TẠP CHÍ KHOA HỌC ĐẠI HỌC ĐÔNG Á
Số 01, 75-80, 2022

ỨNG DỤNG KỸ THUẬT BỘ MÃ TỰ ĐỘNG TÍCH CHẬP (CONVOLUTIONAL AUTOENCODER) TRONG KHỬ NHIỄU HÌNH ẢNH

Trần Tùng Nhi[1,*], Phạm Văn Quân[1], Phan Công Đạt[1]
[1]*Viện Nghiên cứu Quốc tế về Trí tuệ Nhân tạo và Khoa học Dữ liệu, Trường Đại học Đông Á*
33 Xô Viết Nghệ Tĩnh, Hòa Cường Nam, Hải Châu, Đà Nẵng
** Tác giả chịu trách nhiệm chính: nhitt@donga.edu.vn*
Ngày nhận bài: 20.04.2021, Ngày chấp nhận: 20.09.2021, Ngày đăng: 30.03.2022

TÓM TẮT:
Khử nhiễu hình ảnh luôn là một vấn đề được quan tâm hàng đầu trong xử lý ảnh. Trong bài viết này, chúng tôi đề xuất sử dụng kỹ thuật bộ mã tự động kết hợp mạng nơ-ron tích chập để giải quyết bài toán xử lý ảnh nhiễu đồng thời cho thấy hiệu quả của phương pháp này đối với các loại nhiễu thường gặp trong việc số hóa hình ảnh.

Từ khóa: *Convolutional Autoencoder (CAE), khử nhiễu hình ảnh*

DENOISING IMAGES USING CONVOLUTIONAL AUTOENCODER TECHNIQUES

Tran Tung Nhi[1,*], Pham Van Quan[1], Phan Cong Dat[1]
[1]*International Research Institute for Artificial Intelligence and Data Science (IAD), Dong A University, 33 Xo Viet Nghe Tinh, Hoa Cuong Nam, Hai Chau, Danang 55000, Vietnam*
**Corresponding author: nhitt@donga.edu.vn*
Received: April 20, 2021, Accepted: September 20, 2021, Published: March 30, 2022

ABSTRACT:
Denoising images is one of the principal problems in image processing in particular and data cleaning in general - the first step of machine learning. This paper proposed a model using autoencoder techniques combining the convolutional neural network to denoise the images. We also show the effectiveness of this model in the different types of popular noise in digital image processing: Gaussian noise, Salt and pepper noise, Poisson noise.

Keywords: *Convolutional Autoencoder (CAE), Denoise Image*

I. ĐẶT VẤN ĐỀ

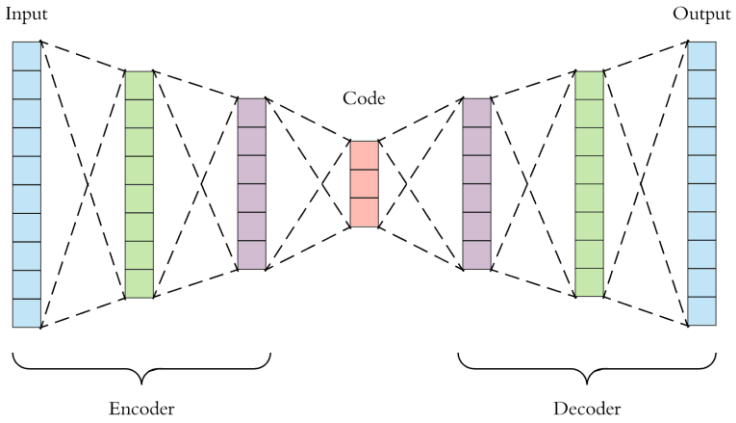
Ngày nay với sự phát triển mạnh mẽ của công nghệ số hóa, thông tin số đặc biệt là hình ảnh số đã trở nên vô cùng phổ biến. Nhiễu ảnh luôn tồn tại trong các bức ảnh kỹ thuật số do quá trình mã hóa, truyền phát, thu nhận hình ảnh. Do đó khử nhiễu cho hình ảnh từ lâu đã trở thành vấn đề luôn được quan tâm đặc biệt trong lĩnh vực Thị giác máy tính. Có rất nhiều phương pháp để khử nhiễu ảnh đã được đề xuất như biến đổi wavelets (Coifman và cộng sự, 1995), partial differential equations (PDEs) (Perona và cộng sự, 1990). Hiện nay, với sự phát triển mạnh mẽ của công nghệ Học máy (Machine Learning), việc khử nhiễu hình ảnh có thể được thực hiện bằng các thuật toán Machine Learning và thu được những kết quả khả quan. Trong đó có thể kể đến thuật toán Bộ mã tự động (Autoencoder), một mạng nơ-ron được phát triển và ứng dụng trong những năm gần đây. Được đánh giá là một phương pháp hiệu quả và không bị giới hạn bởi loại nhiễu cần xử lý,

Autoencoder được sử dụng rộng rãi trong lĩnh vực này (Gondara, 2016; Vincent và cộng sự, 2008, Xie và cộng sự, 2012). Trong khuôn khổ bài báo này, chúng tôi sử dụng một phiên bản của Autoencoder là Bộ mã tự động tích chập để tiến hành loại nhiễu trên tập hình ảnh MNIST. Sau khi được xây dựng, bộ mã tự động này sẽ chạy thử trên các loại nhiễu thường gặp trong xử lý ảnh.

II. BỘ MÃ TỰ ĐỘNG (AUTOENCODER) & MẠNG NƠ-RON TÍCH CHẬP (CONVOLUTIONAL NEURAL NETWORK)

2.1. Bộ mã tự động (Autoencoder)

Autoencoder là mạng nơ-ron tự động mã hóa, một thuật toán máy học không giám sát áp dụng sự lan truyền ngược, đặt các giá trị đích bằng với các đầu vào. Mã tự động được sử dụng để giảm kích thước đầu vào thành một biểu diễn nhỏ hơn. Kiến trúc của Autoencoder tạo ra một nút thắt cổ chai vì vậy những đặc trưng đại diện được giữ lại.



Hình 1. Kiến trúc Autoencoder

Một Autoencoder gồm 3 phần chính (**Hình 1**):

- Encoder: mã hóa dữ liệu đầu vào thành một phần dữ liệu với số chiều nhỏ hơn dữ liệu ban đầu. Cũng có thể nói, đây là bước nén dữ liệu đầu vào.
- Code: được xem như phần thắt nút cổ chai của kiến trúc, đại diện cho phần dữ liệu đã nén để chuẩn bị đưa vào Decoder.
- Decoder: có nhiệm vụ tái tạo phần dữ liệu đã nén sao cho giống dữ liệu đầu vào nhất có thể.

Mục tiêu của autoencoder là tái tạo sao cho đầu ra (output) giống đầu vào (input) nhất có thể.
Dữ liệu đầu vào được ký hiệu như sau:

$$x_1, x_2, \ldots, x_n$$

Dữ liệu đầu ra được ký hiệu:

$$z_1, z_2, \ldots, z_n$$

Khi đưa vào dữ liệu ở dạng vector n chiều $x_1, x_2, \ldots, x_n$, phần encoder sẽ mã hóa (encode) thành 1 vector y m chiều (thường thì m < n) :

$$y = s(Wx + b),$$ với s là 1 hàm phi tuyến.

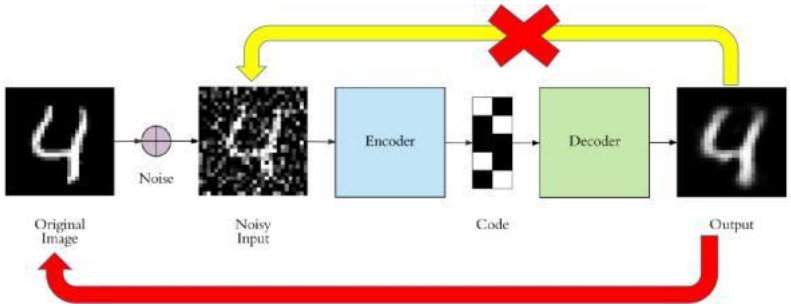
Nhiệm vụ của decoder là nhận dữ liệu m chiều này và giải mã (tái tạo) thành dữ liệu đầu ra $z_1, z_2, \ldots, z_n$.

$$z = s(W'y+b')$$

Khi đó hàm mất mát (loss function) sẽ được tính toán dựa trên sự khác nhau giữa z và x.

2.1.1. Bộ mã tự động khử nhiễu

Như đã nói ở trên, mục tiêu của autoencoder là tái tạo lại dữ liệu hình ảnh ở đầu vào. Tuy nhiên ở bài toán khử nhiễu, đầu vào sẽ là hình ảnh có nhiễu và mục tiêu của mô hình được mở rộng ra là khôi phục lại hình ảnh ban đầu (trước khi thêm nhiễu). Khi đó, tiêu chí để đánh giá mô hình không còn là sự giống nhau giữa hình ảnh đầu vào và đầu ra nữa, mà sẽ là sự giống nhau giữa hình ảnh đầu ra và hình ảnh trước nhiễu (**Hình 2**).



Hình 2. Denoising Autoencoder

Cụ thể, hình ảnh đầu vào lúc này sẽ có dạng:

$$\hat{x} = x + n$$

Với $\hat{x}$ là hình ảnh nhiễu đầu vào, được tạo ra từ phép "cộng" của hình ảnh gốc x và nhiễu n. Ở các phần sau, khi thử nghiệm tính hiệu quả của mô hình, n sẽ là các loại nhiễu thường gặp trong xử lý ảnh như nhiễu Gaussian, nhiễu muối tiêu (Salt & Pepper), nhiễu Speckle, …

Khi đó, phần dữ liệu sau mã hóa (encode) sẽ là:

$$y = s(W\hat{x} + b)$$

Đầu ra của bộ mã tự động khử nhiễu này là :

$$z = s(W'y+b')$$

Như đã nói, loss function của bộ mã tự động này sẽ dựa trên sự khác nhau của z và x chứ không phải giữa z và x̂.

2.1.2 Bộ mã tự động tích chập

Bộ mã tự động tích chập CAE (Convolutional Autoencoder) được xây dựng trên cơ sở của kiến trúc autoencoder với mạng nơ-ron tích chập (convolutional neural network) tại bộ encoder và decoder. Bộ mã tự động tích chập được đánh giá là ưu việt hơn so với bộ mã tự động truyền thống nhờ tích hợp được những thế mạnh của mạng nơ-ron tích chập.

2.2. Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) (LeCun, 1989) là một loại mạng thần kinh chuyên biệt để xử lý dữ liệu có cấu trúc liên kết giống như lưới, ví dụ như dữ liệu trên miền thời gian, được xem như là lưới 1D của các mẫu được lấy trong từng khoảng thời gian nhất định, hay dữ liệu hình ảnh, có thể xem là lưới 2D các giá trị pixel. Mạng tích chập đã rất thành công trong các ứng dụng thực tế. Cái tên "mạng tích chập" ý nói đến phép toán được sử dụng trong mạng nơ-ron là phép toán tích chập. Mạng nơ-ron tích chập có thể hiểu đơn giản là mạng nơ-ron sử dụng tích chập thay cho phép nhân ma trận tổng quát trong ít nhất một lớp của mạng.

Về cơ bản, có 3 loại lớp (layer) cấu thành nên một mạng nơ-ron tích chập :

- Convolutional Layers
- Pooling Layers
- Dense Layers hay còn gọi là Fully Connected Layers

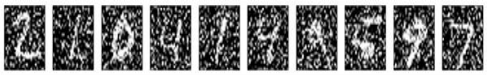
III. KẾT QUẢ VÀ THẢO LUẬN

3.1 Xây dựng, huấn luyện mô hình

3.1.1 Dữ liệu

Để huấn luyện mô hình CAE khử nhiễu, ở đây chúng tôi sử dụng bộ dữ liệu mã nguồn mở MNIST, gồm 60.000 bức ảnh trắng đen của các chữ số viết tay, có kích thước 28x28, thường được dùng trong huấn luyện các hệ thống xử lý ảnh khác nhau. Mục tiêu của việc huấn luyện là tạo ra 1 mô hình CAE có khả năng khử nhiễu hình ảnh, do đó các hình ảnh trong bộ MNIST được thêm nhiễu và đưa vào mô hình để tiến hành huấn luyện (**Hình 3,4**).

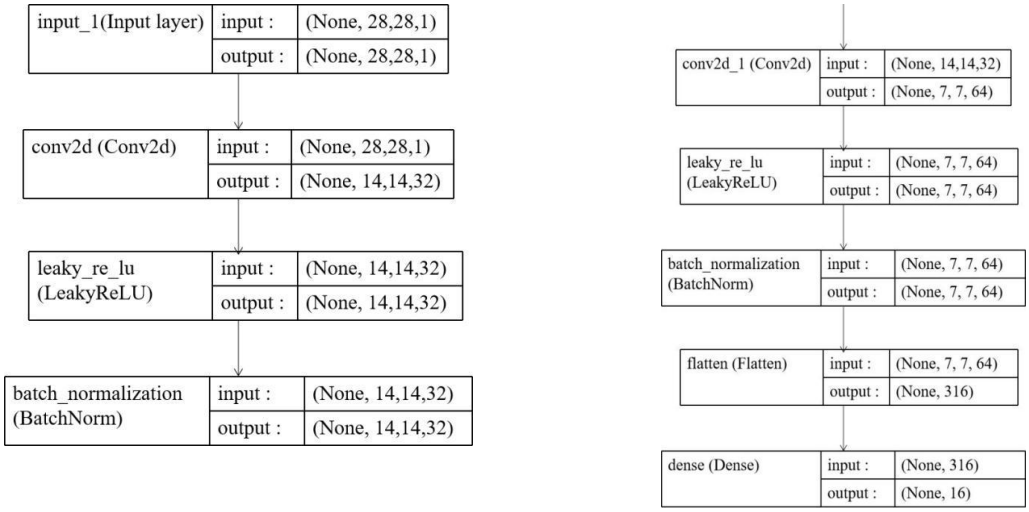
Hình 3. Bộ dữ liệu chữ số viết tay MNIST



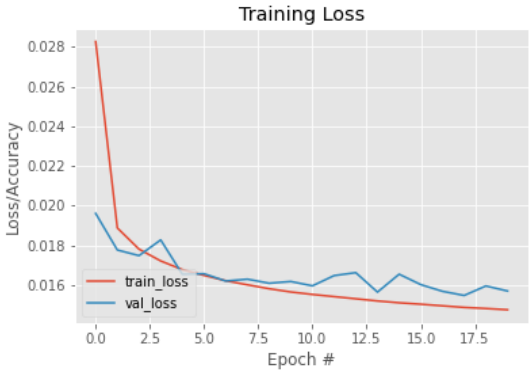
Hình 4. Dữ liệu đầu vào đã thêm nhiễu

3.1.2 Huấn luyện mô hình

Kiến trúc của mô hình bao gồm 2 phần chính: encoder và decoder, trong đó mỗi phần đều có các lớp cơ bản của một mạng nơ-ron tích chập. Các thông số cụ thể về từng lớp trong mạng nơ-ron được thể hiện ở **Hình 5.**



Hình 5. Kiến trúc của mô hình CAE



Hình 6. Kết quả huấn luyện

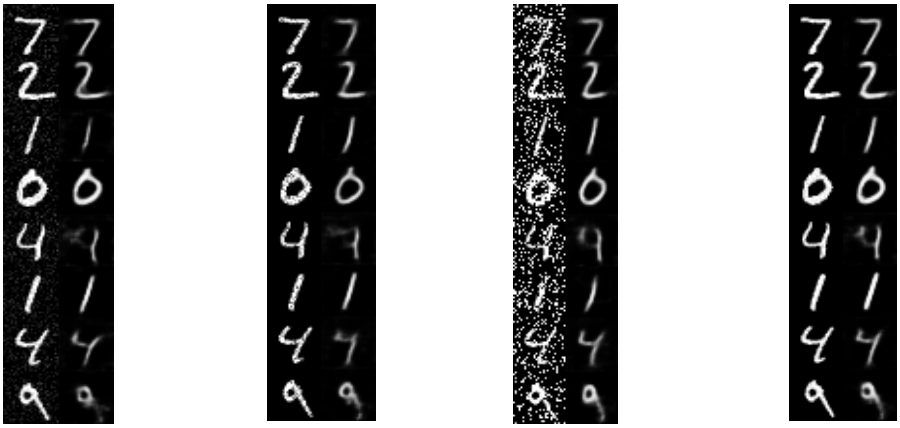
Hình 7. Hình ảnh trước và sau khử nhiễu

Hình 6 biểu diễn kết quả huấn luyện qua từng epoch. Việc huấn luyện mô hình này được thực hiện bằng laptop ASUS Vivobook (Ryzen7 3700U, RAM 8GB, không có GPU) trên nền tảng thư viện Keras, một thư viện mã nguồn mở chuyên dùng trong lĩnh vực Học máy. Loss function được sử dụng ở đây là hàm Mean-squared error (MSE). Quá trình huấn luyện với 30 epochs và kích thước của 1 batch là 1875, diễn ra trong 30 - 35 phút.

Tiến hành thử mô hình đã huấn luyện trên bức ảnh bị nhiễu, ta thu được kết quả ở đầu ra như **Hình 7**.

3.2. Thực hiện khử nhiễu trên các loại nhiễu thường gặp

Như đã đề cập ở trên, trong khuôn khổ bài báo này, chúng tôi không chỉ trình bày về một kỹ thuật khử nhiễu hình ảnh mới, convolutional autoencoder, mà còn thử nghiệm mô hình đã huấn luyện trên các loại nhiễu thường gặp trong xử lý ảnh như nhiễu Gaussian, nhiễu muối tiêu,… Để thêm các loại nhiễu này vào ảnh, chúng tôi sử dụng hàm random_noise trong thư viện scikit-image của Python. Scikit-image (trước đây là scikits.image) là một thư viện xử lý ảnh mã nguồn mở cho ngôn ngữ lập trình Python. Thư viện này bao gồm các thuật toán cho phân đoạn, biến đổi hình học, thao tác không gian màu, phân tích, lọc, hình thái học, phát hiện tính năng và hơn thế nữa. Giống với các thư viện khác của Python như Scikit-learn, Scikit-image cũng được thiết kế để tương thích với NumPy và SciPy. **Hình 8** là kết quả khử nhiễu của mô hình được xây dựng. Để dễ so sánh, các kết quả khử nhiễu sẽ được đặt cạnh hình ảnh ban đầu (đã được thêm nhiễu).



a) Khử nhiễu Gausian | b)Khử nhiễu speckle với $\mu = 0$ & $\sigma^2 = 0.1$ | c) Khử nhiễu salt & pepper với tỷ lệ 0.3 | d) Khử nhiễu Poisson

Hình 8. Kết quả khử nhiễu của mô hình được xây dựng

Bảng 1 So sánh kết quả lọc nhiễu bằng CAE trên các loại nhiễu khác nhau với thông số MSE (Mean Squared Error) giữa ảnh ban đầu và ảnh sau khi lọc nhiễu.

Bảng 1. So sánh kết quả khử nhiễu

Loại nhiễu	**MSE**
Gaussian	0.033
Speckle ($\mu = 0$, $\sigma^2 = 0.1$)	0.035
Salt & pepper (n = 0.3)	0.041
Poisson	0.045

IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận

Với việc ứng dụng kỹ thuật autoencoder kết hợp với mạng nơ-ron tích chập, chúng tôi đã tạo ra một mô hình khử nhiễu có thể áp dụng trên các loại nhiều khác nhau. Từ những kết quả thu được, có thể thấy mô hình hoạt động tốt nhất đối với loại nhiễu speckle, nhiễu Gaussian. Riêng đối với nhiễu salt & pepper, nhiễu Poisson kết quả vẫn chưa được tốt. Tuy nhiên, so với cách làm truyền thống sử dụng các bộ lọc như mean filter, Gaussian filter… thường chỉ có tác dụng với một loại nhiễu nhất định, thì đây là một phương án khả quan và có tiềm năng phát triển.

4.2. Hướng phát triển

Trong tương lai, để nâng cao hiệu suất khử nhiễu, có thể nghiên cứu kết hợp thêm các phương pháp để tiền xử lý hình ảnh, cải thiện chất lượng đầu vào trước khi tiến hành lọc nhiễu. Ngoài ra, việc ứng dụng autoencoder không chỉ dừng lại ở lọc nhiễu ảnh mà còn có thể ứng dụng trong việc phát hiện bất thường (anomaly detection), nén ảnh… Thêm vào đó, việc kết hợp các ưu điểm của autoencoder với các mạng neuro thường gặp như Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM) hứa hẹn là giải pháp hữu hiệu để giải quyết các bài toán trong lĩnh vực Học máy, Học sâu.

TÀI LIỆU THAM KHẢO

Coifman, R. R., & Donoho, D. L. Translation-invariant de-noising. In Wavelets and statistics (125-150). Springer, New York, 1995.

Gondara, L. Medical image denoising using convolutional denoising autoencoders. In 2016 IEEE 16th international conference on data mining workshops (ICDMW) IEEE, 241-246, 2016.

Perona, P., & Malik, J. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence, 12(7), 629-639, 1990.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, 1096-1103, 2008.

Xie, J., Xu, L., & Chen, E. Image denoising and inpainting with deep neural networks. In Advances in neural information processing systems, 341-349, 2012.