

Bài báo nghiên cứu

PHƯƠNG PHÁP NHẬN DẠNG KÍ TỰ SỐ VIẾT TAY DỰA TRÊN MẠNG NƠ-RON HỌC SÂU

Đinh Thị Mận^{1*}, Nguyễn Văn Thịnh², Nguyễn Thế Hữu¹, Trần Thị Vân Anh¹

¹Trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh, Việt Nam

²Trường Đại học Sư phạm Thành phố Hồ Chí Minh

*Tác giả liên hệ: Đinh Thị Mận – Email: mandt@hufi.edu.vn

Ngày nhận bài: 11-10-2022; ngày nhận bài sửa: 25-4-2023; ngày duyệt đăng: 27-4-2023

TÓM TẮT

Trong bài báo này, phương pháp nhận dạng ký tự số viết tay được đề xuất theo hướng tiếp cận dựa trên mạng nơ-ron học sâu (DNN- Deep Neural Network). Đầu tiên, tập dữ liệu ảnh được trích xuất đặc trưng HOG (Histogram of Oriented Gradient) kết hợp với đặc trưng SIFT (Scale-invariant feature transform). Sau đó, một mô hình (model) mạng DNN được xây dựng để huấn luyện nhằm nhận dạng hình ảnh. Cuối cùng, ảnh đầu vào được nhận diện tự động dựa trên mô hình đã được huấn luyện. Nhằm minh chứng tính hiệu quả của phương pháp đề xuất, thực nghiệm được xây dựng và đánh giá trên tập dữ liệu ảnh MNIST. Kết quả thực nghiệm đã cho thấy tính khả thi và hiệu quả của phương pháp, đồng thời dễ dàng mở rộng cho việc nhận diện các hình ảnh chữ viết tay khác.

Từ khóa: ảnh ký tự; handwritten digit recognition; DNN; SIFT; HOG

1. Giới thiệu

Ngày nay, những nghiên cứu về trí tuệ nhân tạo phát triển mạnh mẽ và được tiếp cận rộng rãi trong mọi lĩnh vực, rất nhiều công trình nghiên cứu cùng các ứng dụng đã được công bố và áp dụng rộng rãi trong thực tiễn như: nhận dạng bảng số, hệ thống nhận diện và phát hiện khuôn mặt thời gian thực (Niranjani, 2021), phát hiện hoạt động của con người, trong lĩnh vực y tế có thể tối ưu hóa các thủ tục chẩn đoán hình ảnh y tế (Sprawls Resources, 2021), công nghệ xe tự lái (Szikora, 2021), dự đoán bệnh Alzheimer (Brownlee, 2019), phân loại khối u não (Brownlee, 2019), nhận dạng chữ số viết tay hiệu quả dựa trên HOG và SVM (Reza, 2014), nhận dạng chữ số viết tay dùng mạng nơ-ron nhân tạo (Pham, 2019)... Trong đó, nhận dạng ảnh ký tự đã thu được thành tựu cả về mặt lý thuyết lẫn ứng dụng thực tế (Rother, 2004).

Trong nhận dạng chữ, thông thường sẽ nhận dạng chữ in và nhận dạng chữ viết tay. Đối với nhận dạng chữ viết tay, từ ảnh của hàng nghìn ví dụ của mỗi chữ số được viết bởi nhiều người khác nhau. Khi đưa các bức ảnh này vào trong một thuật toán, kết quả phải chỉ

Cite this article as: Đinh Thị Mận, Nguyễn Văn Thịnh, Nguyễn Thế Hữu, & Trần Thị Vân Anh (2023). Method of handwritten digit recognition based on deep neural network. *Ho Chi Minh City University of Education Journal of Science*, 20(5), 818-830.

ra được mỗi bức ảnh tương ứng với chữ số nào. Việc nhận dạng chữ viết tay đã được nhiều nhà nghiên cứu tiếp cận với các phương pháp và kỹ thuật khác nhau được đề xuất như: các thuật toán phân loại dựa trên học sâu (Dan, 2020), mạng nơ-ron nhân tạo (Pham, 2019), máy véc-tơ hỗ trợ (Balas, 2021)... Tuy nhiên, việc nhận diện chữ viết tay đang gặp nhiều thách thức đó là: sự khác biệt đa dạng trong cách viết của từng cá nhân, chữ viết của cùng một cá nhân viết cũng có nhiều sự khác biệt trong cách viết tùy thuộc vào từng ngữ cảnh, kiểu viết, thay đổi theo thời gian... Điều này gây ra nhiều trở ngại trong việc trích xuất đặc trưng cũng như lựa chọn mô hình nhận dạng (Rother, 2004). Do đó, cần phải có một hệ thống nhận diện chính xác với thuật toán tạo ra một mô hình, tức là một hàm số mà đầu vào là một hình ảnh và đầu ra là một chữ số, khi nhận được một hình ảnh mới mà mô hình chưa nhìn thấy bao giờ, mô hình sẽ dự đoán hình ảnh đó chứa chữ số nào. Vấn đề này đang là yêu cầu cấp thiết và có tính ứng dụng trong thực tiễn.

Trong bài báo này, phương pháp nhận dạng ký tự số viết tay được đề xuất dựa trên mạng DNN nhằm tăng độ chính xác trong nhận dạng. Mạng DNN sẽ được huấn luyện dựa trên đặc trưng cấp thấp của hình ảnh từ các điểm ảnh (pixel), đơn vị nhỏ nhất của hình ảnh thích hợp cho việc xử lý, phân tích hình ảnh để mang lại kết quả nhận diện nhanh và chính xác. Đóng góp của bài báo gồm: (1) thực hiện phân đoạn ảnh bằng phương pháp Graph-cut, (2) đề xuất phương pháp trích xuất đặc trưng SIFT kết hợp đặc trưng HOG, (3) xây dựng mô hình mạng DNN để huấn luyện nhận dạng hình ảnh, (4) xây dựng ứng dụng thực nghiệm về mô hình nhận dạng hình ảnh ký tự số viết tay trên bộ dữ liệu MNIST. Phần còn lại của bài báo gồm: phần 2 phương pháp nhận dạng ảnh ký tự số; mô hình nhận dạng ảnh ký tự số và thực nghiệm được trình bày trong phần 3; kết luận và hướng phát triển được trình bày trong phần 4.

2. Đối tượng và phương pháp nghiên cứu

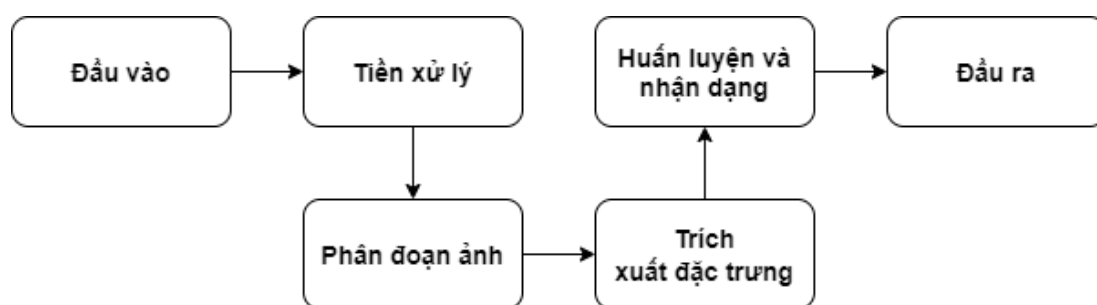
Phương pháp nhận dạng ảnh ký tự số bao gồm các bước: (1) tiền xử lý, (2) phân đoạn ảnh, (3) trích xuất đặc trưng, (4) huấn luyện và nhận dạng. Đầu vào của mô hình là đối tượng cần nhận dạng và đầu ra là ký tự tương đương của đối tượng đầu vào, đầu ra thu được từ việc được xử lý qua các bước sau:

(1) Tiền xử lý: dùng các kỹ thuật phân ngưỡng (thresholding) để loại bỏ phần nhiễu của hình ảnh, thay đổi kích thước (resize) hình ảnh đầu vào để định hướng hình ảnh một cách chính xác. Đồng thời trong giai đoạn này, loại bỏ các ký tự không mong muốn (dấu chấm, chữ viết nguệch ngoạc...) và làm mịn mẫu chữ viết tay (Balas, 2019).

(2) Phân đoạn ảnh: chia hình ảnh thành các vùng có đối tượng riêng biệt.

(3) Trích xuất đặc trưng: là bước quan trọng nhất để đạt được độ chính xác cho việc nhận dạng ký tự (Balas, 2019). Kỹ thuật trích xuất đặc trưng được sử dụng để giảm kích thước của dữ liệu mẫu đầu vào. Đầu ra được biểu diễn dưới dạng một véc-tơ đặc trưng. Các đặc trưng được trích xuất phải có các đặc điểm quan trọng của từng ký tự để phân biệt với những chữ cái khác.

(4) Huấn luyện và nhận dạng: dựa vào các đặc trưng đã được trích xuất bằng cách áp dụng các phương pháp như: các thuật toán phân loại dựa trên học sâu, mạng nơ-ron nhân tạo (Rother, 2004), máy véc-tơ hỗ trợ (Balas, 2019)... Việc nhận dạng ký tự tương ứng bằng cách so sánh với mẫu chuẩn đã được huấn luyện từ trước.



Hình 1. Mô hình nhận diện ảnh kí tự

A. Tiền xử lí ảnh

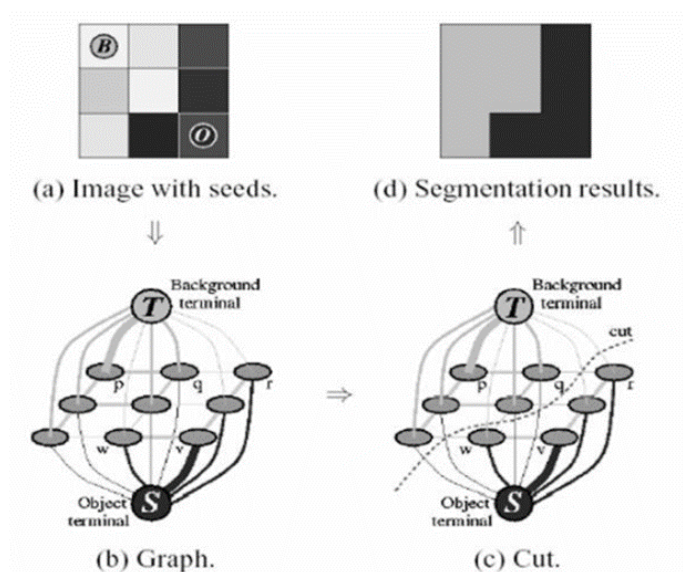
Giai đoạn này làm sạch dữ liệu và chuẩn bị trước dữ liệu, xử lí các dữ liệu không chặt chẽ, dữ liệu nhiễu. Giai đoạn này là giai đoạn quan trọng vì nếu không xử lí chính xác sẽ gây ra sai lệch trong kết quả nhận dạng.

Các thao tác tiền xử lí ảnh:

- Chuyển ảnh màu sang ảnh xám: là quá trình biến đổi ảnh màu (RGB) sang ảnh xám (grayscale);
- Nhị phân ảnh: là quá trình biến đổi một ảnh xám thành ảnh nhị phân;
- Chuẩn kích thước: chuẩn kích thước ảnh kí tự về một kích thước cố định và phóng sát bốn biên của ảnh.

B. Phân đoạn ảnh

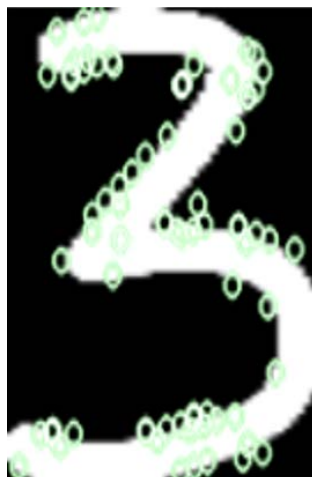
Graph – cut là một phương pháp phân đoạn hình ảnh dựa trên việc cắt giảm đồ thị. Thuật toán ước tính sự phân bố màu của đối tượng và nền bằng cách sử dụng mô hình hỗn hợp Gaussian được sử dụng để xây dựng trường ngẫu nhiên Markov (*Markov random field*) trên các *pixel*, với hàm *energy* ưu tiên các vùng được kết nối có cùng nhãn và chạy tối ưu hóa dựa trên cắt biểu đồ để suy ra giá trị của chúng (Lowe, 2004).



Hình 2. Đồ thị Graph – cut (Dong, 2021)

C. Trích xuất đặc trưng

SIFT (*Scale-invariant feature transform*): dò tìm các điểm đặc trưng theo góc, bất biến theo tỉ lệ và các phép biến đổi *Affine* (co giãn theo tỉ lệ, xoay, tịnh tiến). Đặc trưng SIFT được dùng nhiều trong bài toán so khớp ảnh, tìm ảnh tương tự (Dan, 2020).



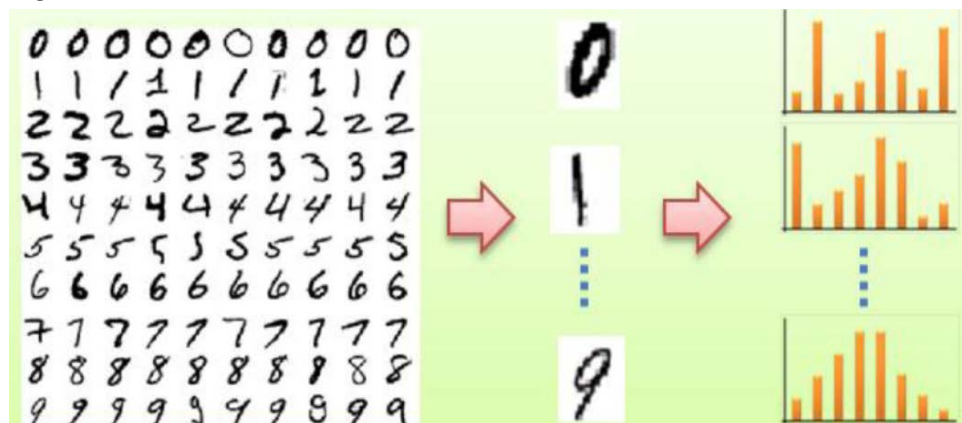
Hình 3. Những *keypoint* được phân bố trên ảnh

Mỗi đối tượng trong một hình ảnh có các *keypoint* khác nhau (Hình 3), các véc-tơ lấy *keypoint* làm điểm gốc gọi là véc-tơ đặc trưng. Véc-tơ đặc trưng này sẽ được sử dụng để nhận dạng đối tượng trong ảnh.

• **Các bước trích xuất Véc-tơ đặc trưng SIFT**

- *Bước 1.* Đọc hình ảnh đầu vào và chuyển ảnh đầu vào thành ảnh xám
- *Bước 2.* Khởi tạo bộ mô tả đặc trưng SIFT
- *Bước 3.* Định vị và tính toán *keypoint*
- *Bước 4.* Tạo véc-tơ đặc trưng SIFT trên các *keypoint*.

HOG (*Histogram of Oriented Gradient*): đặc trưng hình dạng, bất biến theo tỉ lệ vì lấy theo hướng đạo hàm của từng ô (*cell*) trên ảnh. Đặc trưng này thường ứng dụng trong nhận diện đối tượng, nhận diện vật thể, nhận diện con người, nhận dạng khuôn mặt (*face recognition*).



Hình 4. Sử dụng HOG để mô tả những đối tượng cục bộ của ảnh (Reza, 2014)

Bản chất của HOG là sử dụng sự phân bố về cường độ hoặc hướng biên để mô tả các đối tượng cục bộ. Áp dụng HOG bằng cách chia nhỏ ảnh thành các block, ở mỗi block tiếp tục chia thành các cell, tính giá trị bin và histogram về các hướng của cường độ. Kết quả thu được sau khi ghép các histogram lại với nhau là véc-tơ đặc trưng của HOG (Kulandai, 2017).

• **Các bước trích xuất Véc-tơ đặc trưng HOG**

- *Bước 1.* Resize kích thước ảnh cần lấy đặc trưng
- *Bước 2.* Khai báo các tham số để khởi tạo một bộ mô tả HOG (*nbins, cellSize, winSize, blockSize, winStride*)
- *Bước 3.* Khởi tạo bộ mô tả đặc trưng HOG
- *Bước 4.* Thực hiện tính toán đặc trưng HOG
- *Bước 5.* Nhận véc-tơ đặc trưng HOG của hình ảnh.

• **Các bước thực hiện kết hợp đặc trưng SIFT và HOG**

- *Bước 1.* Chuẩn bị danh sách mảng của véc-tơ đặc trưng SIFT và HOG.
- *Bước 2.* Tạo mảng kết hợp bằng cách ghép mảng của SIFT với mảng HOG.
- *Bước 3.* Chuẩn hóa dữ liệu của mảng kết hợp, từ việc chuẩn hóa từng phần tử trong mảng kết hợp.

Chuẩn hóa dữ liệu của véc-tơ đã kết hợp nhằm giảm dư thừa dữ liệu của véc-tơ và cải thiện tính toàn vẹn của dữ liệu. Việc chuẩn hóa dữ liệu trong đoạn 0 1 thực hiện bằng cách áp dụng công thức:

$$z_i = \frac{x_i - \min_x}{\max_x - \min_x}$$

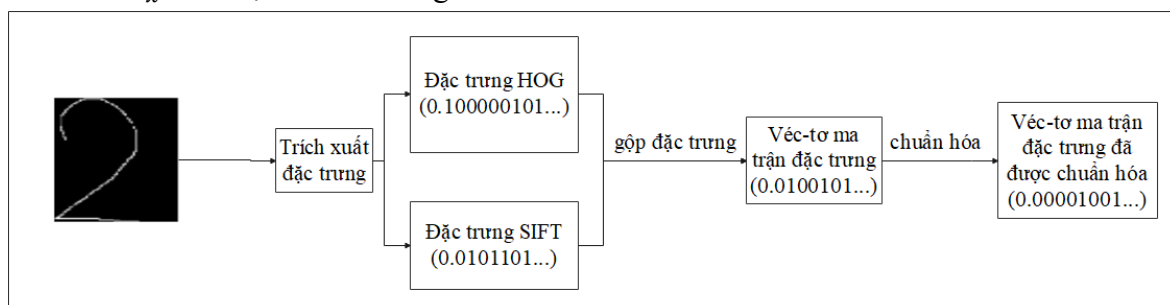
trong đó:

z_i : Giá trị chuẩn hóa thứ i trong véc-tơ

x_i : Giá trị thứ i trong véc-tơ

\min_x : Giá trị nhỏ nhất trong véc-tơ

\max_x : Giá trị lớn nhất trong véc-tơ.



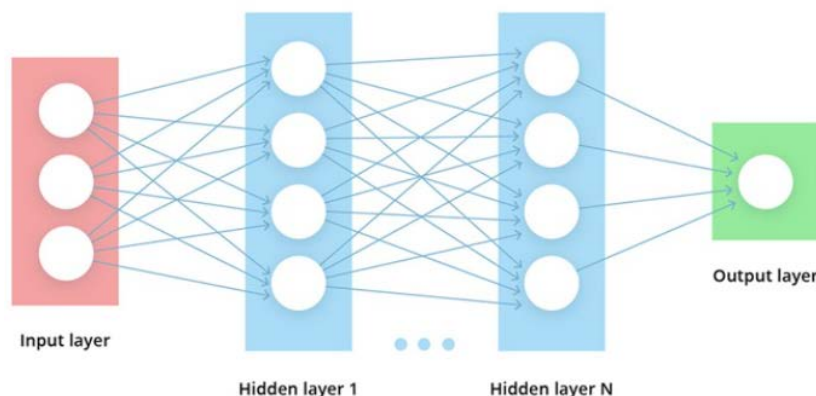
Hình 5. Thực hiện kết hợp đặc trưng

Ví dụ Hình 5 để thực hiện kết hợp đặc trưng véc-tơ đặc trưng của SIFT và véc-tơ đặc trưng HOG của một ảnh kí tự X. Đầu tiên chuẩn bị véc-tơ đặc trưng của SIFT và véc-tơ đặc trưng của HOG. Tiếp theo thực hiện ghép mảng của hai véc-tơ đặc trưng. Sau đó thực hiện

chuẩn hóa các phần tử của mảng gộp. Kết quả thu được một véc-tơ đặc trưng mới của ảnh kí tự X , được kết hợp từ véc-tơ đặc trưng của SIFT và véc-tơ đặc trưng của HOG.

D. Huấn luyện và nhận dạng

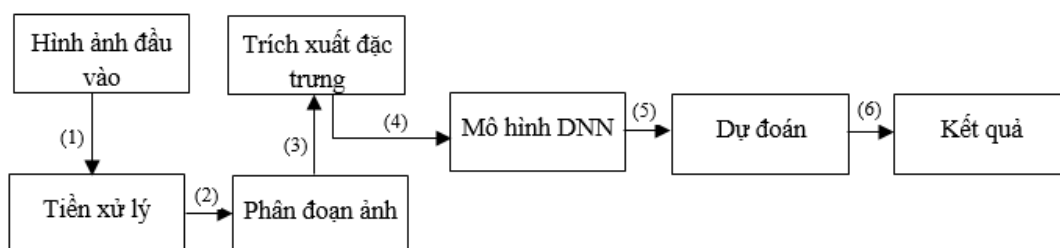
DNN là một mô hình mô phỏng dựa trên hoạt động của hệ thống mạng nơ-ron thần kinh của sinh vật. Một mô hình mạng nơ-ron nhân tạo được tạo thành từ nhiều nơ-ron đơn lẻ gọi là các nút, sự kết hợp của các nút sẽ tạo thành các tầng (*layer*), số lượng các nút ở mỗi tầng có thể khác nhau. Mạng DNN gồm các tầng: một tầng đầu vào (*Input layer*) với nhiều nút nhận dữ liệu đầu vào trực tiếp từ các điểm ảnh; nhiều tầng ẩn (*Hidden layer*) nằm giữa tầng đầu vào và tầng đầu ra (mỗi mạng nơ-ron phải có ít nhất một tầng ẩn) thực hiện huấn luyện đặc trưng từ dữ liệu đầu ra của tầng trước; một tầng đầu ra (*Output layer*) trả về dữ liệu đầu ra (Brownlee, 2019).



Hình 6. Mô hình cơ bản của mạng DNN

Các nút tầng ẩn và tầng đầu ra của mạng liên kết với toàn bộ các nút ở các tầng trước đó. Các nút ở giữa là các nút ẩn có số lượng nhiều hơn nút đầu ra.

• Mô hình nhận dạng kí tự trên mạng DNN



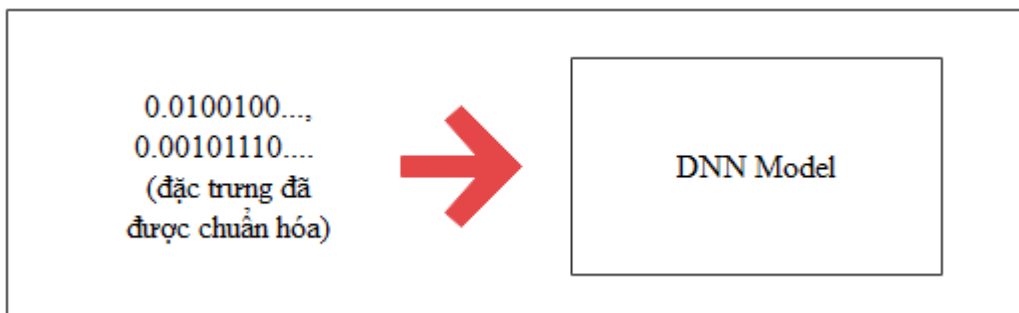
Hình 7. Mô hình nhận dạng theo DNN (Suresh, 2020)

Mô hình nhận dạng theo mạng DNN gồm:

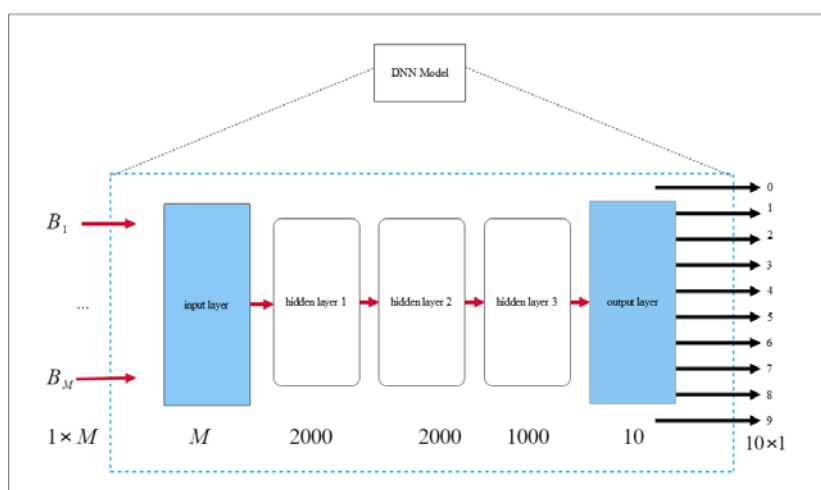
- (1) Phát hiện đối tượng chữ số viết tay, tiền xử lý để loại bỏ nhiễu và các kí tự không mong muốn;
- (2) Phân đoạn ảnh sử dụng phương pháp Graph-cut;
- (3) Kết hợp đặc trưng SIFT và HOG để trích xuất véc-tơ đặc trưng nhằm mô tả đối tượng;
- (4) Xây dựng và huấn luyện mô hình DNN để phân loại đặc trưng đầu vào;

- (5) Sử dụng mạng DNN đã huấn luyện để nhận dạng ảnh đầu vào;
- (6) Sau khi dự đoán và đưa ra kết quả tiến hành đánh giá hiệu suất dựa vào kết quả.

Dữ liệu đưa vào DNN model là các véc-tơ đặc trưng của hình ảnh đã được gộp và chuẩn hóa. DNN model sẽ nhận dạng kí tự tương ứng bằng cách so sánh với mẫu chuẩn đã được huấn luyện từ trước.

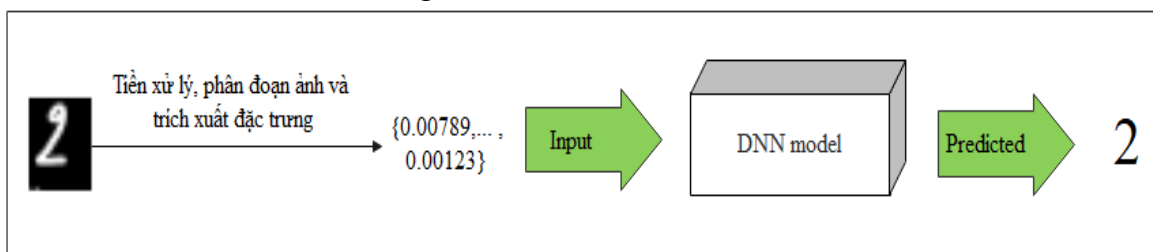


Hình 8. Hình ảnh minh họa quá trình đưa dữ liệu vào DNN model

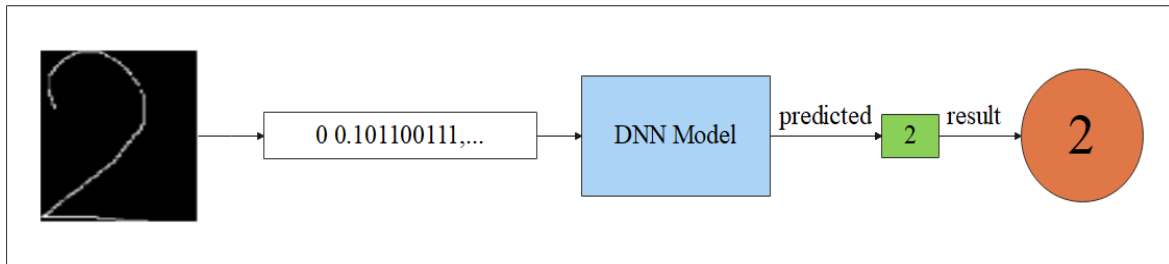


Hình 9. DNN model

Mô hình sẽ dự đoán và đưa ra một kết quả duy nhất. Kết quả này là kết quả dự đoán của hình ảnh cần được nhận dạng.



Hình 10. Minh họa quá trình dự đoán của mô hình



Hình 11. Minh họa quá trình dự đoán và trả về kết quả của mô hình

• **Quá trình huấn luyện**

- *Bước 1. Định nghĩa mạng (Define Network):*

+ Khởi tạo model chứa các tầng

```
model = Sequential()
```

+ Thêm các tầng vào mô hình

```
model.add(Dense(10))
```

- *Bước 2. Biên dịch mạng (Compile Network):* Hàm *compile* dùng để thực thi biên dịch mạng. Sử dụng thuật toán tối ưu hóa *adam*, hàm mất mát được dùng để phân loại nhiều lớp *categorical_crossentropy*.

```
model.compile(optimizer='adam', loss='categorical_crossentropy')
```

Thông tin về độ mất mát *loss* được hiển thị ở mỗi lần học (*epoch*).

```
1/2 [=====>.....] - ETA: 0s - loss: 2.3023
2/2 [=====] - 1s 7ms/step - loss: 2.3023
Epoch 2/100
```

Hình 12. Biên dịch mạng DNN

- *Bước 3. Fit Network:* Hàm *fit* được dùng để huấn luyện mạng

X là tập dữ liệu đầu vào, *y* là tập dữ liệu nhãn:

```
model.fit(X, y, batch_size=10, epochs=100, verbose=1)
```

• **Quá trình kiểm thử**

- *Bước 1. Đánh giá mạng (Evaluate Network):* thực hiện hàm *evaluate* với *X* là tập dữ liệu véc-tơ đặc trưng, *Y* là nhãn của tập dữ liệu. Kết quả thu được chỉ số tổn thất (*loss*) và độ chính xác (*accuracy*).

```
loss, accuracy = model.evaluate(X, Y, batch_size = 10, verbose = 0)
```

- *Bước 2. Dự đoán (Make Predictions):* với *X* là dữ liệu cần được *model* dự đoán và *predict* là hàm thực hiện dự đoán.

```
predictions = model.predict(X)
```


3. Kết quả và thảo luận

3.1. Môi trường thực nghiệm

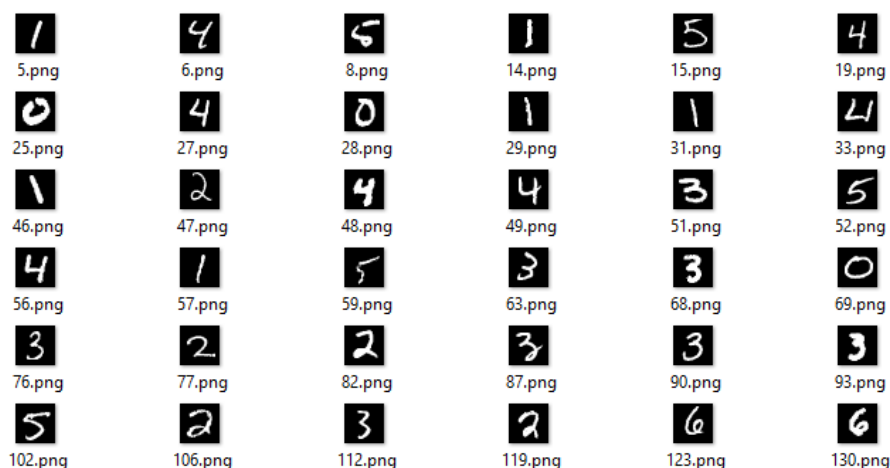
Thực nghiệm được thực thi trên máy PC CPU: Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz 3.19 GHz, RAM: 8.00 GB, hệ điều hành Windows 10 Pro 64 bit, sử dụng ngôn ngữ lập trình C# và Python 3.8.5, thư viện: EmguCV 4.5.4 và TensorFlow 2.7

3.2. Dữ liệu thực nghiệm

- Để đánh giá hiệu quả của mô hình nhận dạng, chúng tôi sử dụng tập dữ liệu ảnh chuẩn: Tập dữ liệu MNIST ở định dạng PNG có 70000 hình ảnh kí tự số viết tay gồm 10 chữ số từ 0-9 mỗi hình ảnh hình ảnh trắng đen với kích thước là 28x28.

- Ngoài ra, chúng tôi thu thập bộ dữ liệu ảnh thực tế được sinh viên viết gồm:

- + Tập dữ liệu có 1860 hình ảnh gồm các kí tự viết trên giấy với mỗi kí tự là 30 hình ảnh;
- + Tập dữ liệu có 2170 hình ảnh gồm các kí tự viết bằng phần mềm Microsoft Paint với mỗi kí tự là 30 hình ảnh.



Hình 13. Tập dữ liệu MNIST ở định dạng PNG

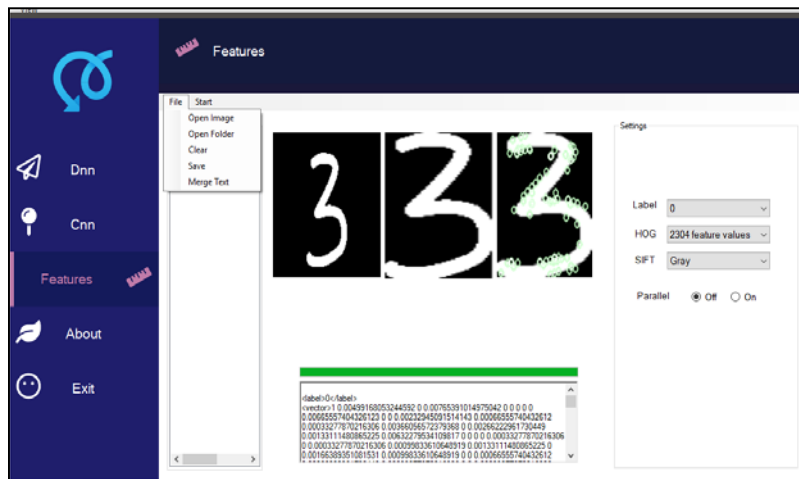


Hình 14. Tập dữ liệu viết trên giấy của sinh viên

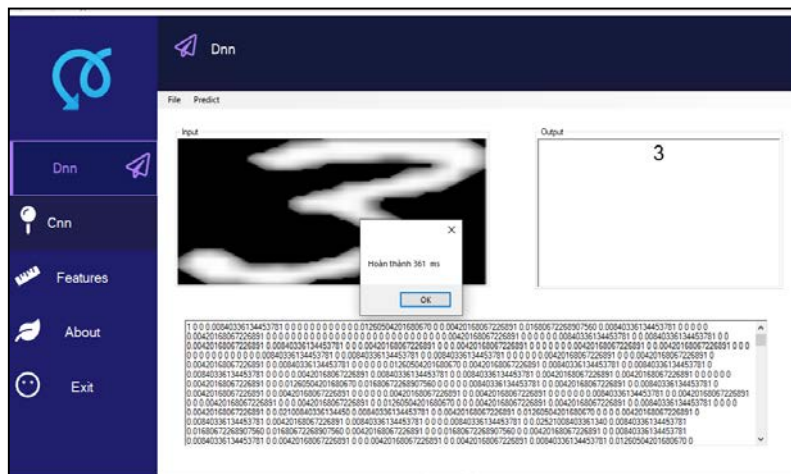


Hình 15. Tập dữ liệu viết trên Microsoft Paint của sinh viên

3.3. Kết quả thực nghiệm



Hình 16. Ứng dụng thực nghiệm của phương pháp đề xuất

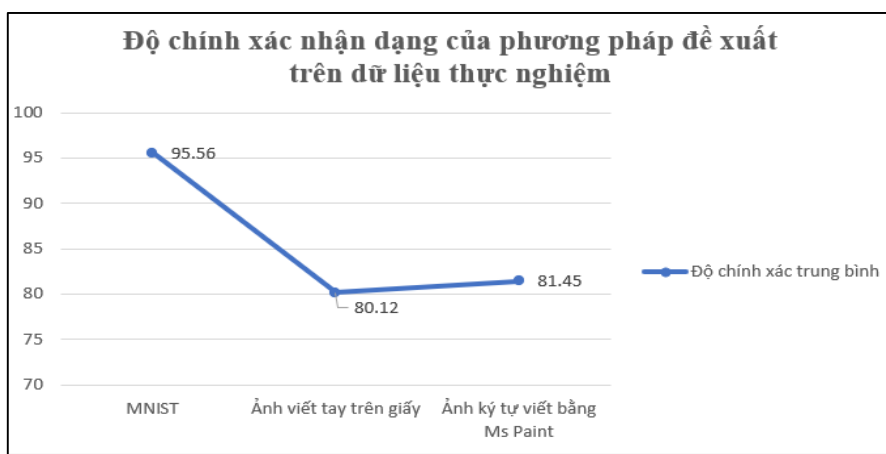


Hình 17. Kết quả nhận diện ký tự số viết tay trên ứng dụng

Kết quả thực nghiệm của phương pháp được mô tả tại Hình 17. Hiệu suất nhận dạng của phương pháp được trình bày trong Bảng 1.

Bảng 1. Độ chính xác nhận dạng của phương pháp đề xuất trên dữ liệu thực nghiệm

Tập dữ liệu ảnh	Tổng số ảnh	Số ảnh huấn luyện	Số ảnh kiểm thử	Độ chính xác trung bình
MNIST	7000	5600	1400	95,56 %
Ảnh viết tay trên giấy	1860	1488	372	80,12%
Ảnh ký tự viết bằng Ms Paint	2170	1736	434	81,45%

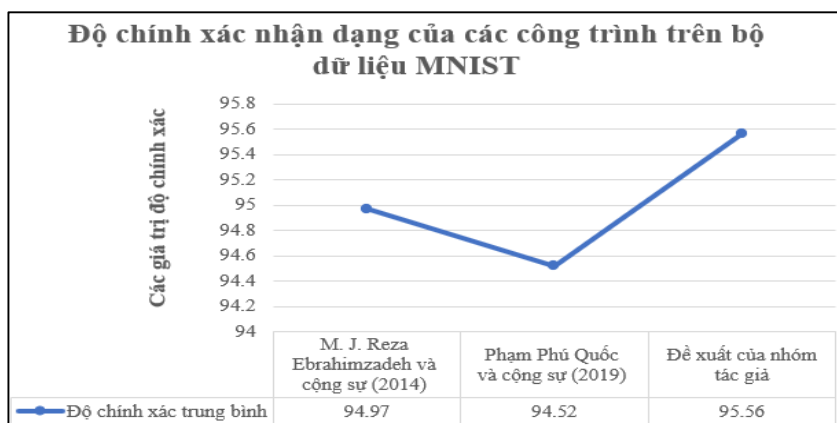


Hình 18. Độ chính xác nhận dạng của phương pháp đề xuất trên dữ liệu thực nghiệm

Chúng tôi so sánh độ chính xác của phương pháp đề xuất với công trình nhận dạng ký tự số viết tay trong những năm gần đây (Reza, 2014) trong Bảng 2. Kết quả cho thấy phương pháp đề xuất của chúng tôi là khá cao so với các công trình khác.

Bảng 2. So sánh độ chính xác nhận dạng giữa phương pháp đề xuất và các phương pháp khác trên bộ dữ liệu MNIST

Phương pháp	Độ chính xác trung bình
Reza Ebrahimzadeh et al. (2014)	94,97%
Pham et al. (2019)	94,52%
Đề xuất của nhóm tác giả	95,56%



Hình 19. So sánh độ chính xác của phương pháp đề xuất với các công trình khác trên bộ dữ liệu MNIST

4. Kết luận

Trong bài báo này, phương pháp nhận diện kí tự số viết tay đã được xây dựng. Một mô hình mạng DNN được tạo ra để thực hiện huấn luyện dựa trên kết quả trích xuất đặc trưng HOG kết hợp với đặc trưng SIFT để nhận diện ảnh kí tự số viết tay. Dựa trên phương pháp và mô hình được đề xuất, thực nghiệm được xây dựng và đánh giá kết quả. Kết quả thực nghiệm được so sánh với các phương pháp khác trên cùng một tập dữ liệu ảnh đã cho thấy phương pháp đề xuất tương đối hiệu quả. Thực nghiệm cũng cho thấy tính đúng đắn của mô hình và các thuật toán đã đề xuất, do đó phương pháp này có thể làm cơ sở để phát triển các hệ nhận diện chữ số viết tay và ứng dụng trong thực tế. Hướng phát triển tiếp theo là cải tiến phương pháp huấn luyện mạng DNN, sử dụng mạng CNN (*Convolutional Neural Networks*) để trích xuất đặc trưng và nhận dạng trực tiếp nhằm nâng cao độ chính xác.

❖ **Tuyên bố về quyền lợi:** Các tác giả xác nhận hoàn toàn không có xung đột về quyền lợi.

TÀI LIỆU THAM KHẢO

- Balas, M. B. K., Valentina, .E, & Kumar, R. (2021). *Handbook of Deep Learning in Biomedical Engineering Techniques and Applications*. Elsevier.
- Brownlee, J. (2019). *Deep Learning for Computer Vision - Image Classification, Object Detection and Face Recognition in Python*. Reserved.
- Dan, J. S. (2020). *Multi-column Deep Neural Networks for Image Classification*.
- Dong, J. (2021). *The example of graph cut algorithm*. Retrieved from https://www.researchgate.net/figure/The-example-of-graph-cut-algorithm-13_fig2_228856802
- Kulandai, J. (2017). Facial recognition using histogram of gradients and support vector machines. In 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP). IEEE
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91-110.
- Niranjani, N. (2021). *The Real Time Face Detection and Recognition System*. Retrieved from https://www.researchgate.net/publication/321669397_The_Real_Time_Face_Detection_and_Recognition_System
- Pham, P. Q., & Vuong, Q. P. (2019). Nhan dang chu so viet tay dung mang neuron nhan tao [Recognition of handwritten digits using artificial neural network]. *Journal of Science and Technology, University of Sciences, Hue University*, 14, 119-129.
- Rother, V. K. C, & Blake, A. (2004). *GrabCut: Interactive foreground extraction using iterated graph cuts*. Springer.
- Reza, E. (2014). Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM. *International Journal of Computer Applications*, 9(104), 0975-8887.

- Suresh, R. U. A., & Vimal, S. (2020). *Deep Neural Networks for Multimodal Imaging and Biomedical Applications*. United States IGI Global.
- Szikora, P. Ph. D. (2021). *Self-driving cars - The human side*. Retrieved from https://www.researchgate.net/publication/324095773_Self-driving_cars_-_The_human_side
- The Sprawls Resources. (2021). *Optimization of Medical Imaging Procedures*. Retrieved from: <http://www.sprawls.org/resources/DIGITAL>

**METHOD OF HANDWRITTEN DIGIT RECOGNITION BASED
ON DEEP NEURAL NETWORK**

Dinh Thi Man^{1*}, Nguyen Van Thinh², Nguyen The Huu¹, Tran Thi Van Anh¹

¹*Ho Chi Minh City University of Food Industry, Vietnam*

²*Ho Chi Minh City University of Food Education, Vietnam*

**Corresponding author: Dinh Thi Man – Email: mandt@hufi.edu.vn*

Received: October 11, 2022; Revised: April 25, 2023; Accepted: April 27, 2023

ABSTRACT

In this paper, a method of handwritten digit recognition is proposed based on a deep neural network (DNN). Firstly, the image dataset is extracted with HOG (Histogram of Oriented Gradient) feature combined with SIFT (Scale-invariant feature transform) feature. Then, a DNN network model is built and trained to recognize the image. Finally, the input image is automatically recognized based on the trained model. To demonstrate the effectiveness of the proposed method, the experiment was built and evaluated on the MNIST image dataset. The experimental results showed the feasibility and effectiveness of the method while making it easier to expand to other handwritten recognition.

Keywords: digit image; handwritten digit recognition; DNN; SIFT; HOG