

ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA TOÁN - TIN HỌC



21TTH - KDL

PHƯƠNG PHÁP SỐ CHO KHOA HỌC DỮ LIỆU

BÁO CÁO CUỐI KÌ

Xử lý khử nhiễu hình ảnh bằng PCA

Tháng 6 năm 2024

Mục lục

1	Thông tin bài làm	2
1.1	Đề tài	2
1.2	Danh sách thành viên nhóm	2
2	Tóm tắt	3
3	Giới thiệu	3
3.1	Lịch sử ra đời	3
3.2	Mục tiêu báo cáo	3
4	Phương pháp thực hiện	4
4.1	Quy trình tiếp cận	4
4.2	Hình ảnh đầu vào	4
4.3	Làm nhiễu ảnh	4
4.4	Xử lý PCA	5
4.5	Ảnh sau khi khử nhiễu	6
5	Kết quả thực nghiệm	7
5.1	Dữ liệu đầu vào	7
5.2	Ảnh nhiễu	7
5.3	Kết quả thu được	8
5.3.1	Khử nhiễu bằng PCA khi dùng thư viện hỗ trợ	8
5.3.2	Khử nhiễu bằng PCA khi dùng không dùng thư viện hỗ trợ	8
6	Đánh giá	8
7	Kết luận	9
7.1	Tổng kết	9
7.2	Các ứng dụng và tiềm năng của khử nhiễu bằng PCA	9
8	Tài liệu tham khảo	9

1 Thông tin bài làm

1.1 Đề tài

Đề tài thực hiện: Xử lý khử nhiễu hình ảnh bằng PCA.

Môn học: Phương pháp số cho Khoa học dữ liệu

Mã môn học: MTH10607

1.2 Danh sách thành viên nhóm

STT	Họ và tên	Mã số sinh viên	Vai trò
1	Nguyễn Ngọc Thành	21280108	Nhóm trưởng
2	Lâm Gia Phú	21280104	Thành viên
3	Trần Ngọc Khánh Như	21280040	Thành viên
4	Phạm Hoàng Đăng Khoa	21280021	Thành viên
5	Nguyễn Thái Duy	21110277	Thành viên

2 Tóm tắt

Trong báo cáo này, chúng em sử dụng Phân tích Thành phần Chính (Principal Component Analysis - PCA) để khử nhiễu ảnh. PCA là một kỹ thuật giảm chiều dữ liệu mạnh mẽ, cho phép trích xuất các thành phần chính của dữ liệu và loại bỏ các thành phần chứa nhiễu. Phương pháp của chúng tôi bao gồm các bước sau: đầu tiên, áp dụng PCA để biến đổi tập hợp ảnh nhiễu sang không gian thành phần chính; sau đó, giữ lại các thành phần chính quan trọng nhất và loại bỏ các thành phần không quan trọng; cuối cùng, tái tạo lại ảnh từ các thành phần đã chọn. Kết quả thực nghiệm cho thấy phương pháp PCA không chỉ giảm nhiễu hiệu quả mà còn duy trì các đặc trưng quan trọng của ảnh gốc. Các thí nghiệm trên nhiều loại ảnh khác nhau cho thấy phương pháp đề xuất có khả năng ứng dụng rộng rãi và hiệu quả trong việc khử nhiễu ảnh.

3 Giới thiệu

3.1 Lịch sử ra đời

PCA được phát triển bởi Karl Pearson vào năm 1901. Pearson là một nhà toán học và thống kê học người Anh, được coi là một trong những người đặt nền móng cho thống kê hiện đại. Sau đó, Harold Hotelling, một nhà kinh tế và thống kê người Mỹ, đã phát triển thêm các lý thuyết về PCA vào những năm 1930. Hotelling đã mở rộng và chính thức hóa các khái niệm về PCA, giúp nó trở thành một công cụ quan trọng trong phân tích dữ liệu.

PCA (Principal Component Analysis - Phân Tích Thành Phần Chính) là một kỹ thuật thống kê dùng để giảm chiều dữ liệu bằng cách biến đổi dữ liệu ban đầu sang một không gian mới mà ở đó các thành phần chính (principal components) là các hướng biến thiên lớn nhất của dữ liệu. PCA tìm ra các hướng của biến đổi dữ liệu lớn nhất và chiếu dữ liệu lên các hướng này để giữ lại thông tin quan trọng nhất.

PCA được sử dụng rộng rãi trong nhiều lĩnh vực:

- Xử lý ảnh: Giảm nhiễu, nhận diện khuôn mặt, nén ảnh.
- Học máy: Tiền xử lý dữ liệu, giảm chiều dữ liệu để tăng hiệu quả của các mô hình học máy.
- Tài chính: Phân tích dữ liệu tài chính, dự đoán thị trường.
- Sinh học: Phân tích biểu hiện gen, phân loại sinh học.

3.2 Mục tiêu báo cáo

Trong lĩnh vực xử lý ảnh, nhiễu là một vấn đề phổ biến và có thể ảnh hưởng tiêu cực đến chất lượng của các kết quả phân tích và nhận dạng. Khử nhiễu ảnh, do đó, là một bước quan trọng nhằm cải thiện chất lượng ảnh và độ tin cậy của các hệ thống xử lý ảnh. Một trong những phương pháp hiệu quả để khử nhiễu là sử dụng Phân tích Thành phần Chính (Principal Component Analysis - PCA).

Có hai cách chính để thực hiện PCA:

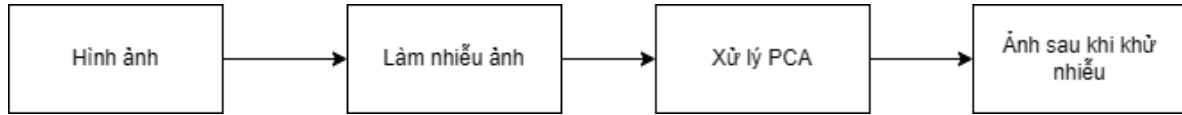
1. **Sử dụng các thư viện có sẵn:** Các thư viện như scikit-learn trong Python cung cấp các công cụ mạnh mẽ để thực hiện PCA một cách dễ dàng và nhanh chóng. Bằng cách sử dụng các hàm có sẵn, chúng em có thể áp dụng PCA để trích xuất các thành phần chính của ảnh một cách trực tiếp và tái tạo lại ảnh với nhiễu đã được giảm thiểu.
2. **Sử dụng vector riêng của ma trận hiệp phương sai:** Phương pháp này yêu cầu tính toán ma trận hiệp phương sai của dữ liệu, sau đó xác định các vector riêng và giá trị riêng của ma trận này. Các vector riêng tương ứng với các thành phần chính của dữ liệu, và bằng cách giữ lại các vector riêng với giá trị riêng lớn nhất, chúng em có thể tái tạo lại ảnh với mức nhiễu được giảm thiểu.

Phương pháp sử dụng vector riêng của ma trận hiệp phương sai mặc dù phức tạp hơn nhưng mang lại sự kiểm soát tốt hơn trong việc hiểu và điều chỉnh quá trình PCA. Trong báo cáo này, chúng em sẽ trình bày chi tiết cả hai cách tiếp cận và so sánh hiệu quả của chúng trong việc khử nhiễu ảnh.

4 Phương pháp thực hiện

4.1 Quy trình tiếp cận

Đây là quy trình tiếp cận trong quá trình thực hiện khử nhiễu ảnh:



4.2 Hình ảnh đầu vào

Hình ảnh đầu vào có thể lựa chọn tự do với nhiều kích thước khác nhau.

4.3 Làm nhiễu ảnh

Trong xử lý ảnh số, làm nhiễu bằng phân phối Gauss là một kỹ thuật thường được áp dụng để mô phỏng và nghiên cứu hiệu ứng của nhiễu trong các ứng dụng thực tế. Phân phối Gauss, hay còn gọi là phân phối chuẩn, được biểu diễn bởi hàm mật độ xác suất:

$$f(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Trong đó:

- x là giá trị ngẫu nhiên.
- μ là giá trị kỳ vọng (mean) của phân phối (thường bằng 0 trong trường hợp làm nhiễu).
- σ là độ lệch chuẩn (standard deviation), xác định mức độ biến động của dữ liệu.

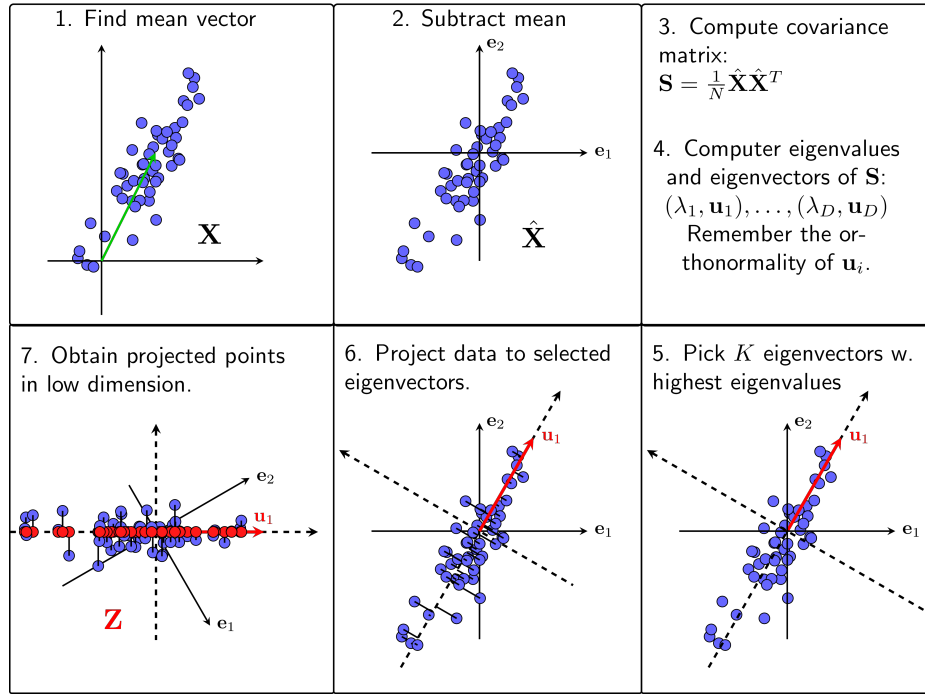
Quá trình làm nhiễu bằng phân phối Gauss bao gồm các bước sau:

1. Tạo ra nhiễu Gaussian : Đầu tiên, tạo ra các giá trị nhiễu ngẫu nhiên từ phân phối Gauss với mean $\mu = 0$ và độ lệch chuẩn σ .
2. Áp dụng nhiễu vào ảnh : Các giá trị nhiễu được áp dụng lên các pixel của ảnh gốc.
3. Tác động của nhiễu : Nhiễu Gauss có thể gây ra các hiện tượng như điểm sáng hoặc tối không mong muốn, làm mất đi chi tiết và làm giảm độ phân giải của hình ảnh.

Kỹ thuật này không chỉ được sử dụng để nghiên cứu hiệu ứng của nhiễu mà còn để kiểm tra tính chất của các thuật toán xử lý ảnh trong điều kiện nhiễu.

4.4 Xử lý PCA

PCA procedure



1. Chuẩn hóa dữ liệu: Đảm bảo rằng dữ liệu có trung bình bằng 0 và phương sai bằng 1. Trước khi áp dụng PCA, dữ liệu cần được chuẩn hóa để đảm bảo rằng tất cả các biến có cùng đơn vị đo lường và đóng góp vào phân tích một cách công bằng. Chuẩn hóa thường bao gồm việc trừ đi trung bình và chia cho độ lệch chuẩn của từng biến.

$$X_{\text{norm}} = \frac{X - \mu}{\sigma}$$

Trong đó X là dữ liệu ban đầu, μ là trung bình và σ là độ lệch chuẩn.

2. Tính ma trận hiệp phương sai: Ma trận hiệp phương sai Σ là một công cụ quan trọng trong phân tích thống kê và xử lý dữ liệu, giúp đo lường mối quan hệ giữa các biến.

Đối với dữ liệu đã chuẩn hóa, ma trận hiệp phương sai được tính bằng công thức sau:

$$\Sigma = \frac{1}{n-1} X_{\text{norm}}^T X_{\text{norm}}$$

Trong đó:

- Σ là ma trận hiệp phương sai.
- X_{norm} là ma trận dữ liệu đã chuẩn hóa (trung bình = 0, độ lệch chuẩn = 1).
- n là số lượng mẫu trong dữ liệu.

Ý nghĩa của ma trận hiệp phương sai:

- Đánh giá mối quan hệ tuyến tính : Ma trận hiệp phương sai cho biết độ biến động và mối quan hệ tuyến tính giữa các biến. Giá trị lớn hơn trên đường chéo chính của ma trận chỉ ra mức độ biến động lớn của từng biến.
- Phân tích thành phần chính (PCA) : Ma trận hiệp phương sai cũng là cơ sở cho phân tích thành phần chính (PCA), một phương pháp quan trọng để giảm chiều dữ liệu và phát hiện các biến quan trọng.

- Do lường sự phụ thuộc giữa các biến : Từ ma trận hiệp phương sai, có thể tính toán các chỉ số như hệ số tương quan giữa các biến, giúp hiểu rõ hơn về mối quan hệ giữa chúng và phân tích sự phụ thuộc.

Do đó, tính ma trận hiệp phương sai không chỉ đơn thuần là tính toán số liệu mà còn là một công cụ quan trọng để giúp hiểu và phân tích dữ liệu một cách toàn diện.

3. Tính toán các vector riêng và giá trị riêng: Vector riêng v và giá trị riêng λ của ma trận hiệp phương sai giúp xác định các phương chiều chính của dữ liệu.

$$\Sigma v = \lambda v$$

Trong đó λ là giá trị riêng và v là vector riêng.

4. Đưa các vector riêng thành một hệ trực chuẩn: Chia các vector riêng cho chính độ dài của nó để đưa về một hệ trực chuẩn các vector riêng.

$$u_i = \frac{v_i}{\|v_i\|_2}$$

5. Sắp xếp các vector riêng: Sắp xếp các vector riêng theo thứ tự giảm dần của các giá trị riêng. Các vector riêng với các giá trị riêng lớn nhất sẽ được giữ lại vì chúng biểu diễn nhiều thông tin nhất. Chọn k vector riêng hàng đầu tương ứng với k giá trị riêng lớn nhất để tạo thành ma trận các vector riêng W .

6. Chọn các vector riêng hàng đầu và biến đổi dữ liệu ban đầu: Sử dụng các vector riêng đã chọn để chiếu dữ liệu ban đầu lên không gian mới. Không gian mới này có số chiều ít hơn nhưng vẫn giữ lại được phần lớn thông tin quan trọng của dữ liệu.

$$X_{\text{new}} = X_{\text{norm}} W$$

Trong đó W là ma trận các vector riêng đã chọn.

4.5 Ảnh sau khi khử nhiễu

Ảnh sau khi khử nhiễu sẽ được khôi phục đến một khoảng giá trị và được đánh giá bằng PSNR (Peak Signal-to-Noise Ratio) và SSIM (Structural Similarity Index).

PSNR (Peak Signal-to-Noise Ratio) là một chỉ số được sử dụng để đo lường chất lượng của một hình ảnh sau khi nó đã bị nhiễu hoặc đã trải qua quá trình nén. Công thức tính PSNR được biểu diễn như sau:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right)$$

Trong đó:

- MAX là giá trị tối đa của pixel (trong ảnh đen trắng, MAX thường là 255 nếu sử dụng 8-bit per channel).
- MSE (Mean Squared Error) là trung bình cộng bình phương của sai số giữa hình ảnh gốc và hình ảnh đã xử lý.

PSNR thường được đo theo đơn vị decibel (dB) và giá trị càng cao thể hiện rằng chất lượng hình ảnh càng cao và sai số càng thấp. Đây là một công cụ hữu ích để so sánh và đánh giá các thuật toán xử lý ảnh, cũng như để đo lường mức độ giảm chất lượng khi hình ảnh bị nén hoặc truyền tải.

SSIM (Structural Similarity Index) là một chỉ số đo lường chất lượng hình ảnh, được thiết kế để cải thiện các phương pháp truyền thống như PSNR (Peak Signal-to-Noise Ratio) và MSE (Mean Squared Error). SSIM được phát triển bởi Wang et al. vào năm 2004 và đã trở thành một trong những phương pháp phổ biến nhất để đánh giá sự tương đồng giữa hai hình ảnh.

SSIM nhằm đo lường sự tương đồng giữa hai hình ảnh dựa trên việc so sánh các mẫu nhỏ, được gọi là cửa sổ (window), trên hình ảnh. Mục tiêu chính là bắt chước cách mà con người nhận thức về sự khác biệt giữa các hình ảnh, đặc biệt là tập trung vào sự thay đổi về cấu trúc, ánh sáng và tương phản.

SSIM bao gồm ba thành phần chính:

- **Luminance (Độ sáng):** So sánh độ sáng trung bình của hai hình ảnh.
- **Contrast (Độ tương phản):** So sánh độ tương phản giữa hai hình ảnh.
- **Structure (Cấu trúc):** So sánh cấu trúc của các mẫu nhỏ giữa hai hình ảnh.

Các thành phần này được kết hợp để tạo thành chỉ số SSIM duy nhất, với công thức như sau:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Trong đó:

- μ_x và μ_y là giá trị trung bình của cửa sổ trên hình ảnh x và y .
- σ_x^2 và σ_y^2 là phương sai của cửa sổ trên hình ảnh x và y .
- σ_{xy} là hiệp phương sai giữa x và y .
- C_1 và C_2 là các hằng số để tránh mẫu số bằng 0.

5 Kết quả thực nghiệm

5.1 Dữ liệu đầu vào

Đây là hình ảnh đầu vào:



Với kích thước là: 1800, 1009.

Hình ảnh sẽ được đưa vào làm nhiễu bức ảnh này đi với mức độ nhiễu là 50.

5.2 Ảnh nhiễu

Đây là ảnh sau khi làm nhiễu bằng phương pháp Gauss:

Noisy image



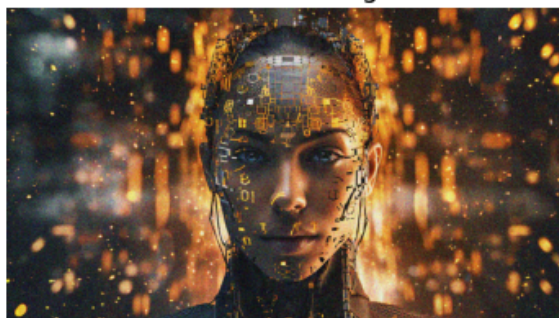
Sau đó ảnh nhiễu sẽ được đưa vào xử lý khử nhiễu với mức độ giữ lại thông tin được cài đặt sẵn là 0.98 để dễ thấy sự khác biệt giữa hai phương pháp và thu được kết quả bên dưới như sau.

5.3 Kết quả thu được

5.3.1 Khử nhiễu bằng PCA khi dùng thư viện hỗ trợ

Đây là kết quả thu được sau khi khử nhiễu bằng thư viện hỗ trợ:

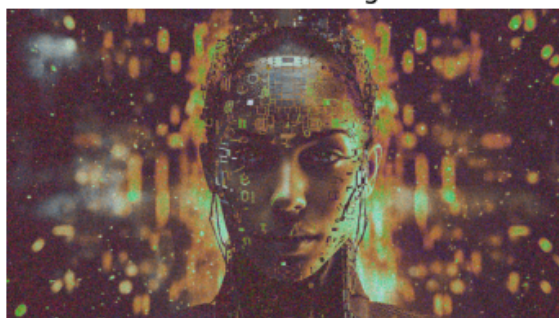
Denoised Image



5.3.2 Khử nhiễu bằng PCA khi dùng không dùng thư viện hỗ trợ

Đây là kết quả thu được sau khi khử nhiễu:

Denoised image



6 Đánh giá

Sau khi thu được kết quả từ hai phương pháp trên và sử dụng chỉ số PSNR và SSIM để đo lường thì kết quả thu về như sau:

Phương pháp	PSNR	SSIM
PCA từ thư viện	18.0051	0.1918
PCA không dùng thư viện	14.6740	0.1385

1. **Chất lượng hình ảnh:** Phương pháp PCA từ thư viện cho chất lượng hình ảnh tốt hơn (cao hơn về cả PSNR và SSIM) so với phương pháp PCA không dùng thư viện.
2. **Giữ lại chi tiết cấu trúc:** SSIM cao hơn từ PCA từ thư viện chỉ ra rằng phương pháp này giữ lại nhiều chi tiết cấu trúc của hình ảnh hơn so với phương pháp không dùng thư viện.
3. **Hiệu suất:** Phương pháp PCA từ thư viện hiệu quả hơn trong việc duy trì chất lượng và chi tiết của hình ảnh so với phương pháp tự viết không dùng thư viện.

7 Kết luận

7.1 Tổng kết

Trong báo cáo này, chúng ta đã tìm hiểu về phương pháp khử nhiễu hình ảnh bằng PCA, bao gồm các khái niệm cơ bản về khử nhiễu, PCA và các bước thực hiện PCA để khử nhiễu. Phương pháp PCA đã chứng minh hiệu quả cao trong việc khử nhiễu và cải thiện chất lượng hình ảnh, đồng thời giúp các bước xử lý ảnh tiếp theo trở nên chính xác hơn.

7.2 Các ứng dụng và tiềm năng của khử nhiễu bằng PCA

PCA không chỉ hữu ích trong việc khử nhiễu hình ảnh mà còn có nhiều ứng dụng tiềm năng trong các lĩnh vực khác như:

- **Nhận diện khuôn mặt:** PCA giúp cải thiện độ chính xác của các hệ thống nhận diện khuôn mặt bằng cách giảm nhiễu và giữ lại các đặc trưng quan trọng.
- **Y tế:** Trong y tế, PCA được sử dụng để khử nhiễu các hình ảnh y khoa như X-quang, MRI, giúp bác sĩ chẩn đoán chính xác hơn.
- **Phân tích hình ảnh vệ tinh:** PCA giúp cải thiện chất lượng các hình ảnh vệ tinh, làm rõ các chi tiết quan trọng phục vụ cho việc nghiên cứu và phân tích địa lý.
- **Nén ảnh:** PCA có thể được sử dụng để nén ảnh mà vẫn giữ được chất lượng cao, giúp tiết kiệm dung lượng lưu trữ và băng thông truyền tải.

PCA là một công cụ mạnh mẽ và linh hoạt trong xử lý ảnh và nhiều lĩnh vực khác. Việc nghiên cứu và áp dụng PCA có thể mở ra nhiều cơ hội và tiềm năng mới trong khoa học và công nghệ.

8 Tài liệu tham khảo

Danh sách tài liệu tham khảo

- [1] Machine Learning cơ bản. *Phân tích thành phần chính (PCA)*. 2017. URL: <https://machinelearningcoban.com/2017/06/15/pca/>.
- [2] Channel Name. *Understanding PCA (Principal Component Analysis)*. Publishing Year. URL: https://www.youtube.com/watch?v=HMOI_lkzW08.
- [3] Minh Nguyễn. *Ma trận hiệp phương sai (Covariance Matrix)*. Publishing Year. URL: <https://minhng.info/toan-hoc/ma-tran-hiep-phuong-sai-covariance-matrix.html>.
- [4] Author Name. In: *International Journal of Innovative Science and Engineering Technology (IJSET)* 3.1 (Publishing Year), Page Numbers. URL: https://ijiset.com/vol3/v3s1/IJSET_V3_I1_22.pdf.
- [5] Joseph Salmon. Website. URL: https://josephsalmon.eu/code/index_codes.php?page=PatchPCA_denoising.