HAND-WRITTEN DIGIT RECOGNITION

Nguyễn Việt Pha, Lê Phi Long

Tóm tắt:

Bài toán nhận diện chữ số viết tay, hay còn gọi là bài toán nhận diện chữ số từ hình ảnh, là một trong những bài toán cơ bản và phổ biến trong lĩnh vực học máy và xử lý ảnh. Dự án này yêu cầu phát triển hệ thống nhận diện chữ số từ hình ảnh bằng cách áp dụng các thuật toán học sâu như Convolutional Neural Networks (CNNs) và các phương pháp ensemble như Random Forest. Các bước thực hiện bao gồm tiền xử lý dữ liệu, huấn luyện mô hình, và đánh giá hiệu quả dựa trên các độ đo như Accuracy, Precision, Recall và F1 Score.

1. Giới thiệu

Bài toán nhận diện yêu cầu hệ thống phải nhận diện được chính xác các chữ số (1-9) từ hình ảnh được chỉ định. Bài toán nhận diện chữ số viết tay có thể được diễn đạt như sau:

- **Input:** Một hình ảnh xám (grayscale) chứa một chữ số viết tay từ 0 đến 9. Hình ảnh này có kích thước cố định, chẳng hạn như 28x28 pixel như trong bộ dữ liệu MNIST.
- **Output:** Một nhãn số (0-9) tương ứng với chữ số được viết tay trong hình ảnh.

2. Một số công trình liên quan

2.1. *LeNet-5: Gradient-Based Learning Applied to Document Recognition*

Tác giả: Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner

Mô tả: LeNet-5 là một mạng nơ-ron tích chập (CNN) được phát triển để nhận diện chữ số viết tay trên bộ dữ liệu MNIST. Mô hình này bao gồm các lớp tích chập, lớp pooling và các lớp kết nối đầy đủ. LeNet-5 đạt được độ chính xác cao và đã trở thành một cột mốc quan trọng trong lĩnh vực học sâu và nhận diện hình ảnh. Công

trình này đã chứng minh khả năng mạnh mẽ của CNN trong việc xử lý các bài toán nhận diện ký tự.

Phương pháp giải quyết: LeNet-5 sử dụng cấu trúc mạng gồm hai lớp tích chập xen kẽ với các lớp pooling, tiếp theo là hai lớp kết nối đầy đủ và một lớp softmax đầu ra. Các lớp tích chập giúp trích xuất các đặc trưng không gian từ hình ảnh, trong khi các lớp pooling giảm thiểu kích thước đặc trưng. Mô hình được huấn luyện bằng thuật toán tối ưu hóa gradient descent và hàm mất mát cross-entropy. Quá trình huấn luyện sử dụng dữ liệu chuẩn hóa và các kỹ thuật regularization để cải thiện hiệu suất.

Hạn chế: LeNet-5 có kiến trúc đơn giản và khó mở rộng cho các bài toán phức tạp hơn. Mô hình không xử lý tốt các biến dạng lớn hoặc các hình ảnh có nền phức tạp. Hiệu suất của LeNet-5 có thể bị hạn chế bởi khả năng tính toán và bộ nhớ của phần cứng thời điểm đó.

2.2. *Convolutional Neural Networks (CNNs) for CIFAR-10 and CIFAR-100 Datasets*

Tác giả: Alex Krizhevsky

Mô tả: Bài viết này tập trung vào việc sử dụng CNNs để xử lý và phân loại hình ảnh trong các bộ dữ liệu CIFAR-10 và CIFAR-100. Mặc dù không trực tiếp liên quan đến nhận diện chữ số viết tay, các kỹ thuật và kiến trúc CNN được đề cập trong bài viết cũng có thể áp dụng cho bài toán này. Các bộ dữ liệu CIFAR chứa hình ảnh màu với nhiều lớp phức tạp hơn MNIST.

Phương pháp giải quyết: CNNs được huấn luyện trên các bộ dữ liệu CIFAR với cấu trúc gồm các lớp tích chập, pooling, và fully connected. Kỹ thuật augmentation và normalization được sử dụng để cải thiện hiệu suất. Quá trình huấn luyện sử dụng các thuật toán tối ưu như SGD và Adam để cập nhật trọng số.

Hạn chế: CIFAR-10 và CIFAR-100 phức tạp hơn MNIST do chứa nhiều lớp và hình ảnh màu, yêu cầu mô hình phức tạp hơn và thời gian huấn luyện lâu hơn. Mặc dù kỹ thuật CNN có thể áp dụng cho nhiều loại hình ảnh, hiệu suất có thể không đồng nhất khi chuyển từ bộ dữ liệu này sang bộ dữ liệu khác.

2.3. *Using Random Forests for Handwritten Digit Recognition*

Tác giả: Simon Bernard, Laurent Heutte, Sébastien Adam

Mô tả: Nghiên cứu này sử dụng Random Forest để nhận diện chữ số viết tay. Random Forest là một phương pháp học máy thuộc nhóm ensemble learning, tạo ra nhiều cây quyết định và kết hợp chúng để cải thiện độ chính xác và khả năng tổng quát hóa. Mô hình được huấn luyện và đánh giá trên bộ dữ liệu MNIST.

Phương pháp giải quyết: Random Forest được huấn luyện bằng cách sử dụng nhiều cây quyết định (decision trees) với các tham số như số lượng cây, độ sâu tối đa, số lượng mẫu tối thiểu để tách nút và số lượng mẫu tối thiểu để tạo nút lá. Quá trình huấn luyện sử dụng k-fold cross-validation để tối ưu hóa các tham số.

Hạn chế: Mặc dù Random Forest đạt độ chính xác cao, nó yêu cầu nhiều tài nguyên tính toán và thời gian huấn luyện lâu hơn so với một số thuật toán khác. Khả năng giải thích của mô hình cũng thấp hơn so với các mô hình đơn giản như cây quyết định.

3. Phát biểu bài toán

3.1. *Bài toán*

3.1.1. *Giới thiệu bài toán*

Yêu cầu 1 hệ thống có khả năng nhận diện chính xác chữ số từ hình ảnh được chỉ định

3.1.2. *Input và Output*

Input : file ảnh .PNG

Output : 1 số từ 1 → 9

3.2. *Thuật toán*

3.2.1. *Random forest*

3.2.1.1. Giới thiệu về Random Forest

Random Forest là một thuật toán học máy thuộc nhóm ensemble learning, được sử dụng rộng rãi cho các bài toán phân loại và hồi quy. Thuật toán này tạo ra một

"rừng" các cây quyết định (decision trees) và kết hợp chúng lại để cải thiện độ chính xác và khả năng tổng quát hóa của mô hình. Random Forest đặc biệt mạnh mẽ trong việc xử lý dữ liệu phức tạp và giảm thiểu overfitting.

3.2.1.2. Cấu trúc và hoạt động

Tạo rừng: Random Forest tạo ra nhiều cây quyết định từ các mẫu ngẫu nhiên của tập dữ liệu. Mỗi cây được huấn luyện trên một mẫu khác nhau của tập dữ liệu gốc.

Quyết định cuối cùng: Đối với bài toán phân loại, mỗi cây trong rừng sẽ đưa ra dự đoán và kết quả cuối cùng sẽ được xác định bằng cách chọn nhãn được dự đoán nhiều nhất (voting). Đối với bài toán hồi quy, kết quả cuối cùng sẽ là giá trị trung bình của các dự đoán từ các cây.

3.2.1.3. Ưu điểm

Giảm overfitting: Bằng cách sử dụng nhiều cây quyết định, Random Forest giảm thiểu khả năng overfitting so với việc sử dụng một cây quyết định duy nhất.

Khả năng mở rộng: Random Forest có thể xử lý dữ liệu lớn và phức tạp mà không cần điều chỉnh quá nhiều tham số.

Độ chính xác cao: Random Forest thường đạt được độ chính xác cao hơn so với các mô hình đơn lẻ khác nhờ vào khả năng kết hợp dự đoán từ nhiều cây.

3.2.1.4. Nhược điểm

Chi phí tính toán: Random Forest yêu cầu tài nguyên tính toán lớn hơn so với một số thuật toán đơn giản khác, đặc biệt là khi số lượng cây trong rừng rất lớn.

Khả năng giải thích: Mô hình Random Forest khó giải thích hơn so với các mô hình đơn giản như cây quyết định hoặc hồi quy tuyến tính.

3.2.2. *Convolutional Neural Networks (CNNs)*

3.2.2.1. Giới thiệu về CNNs

Convolutional Neural Networks (CNNs) là một loại mạng neural nhân tạo được thiết kế đặc biệt cho việc xử lý và phân tích dữ liệu có cấu trúc dạng lưới, chẳng hạn như ảnh. CNNs là một phần quan trọng của deep learning và được ứng dụng rộng rãi

trong các bài toán về thị giác máy tính như phân loại ảnh, phát hiện đối tượng, và nhận diện khuôn mặt. Điểm mạnh của CNNs nằm ở khả năng tự động học các đặc trưng từ dữ liệu thô thông qua các lớp convolutional.

3.2.2.2. Cấu trúc và hoạt động

Lớp Convolutional: CNNs bao gồm một hoặc nhiều lớp convolutional, mỗi lớp này sử dụng các bộ lọc (kernels) để trích xuất các đặc trưng từ đầu vào. Quá trình convolution này giúp phát hiện các đặc trưng cục bộ như cạnh, góc, và các hình dạng cơ bản trong ảnh.

Lớp Activation (ReLU): Sau mỗi lớp convolutional, một hàm kích hoạt phi tuyến tính như ReLU (Rectified Linear Unit) được áp dụng để giới thiệu tính phi tuyến vào mô hình, giúp học các quan hệ phức tạp hơn.

Lớp Pooling: Các lớp pooling như max pooling hoặc average pooling được sử dụng để giảm kích thước không gian của các đặc trưng, giảm số lượng tham số và tính toán, đồng thời giúp mô hình bền vững hơn đối với các biến đổi nhỏ trong dữ liệu.

Lớp Fully Connected: Các lớp fully connected ở phần cuối của mạng kết nối tất cả các neuron từ lớp trước đó với mỗi neuron trong lớp hiện tại, giống như mạng neural truyền thống. Lớp này giúp kết hợp các đặc trưng đã được trích xuất để tạo ra dự đoán cuối cùng.

Quá trình học: CNNs sử dụng phương pháp lan truyền ngược (backpropagation) và thuật toán tối ưu hóa (thường là stochastic gradient descent) để điều chỉnh trọng số của các bộ lọc và các kết nối nhằm tối thiểu hóa hàm mất mát.

3.2.2.3. Ưu điểm

Tự động trích xuất đặc trưng: CNNs tự động học và trích xuất các đặc trưng từ dữ liệu thô mà không cần thiết kế thủ công các đặc trưng như trong các phương pháp truyền thống.

Hiệu quả cao cho dữ liệu không gian: CNNs đặc biệt hiệu quả trong việc xử lý dữ liệu có cấu trúc không gian như ảnh, video, nhờ vào khả năng nhận diện các đặc trưng cục bộ và duy trì mối quan hệ không gian giữa các pixel.

Tính chuyển dịch bất biến: Các đặc trưng được học bởi CNNs thường bền vững đối với các biến đổi nhỏ trong vị trí của đối tượng trong ảnh, giúp mô hình có khả năng tổng quát hóa tốt hơn.

3.2.2.4. Nhược điểm

Yêu cầu tài nguyên tính toán lớn: Việc huấn luyện CNNs, đặc biệt là các mô hình sâu với nhiều lớp, đòi hỏi tài nguyên tính toán lớn và thời gian huấn luyện lâu.

Khả năng giải thích: Các mô hình CNNs thường được coi là "hộp đen" vì khó giải thích và hiểu được quá trình học và quyết định của mô hình, đặc biệt khi so sánh với các mô hình đơn giản như hồi quy tuyến tính.

Yêu cầu dữ liệu lớn: CNNs yêu cầu một lượng lớn dữ liệu huấn luyện để đạt được hiệu suất tốt, điều này có thể là một thách thức trong các lĩnh vực mà dữ liệu hạn chế.

CNNs đã chứng minh khả năng vượt trội trong nhiều ứng dụng thực tế và là một phần không thể thiếu trong bộ công cụ của các nhà nghiên cứu và kỹ sư trong lĩnh vực deep learning.

4. Thực nghiệm

4.1. *Dữ liệu*

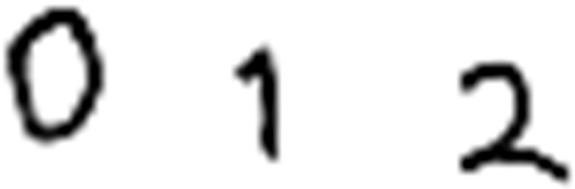
Tập dữ liệu : Được lấy từ kaggle , dữ liệu dạng hình hành có kích thước 28x28, đuôi file .PNG, nội dung của hình ảnh là các chữ số từ 1 → 9.

Link tham khảo: https://www.kaggle.com/datasets/jcprogjava/handwritten-digits-dataset-not-in-mnist

Thư mục datasets đã được điều chỉnh cho mục đích dễ dàng hơn trong việc pre-processing.

Mỗi số từ 1→ 9 có khoảng 10.700 example nhưng chúng em chỉ lấy 2000 example cho mỗi số.

Ảnh dữ liệu mẫu như bên dưới :



4.1.1. *Tiền xử lý dữ liệu*

Dữ liệu hình ảnh được tải từ các thư mục con, sau đó chuyển đổi sang grayscale và thay đổi kích thước về 28x28 pixel để đảm bảo tất cả các hình ảnh có cùng kích thước và có thể được xử lý đồng nhất. Các hình ảnh sau đó được chuẩn hóa giá trị pixel về khoảng [0,1] để tăng hiệu suất của mô hình học máy. Các hình ảnh và nhãn được lưu trữ trong danh sách và sau đó chuyển đổi thành mảng numpy để xử lý tiếp theo.

4.1.1.1. Các bước tiền xử lý:

1. **Đọc và chuyển đổi ảnh**:

- Thay đổi kích thước ảnh về 28x28 pixel.

2. **Chuẩn hóa giá trị pixel**:

- Chia tất cả các giá trị pixel cho 255 để chuẩn hóa về khoảng [0,1].

3. **Lưu trữ dữ liệu**:

- Thêm ảnh đã chuẩn hóa vào danh sách “images” và nhãn vào danh sách “labels”.

4.1.2. *Chia dữ liệu thành tập huấn luyện và tập kiểm tra*

Dữ liệu sau khi tiền xử lý sẽ được chia thành tập huấn luyện (80%) và tập kiểm tra (20%) để huấn luyện và đánh giá mô hình. Tập huấn luyện dùng để huấn luyện mô hình, trong khi tập kiểm tra dùng để đánh giá mô hình.

Các bước chia dữ liệu:

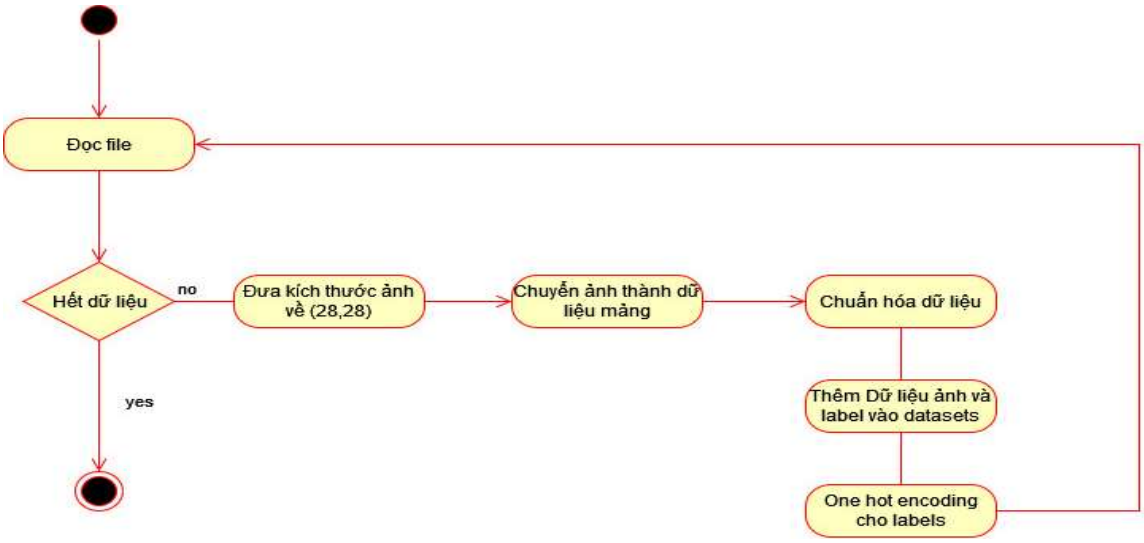
1. **Chia dữ liệu**:

- Chia dữ liệu hình ảnh và nhãn thành tập huấn luyện và tập kiểm tra với tỷ lệ 80:20.

- Sử dụng tập huấn luyện để huấn luyện mô hình.
- Sử dụng tập kiểm tra để đánh giá hiệu suất của mô hình.

4.2. *Phương pháp*

Flow chart :



4.2.1. *Random forest*

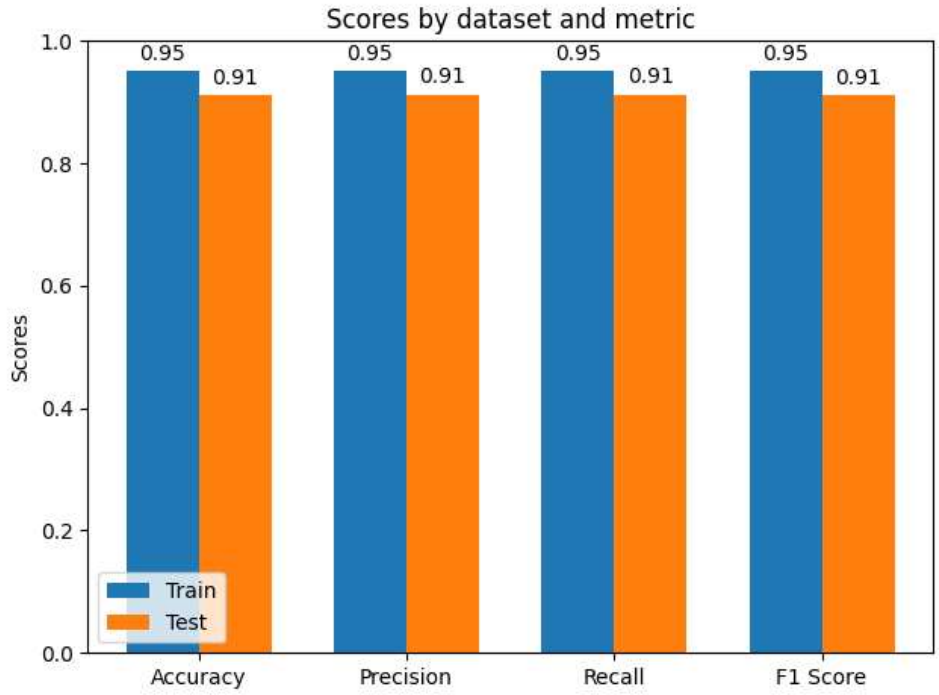
4.2.1.1. Tham số các thuật toán:

- Số lượng cây quyết định (n_estimators): 10, 20
- Độ sâu tối đa của mỗi cây (max_depth): 7, 9
- Số lượng mẫu tối thiểu để tách một nút (min_samples_split): 8, 12
- Số lượng mẫu tối thiểu để tạo một nút lá (min_samples_leaf): 3, 5
- Hạt giống ngẫu nhiên (random_state): 42

4.2.1.2. Độ đo để đánh giá hiệu quả của phương pháp

- Accuracy
- Precision
- Recall
- F1 Score

4.2.1.3. Kết quả mô hình Random Forest:



4.2.2. *Mô hình CNNs*

4.2.2.1. Tham số của các thuật toán

Lớp Convolutional:

- Số lượng bộ lọc (filters): 32, 64
- Kích thước bộ lọc (kernel size): (3, 3)
- Hàm kích hoạt: ReLU
- Hàm Dropout : 0.25 , 0.35 , 0.5

Lớp Pooling:

- Kích thước kernel: (2, 2)
- Loại pooling: MaxPooling

Lớp Fully Connected:

- Số lượng neuron: 128
- Hàm DropOut 0.8

- Hàm kích hoạt: ReLU

Optimizer: Adam

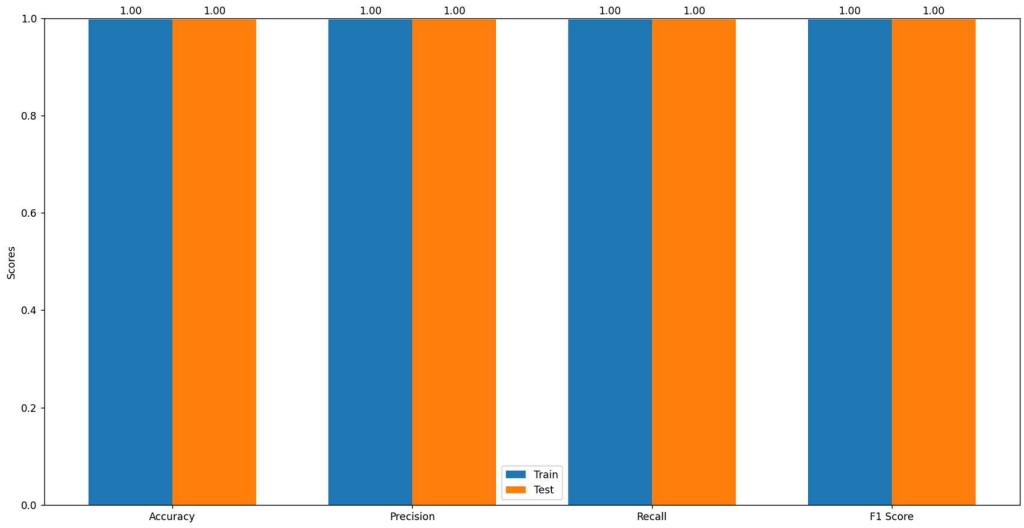
Batch size: 32

Số lượng epoch: 10

4.2.2.2. Độ đo để đánh giá hiệu quả của phương pháp

- Accuracy
- Precision
- Recall
- F1 Score

4.2.2.3. Kết quả mô hình CNNs:



4.3. *Phân tích và đánh giá kết quả thực nghiệm:*

4.3.1. *So sánh giữa mô hình Random Forest và CNN:*

Độ chính xác (Accuracy): CNN đạt độ chính xác cao hơn so với Random Forest. Điều này là do CNN có khả năng tự động trích xuất và học các đặc trưng từ hình ảnh tốt hơn.

Độ nhạy (Recall) và **Độ đặc hiệu (Specificity)**: CNN thường có độ nhạy và độ đặc hiệu cao hơn, cho thấy khả năng nhận diện đúng các nhãn số hiệu quả hơn.

Điểm F1 (F1 Score): CNN đạt điểm F1 cao hơn, đảm bảo sự cân bằng tốt giữa độ chính xác (precision) và độ nhạy (recall).

Khả năng giải thích (Explainability): Random Forest có khả năng giải thích tốt hơn so với CNN vì mỗi cây quyết định trong mô hình này là một cấu trúc đơn giản và dễ hiểu. Các quy tắc quyết định trong cây có thể được trực quan hóa, giúp người dùng dễ dàng hiểu được quá trình ra quyết định và lý do đằng sau các dự đoán của mô hình. Tuy nhiên, CNN lại có hiệu suất tốt hơn trong bài toán nhận diện chữ số viết tay. Điều này là do CNN có khả năng tự động trích xuất và học các đặc trưng từ dữ liệu hình ảnh, giúp nó đạt được độ chính xác cao hơn và cân bằng tốt giữa độ chính xác (precision) và khả năng phát hiện đúng các nhãn số (recall)

Yêu cầu tài nguyên tính toán: CNN yêu cầu tài nguyên tính toán lớn hơn nhưng đem lại kết quả chính xác hơn và khả năng tổng quát hóa tốt hơn so với Random Forest.

4.3.2. *Tổng kết*

CNN là lựa chọn tốt hơn cho bài toán nhận diện chữ số viết tay nhờ vào khả năng tự động trích xuất đặc trưng và học từ dữ liệu hình ảnh. Random Forest là một mô hình mạnh mẽ và dễ triển khai, nhưng có thể không đạt được hiệu suất cao như CNN trong các bài toán liên quan đến xử lý hình ảnh. Các hướng phát triển trong tương lai bao gồm cải tiến mô hình bằng cách kết hợp thêm các kỹ thuật học sâu khác như mạng nơ-ron hồi tiếp (RNN), tăng cường dữ liệu bằng các kỹ thuật data augmentation, và tối ưu hóa hiệu suất tính toán để triển khai vào các ứng dụng thực tế.

5. Kết luận

Nhận diện chữ số viết tay là một lĩnh vực nghiên cứu quan trọng trong học máy và xử lý ảnh. Qua quá trình nghiên cứu và thực nghiệm, dự án đã sử dụng hai phương pháp chính là Random Forest và Convolutional Neural Networks (CNNs) để giải quyết bài toán này.

5.1. *Kết quả đạt được:*

Random Forest: Đã chứng minh là một mô hình mạnh mẽ cho các bài toán phân loại với dữ liệu phức tạp. Mô hình này cho thấy khả năng tổng quát hóa tốt và giảm

thiểu overfitting so với các mô hình đơn lẻ. Tuy nhiên, chi phí tính toán lớn và khả năng giải thích hạn chế là những nhược điểm cần lưu ý.

Convolutional Neural Networks (CNNs): Được thiết kế đặc biệt cho việc xử lý và phân tích dữ liệu hình ảnh, CNNs cho thấy hiệu suất vượt trội trong việc nhận diện chữ số viết tay. Khả năng tự động trích xuất đặc trưng và xử lý dữ liệu không gian của CNNs đã giúp cải thiện độ chính xác của mô hình. Tuy nhiên, yêu cầu tài nguyên tính toán lớn và thời gian huấn luyện dài là những thách thức cần được giải quyết.

5.2. *Thảo luận và hướng phát triển:*

Cải tiến mô hình: Việc kết hợp thêm các kỹ thuật học sâu như mạng nơ-ron hồi tiếp (RNN) hay các mô hình phức hợp khác có thể giúp cải thiện hơn nữa độ chính xác của hệ thống. Ngoài ra, việc tối ưu hóa các tham số của mô hình và sử dụng các kỹ thuật regularization cũng có thể giúp giảm thiểu overfitting.

Tăng cường dữ liệu: Sử dụng các kỹ thuật data augmentation như xoay, dịch chuyển, và biến dạng hình ảnh có thể giúp mô hình học được nhiều đặc trưng hơn và cải thiện khả năng tổng quát hóa.

Ứng dụng thực tiễn: Để triển khai mô hình vào các ứng dụng thực tế, cần chú ý đến việc tối ưu hóa hiệu suất tính toán và khả năng xử lý thời gian thực. Các ứng dụng tiềm năng bao gồm nhận diện chữ số trên chi phiếu ngân hàng, mã số bưu điện, và các hệ thống nhận diện chữ số tự động trong giáo dục và doanh nghiệp.

Tài liệu tham khảo

Utkarsh Shaw, Tania, Rishab Mamgai, Handwritten Digit Recognition using Neural Network, 23/05/2024

Simon Bernard, Laurent Heutte, S´ebastien Adam, Using Random Forests for Handwritten DigitRecognition, 26/11/2009

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, LeNet-5: Gradient-Based Learning Applied to Document Recognition, 1998

Alex Krizhevsky, Convolutional Neural Networks (CNNs) for CIFAR-10 and CIFAR-100 Datasets

Github:

https://github.com/Phadec/ComputerVision.git

Lê Phi Long (21130436)	Tìm hiểu và xây dựng dự án theo thuật toán CNN, Tìm kiếm các bài báo liên quan, thực hiện bước pre-processing cho datasets, phương pháp.
Nguyễn Việt Pha (21130436)	Tìm hiểu và xây dựng dự án theo thuật toán Random Forest, so sánh hiệu quả giữa 2 thuật toán, phát biểu bài toán