeksforGeeks*. https://www.geeksforgeeks.org/statistics-with-python/
6. *Statistics.quantiles | Interactive Chaos*. (n.d.). Retrieved November 9, 2023, from https://interactivechaos.com/en/python/function/statisticsquantiles
7. *What is the statistics quantiles() method in Python?* (n.d.). Educative. Retrieved November 9, 2023, from https://www.educative.io/answers/what-is-the-statistics-quantiles-method-in-python
8. *What is the statistics.covariance() method in Python?* (n.d.). Educative. Retrieved November 9, 2023, from https://www.educative.io/answers/what-is-the-statisticscovariance-method-in-python
9. *What is the statistics.linear\_regression() method in Python?* (n.d.). Educative. Retrieved November 9, 2023, from https://www.educative.io/answers/what-is-the-statisticslinearregression-method-in-python