|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP  **KHOA ĐIỆN TỬ**  **Bộ môn: Công nghệ thông tin**      **BÀI TẬP LỚN**    MÔN HỌC  **TRÍ TUỆ NHÂN TẠO**       |  |  |  | | --- | --- | --- | | Sinh viên | : | Đỗ Văn Hiếu | | MSSV | : | K205480106044 | | Lớp | : | 56KMT | | GVHD | : | TS. Nguyễn Tuấn Linh |             **Thái Nguyên – 2024** |

**TRƯỜNG ĐHKTCN CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**KHOA ĐIỆN TỬ*Độc lập - Tự do - Hạnh phúc***

# **BÀI TẬP LỚN**

**MÔN HỌC: TRÍ TUỆ NHÂN TẠO**

BỘ MÔN CÔNG NGHỆ THÔNG TIN

Sinh viên: Phạm Tuấn Dương MSSV: K205480106044

Lớp: 56KMT.01 Ngành: Tin học công nghiệp

Giáo viên hướng dẫn: TS. Nguyễn Tuấn Linh

Ngày giao đề tài: 16/01/2024 Ngày hoàn thành: 09/03/2024

Tên đề tài: **Xây dựng hệ thống nhận diện chữ số viết tay qua ảnh.**

Yêu cầu: (Tóm tắt yêu cầu chức năng và phi chức năng)

* Yêu Cầu Chức Năng:

+ Hệ thống nhận diện và phân loại chữ số từ ảnh đầu vào với độ chính xác cao.

+ Hỗ trợ xử lý ảnh đa dạng từ nhiều nguồn và định dạng khác nhau.

+ Giao diện dễ sử dụng, hiển thị kết quả nhận diện một cách rõ ràng.

* Yêu cầu phi chức năng:

+ Hỗ trợ điều chỉnh tham số mô hình để tối ưu hóa hiệu suất.

+ Hỗ trợ đào tạo lại mô hình với dữ liệu mới để cải thiện độ chính xác.

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

...........................................................................................................................................................................................................................

........................................................................................................................................................................................................................... ...........................................................................................................................................................................................................................

*Thái Nguyên, ngày….tháng…..năm 20....*

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

**MỤC LỤC**

LỜI NÓI ĐẦU

Trong thế kỷ 21, với sự bùng nổ của dữ liệu số và sự phát triển nhanh chóng của trí tuệ nhân tạo (AI), việc phát triển các hệ thống nhận dạng chữ số viết tay đã trở thành một lĩnh vực nghiên cứu quan trọng và hứa hẹn. Nhận dạng chữ số viết tay là quá trình nhận diện và phân loại các chữ số từ hình ảnh số viết tay được chụp hoặc quét từ các nguồn như tài liệu giấy tờ, bưu thiếp, v.v. Ứng dụng của nhận dạng chữ số viết tay rất đa dạng, từ tự động ghi chú từ hình ảnh, xử lý bưu thiếp, đến phân tích tài liệu cổ, và cả trong hệ thống nhận dạng biển số xe.

Nhận dạng chữ số viết tay không chỉ đòi hỏi sự hiểu biết vững vàng về các kỹ thuật xử lý ảnh và máy học, mà còn yêu cầu sự tiếp cận đa dạng và sáng tạo trong việc xử lý dữ liệu phức tạp và không đồng nhất. Trong bài báo cáo này, chúng tôi sẽ trình bày một giải pháp nhận dạng chữ số viết tay sử dụng các kỹ thuật và công nghệ hiện đại của trí tuệ nhân tạo, kèm theo các thử nghiệm và đánh giá kết quả.

Chúng tôi sẽ bắt đầu bằng việc đề cập đến nguồn gốc và ý nghĩa của vấn đề nhận dạng chữ số viết tay, tiếp đến là một cái nhìn tổng quan về các phương pháp và công nghệ phổ biến được sử dụng trong lĩnh vực này. Sau đó, chúng tôi sẽ trình bày chi tiết giải pháp của chúng tôi, bao gồm quá trình tiền xử lý ảnh, mô hình học máy được sử dụng, và cách tiếp cận tổng thể của hệ thống nhận dạng. Cuối cùng, chúng tôi sẽ thảo luận về các kết quả thử nghiệm và đánh giá hiệu suất của hệ thống của chúng tôi, cùng với những thảo luận và kết luận.

Bài báo cáo này hy vọng sẽ cung cấp một cái nhìn tổng quan và sâu sắc về lĩnh vực nhận dạng chữ số viết tay, đồng thời đóng góp vào sự phát triển và nghiên cứu trong lĩnh vực quan trọng này của trí tuệ nhân tạo.

Em xin chân thành cảm ơn thầy giáo TS. Nguyễn Tuấn Linh đã trang bị cho em những kiến thức cần thiết qua môn “Trí tuệ nhân tạo” để có thể hoàn thành bài tập lớn này.

Tuy nhiên do thời gian cùng với trình độ hiểu biết còn hạn hẹp nên khó tránh khỏi những thiếu sót. Rất mong nhận được sự đóng góp của quý thầy cô và các bạn về bài tập lớn của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

CHƯƠNG I: PHẦN MỞ ĐẦU

* 1. **Mục đích và ý nghĩa của bài tập lớn**

Trí tuệ nhân tạo là một lĩnh vực đang ngày càng tác động mạnh mẽ tới cuộc sống của chúng ta. Trong quá trình học tập môn học “trí tuệ nhân tạo” em cảm thấy rất hứng thú với môn học này.

Nhằm củng cố kiến thức học được em đã làm bài tập lớn này, Mục đích và ý nghĩa của bài tập giúp em hiểu hơn về trí tuệ nhân tạo nói chung và cách sử dụng nó trong quá trình xử lý ảnh, ứng dụng những kiến thức này vào bài tập thực tế.

**1.2** **Giới thiệu sơ lược về chủ đề được chọn**

Bài tập lớn về nhận dạng chữ số viết tay qua ảnh được thiết kế nhằm mục đích cung cấp cho sinh viên một cơ hội để áp dụng kiến thức đã học trong môn học Trí tuệ Nhân tạo vào một vấn đề cụ thể và thú vị. Bài tập lớn này yêu cầu sinh viên thiết kế và triển khai một hệ thống nhận dạng chữ số viết tay qua ảnh, từ việc tiền xử lý dữ liệu đến xây dựng và huấn luyện mô hình học máy.

Phác Thảo Ý Nghĩa của Việc Nghiên Cứu Nhận Dạng Chữ Số Viết Tay qua Ảnh trong Ngữ Cảnh của Trí Tuệ Nhân Tạo và Ứng Dụng của Nó trong Thế Giới Thực:

Việc nghiên cứu và phát triển các hệ thống nhận dạng chữ số viết tay qua ảnh đóng vai trò quan trọng trong việc khai phá tiềm năng của trí tuệ nhân tạo trong ứng dụng thực tế. Dưới đây là một số ý nghĩa của việc nghiên cứu này:

Cải Thiện Trải Nghiệm Người Dùng: Việc có thể nhận dạng và chuyển đổi các chữ số từ hình ảnh sẽ cải thiện trải nghiệm người dùng trong các ứng dụng như nhận diện văn bản trong hình ảnh hoặc quét thông tin từ tài liệu giấy.

Tăng Cường Hiệu Quả Làm Việc: Trí tuệ nhân tạo và học máy có thể được sử dụng để tạo ra các hệ thống nhận dạng chữ số viết tay hiệu quả, giúp tăng cường hiệu suất và tiết kiệm thời gian trong các quy trình tự động hóa công việc.

Ứng Dụng Trong Công Nghệ Cảm Biến: Việc nhận dạng chữ số viết tay qua ảnh cũng có thể được áp dụng trong các ứng dụng công nghệ cảm biến như nhận dạng chữ ký hoặc xác định số điện thoại từ hình ảnh.

Phát Triển Công Nghệ Thông Minh: Việc nghiên cứu về nhận dạng chữ số viết tay qua ảnh đóng vai trò quan trọng trong việc phát triển các công nghệ thông minh và tự động hóa, góp phần vào sự phát triển của xã hội thông minh và kỹ thuật số.

Do đó, bài tập lớn này không chỉ là cơ hội để áp dụng kiến thức được học trong môn học, mà còn là cơ hội để sinh viên hiểu rõ hơn về ý nghĩa và tiềm năng của trí tuệ nhân tạo trong cuộc sống hàng ngày và trong các ứng dụng thực tế.

**1.3 Đặt Vấn Đề**

Phát biểu vấn đề cụ thể mà bài tập lớn của bạn sẽ giải quyết, chẳng hạn như việc nhận dạng chữ số viết tay từ các ảnh số đã cho.

Nêu rõ các thách thức và khó khăn trong việc giải quyết vấn đề này.

**1.4 Mục Tiêu của Bài Tập Lớn:**

Liệt kê các mục tiêu cụ thể và định lượng mà bạn đặt ra cho bài tập lớn.

Ví dụ:

* Phát triển một hệ thống nhận dạng chữ số viết tay đạt độ chính xác trên tập dữ liệu kiểm tra không dưới 90%.
* Tối ưu hóa thời gian xử lý để đảm bảo hệ thống có thể hoạt động trong thời gian thực.
* Thử nghiệm và đánh giá hiệu suất của hệ thống trên các tập dữ liệu thực tế.

CHƯƠNG II: TỔNG QUAN LÝ THUYẾT

**2.1. Cơ Sở Lý Thuyết Trí Tuệ Nhân Tạo**

2.1.1. Khái quát về trí tuệ nhân tạo

-Trí tuệ nhân tạo là sự mô phỏng quá trình trí tuệ của con người bằng máy móc, đặc biệt là các hệ thống máy tính.

-Trí tuệ nhân tạo giúp máy móc có thể học hỏi kinh nghiệm, điều chỉnh các đầu vào mới và thực hiện các nhiệm vụ giống con người.

-Trí tuệ nhân tạo là một lĩnh vực của khoa học máy tính, nhấn mạnh việc tạo ra các máy móc thông minh, hoạt động và phản ứng như con người.

-Các ứng dụng cụ thể của AI bao gồm xử lý các ngôn ngữ tự nhiên, nhận dạng giọng nói và thị giác, quản lý hệ thống, …

2.1.2. Lịch sử phát triển

Thực ra trí tuệ nhân tạo AI được ra đời không hề sớm như đã nói, nhưng nó là thành quả tất yếu của sự phát triển khoa học và công nghệ, là giải pháp giải quyết những bài toán khó của sự phát triển loài người trong tương lai. Dưới đây chúng ta cùng điểm lại những cột mốc của lịch sử phát triển trí tuệ nhân tạo AI ngày nay, việc tiếp tục nghiên cứu và cải tiến trí tuệ nhân tạo AI về các công nghệ nền tảng đã thể hiện rõ trong các kỹ năng tự động hóa và lý luận có thể được tích hợp trong điện thoại, máy tính và máy móc…. Trí tuệ nhân tạo AI theo cách nào đang trở thành một thực tế nền tảng của thế giới hiện nay.

A human brain with a circuit board and text

Description automatically generated with medium confidence

- **1943**: Warren McCullough và Walter Pitts xuất bản "A Logical Calculus of Ideas Immanent in Nervous Activity," đề xuất mô hình toán học đầu tiên cho mạng lưới thần kinh.

- **1949**: Donald Hebb đề xuất lý thuyết về cường độ kết nối giữa tế bào thần kinh trong cuốn sách "The Organization of Behavior: A Neuropsychological Theory."

- **1950**: Alan Turing đề xuất Thử nghiệm Turing để đo độ thông minh của máy tính. Marvin Minsky và Dean Edmonds xây dựng SNARC, máy tính mạng thần kinh đầu tiên. Claude Shannon xuất bản bài báo "Lập trình máy tính để chơi cờ." Isaac Asimov xuất bản "Ba định luật về robot."

- **1952**: Arthur Samuel phát triển chương trình tự học để chơi cờ.

- **1954**: Thí nghiệm dịch máy Georgetown-IBM tự động dịch 60 câu tiếng Nga sang tiếng Anh.

- **1956**: Tại Hội nghị Dartmouth, cụm từ "trí tuệ nhân tạo" được đề cập đến. Allen Newell và Herbert Simon trình diễn Nhà lý luận logic (LT), chương trình lý luận đầu tiên.

- **1958**: John McCarthy phát triển ngôn ngữ lập trình AI Lisp.

- **1959**: Allen Newell, Herbert Simon và JC Shaw giải quyết vấn đề chung (GPS), một chương trình để mô phỏng giải quyết vấn đề của con người.

- **1966**: Báo cáo của ALPAC nêu chi tiết về thiếu tiến bộ trong nghiên cứu dịch máy.

- **1969**: DENDRAL và MYCIN, các hệ thống chuyên gia đầu tiên, được phát triển tại Stanford.

- **1972**: Xuất hiện ngôn ngữ lập trình logic PRITAL.

- **1973**: "Báo cáo Lighthill" công bố sự thất bại trong nghiên cứu AI, dẫn đến cắt giảm tài trợ.

- **1980**: Xuất hiện R1 (XCON), hệ thống chuyên gia thương mại thành công đầu tiên, kết thúc "Mùa đông AI đầu tiên."

- **1982**: Nhật Bản khởi động dự án Hệ thống máy tính thế hệ thứ năm.

- **1987-1993**: "Mùa đông AI thứ hai" do sụp đổ thị trường máy Lisp và sự xuất hiện của công nghệ điện toán đám mây.

- **1991**: DART, công cụ lập kế hoạch tự động, triển khai trong Chiến tranh vùng Vịnh.

**- 2005**: STANLEY, chiếc xe tự lái, chiến thắng DARPA Grand Challenge.

- **2008**: Google đạt thành công trong nhận dạng giọng nói và tích hợp tính năng này vào ứng dụng iPhone.

- **2011**: Watson của IBM chiến thắng Jeopardy.

- **2012**: Andrew Ng sử dụng 10 triệu video YouTube để huấn luyện mạng lưới thần kinh với Google Brain Deep Learning.

- **2014**: Google tạo ra chiếc xe tự lái đầu tiên để vượt qua bài kiểm tra lái xe của nhà nước.

- **2016**: AlphaGo của Google DeepMind đánh bại nhà vô địch thế giới cờ vây Lee Sedol.

2.1.3. Ứng dụng của trí tuệ nhân tạo

-Trí tuệ nhân tạo đã đi vào một số lĩnh vực đã thâm nhập sâu vào nhiều ứng dụng thực tiện trong xã hội.

**-AI trong chăm sóc sức khỏe**

+Ứng dụng nổi bật của trí tuệ nhân tạo AI là cải thiện sức khỏe của con người và giảm chi phí. Các bệnh viện đang áp dụng máy để chẩn đoán tốt hơn và nhanh hơn con người. Một trong những biết công nghệ tốt nhất chăm sóc sức khỏe là IBM Watson. Nó có thể hiểu ngôn ngữ tự nhiên và có khả năng trả lời các câu hỏi. Hệ thống khai thác dữ liệu bệnh nhân và các nguồn dữ liệu có sẵn khác để tạo thành một giả thuyết, sau đó đưa ra một lược đồ chấm điểm tin cậy.

+Các ứng dụng AI khác bao gồm chatbot, chương trình máy tính được sử dụng trực tuyến để trả lời các câu hỏi và hỗ trợ khách hàng, giúp sắp xếp các cuộc hẹn theo dõi hoặc hỗ trợ bệnh nhân thông qua quy trình thanh toán và trợ lý sức khỏe ảo cung cấp phản hồi y tế cơ bản.

**-AI trong kinh doanh**

+Tự động hóa quá trình robot đang được áp dụng cho các công việc có tính chất lặp đi lặp lại.

+ Các thuật toán được tích hợp vào các nền tảng phân tích và CRM để khám phá thông tin về cách phục vụ khách hàng tốt hơn.

+Chatbots đã được kết hợp vào các trang web để cung cấp dịch vụ ngay lập tức cho khách hàng.

**-AI trong giáo dục**

AI có thể tự động hóa việc chấm điểm, giúp các giáo viên có thêm thời gian. AI có thể đánh giá năng lực và quản lý sinh viên. Gia sư AI có thể cung cấp hỗ trợ cho sinh viên, đảm bảo họ hoàn thành đúng theo giáo trình từ trước. Thậm AI có thể thay thế một số giáo viên.

**-AI trong tài chính**

AI được ứng dụng trong lĩnh vực tài chính, các ứng dụng như thu thập dữ liệu cá nhân và cung cấp tư vấn tài chính.

**-AI trong pháp luật**

Quá trình khám phá, sàng lọc thông tin tài liệu trong pháp luật thường là công việc dành cho con người. Tuy nhiên tự động hóa trong quá trình này đang giúp con người sử dụng thời gian hiệu quả hơn.

**-AI trong sản xuất**

Đây là một lĩnh vực đã đi đầu trong việc kết hợp robot vào quy trình làm việc. Robot công nghiệp được sử dụng để thực hiện các nhiệm vụ đơn lẻ và giải phóng sức lao động của con người.

**2.2. Khái Niệm và Lý Thuyết Học Máy**

2.2.1. Khái niệm học máy

-Học máy là một lĩnh vực của trí tuệ nhân tạo (AI) tập trung vào việc phát triển các thuật toán và mô hình máy tính có khả năng tự học từ dữ liệu và thực hiện các nhiệm vụ mà không cần phải được lập trình cụ thể. Mục tiêu của học máy là làm cho máy tính có khả năng tự cải thiện và thích nghi với thông tin mới. Các thuật toán học máy có thể được chia thành ba loại chính: học giám sát (supervised learning), học không giám sát (Unsupervised learning), học tăng cường (Reinforcement Learning).

**Các Loại Học Máy:**

* **Học Có Giám Sát (Supervised Learning):**

**- Khái Niệm: Trong học có giám sát, mô hình được đào tạo trên một tập dữ liệu đã được gán nhãn, nghĩa là mỗi mẫu dữ liệu đi kèm với đầu ra mong muốn. Mục tiêu là học cách ánh xạ từ đầu vào đến đầu ra.**

**- Ứng Dụng: Phân loại (classification) và dự đoán (regression) là hai nhiệm vụ chính trong học có giám sát.**

* **Học Không Giám Sát (Unsupervised Learning):**

**- Khái Niệm: Trong học không giám sát, mô hình phải tự "tìm hiểu" cấu trúc trong dữ liệu mà không có nhãn. Nó có thể bao gồm việc gom nhóm (clustering), giảm chiều (dimensionality reduction), hoặc khám phá các mối quan hệ ẩn.**

**- Ứng Dụng: Gom nhóm dữ liệu tương tự, giảm chiều dữ liệu để hiểu cấu trúc, và khám phá thông tin tiềm ẩn.**

* **Học Tăng Cường (Reinforcement Learning):**

**- Khái Niệm: Trong học tăng cường, một tác nhân tương tác với môi trường và học từ trạng thái và hành động để tối ưu hóa một mục tiêu cụ thể. Tác nhân nhận được phản hồi thông qua phần thưởng hoặc hình phạt từ môi trường.**

**- Ứng Dụng: Điều khiển robot, trò chơi video, và các nhiệm vụ đòi hỏi quyết định hành động liên tục.**

**2.2.2. Lý thuyết học máy**

**- Định Nghĩa: Lý thuyết học máy nghiên cứu về cơ sở toán học và thống kê của các mô hình học máy, với mục tiêu hiểu rõ các khía cạnh như tính khả thi, khả năng tổng quát hóa, và cách đánh giá hiệu suất.**

**- Tính Khả Thi (Feasibility): Lý thuyết học máy giúp xác định khả năng giải quyết các vấn đề cụ thể thông qua phương pháp học máy.**

**- Khả Năng Tổng Quát Hóa (Generalization): Lý thuyết này nghiên cứu cách mô hình hóa có thể tổng quát hóa từ dữ liệu đào tạo sang dữ liệu mới.**

**- Đánh Giá Hiệu Suất (Performance Evaluation): Nghiên cứu về cách đánh giá và đo lường hiệu suất của mô hình học máy trên dữ liệu kiểm thử.**

**- Học Online và Học Offline: Nghiên cứu về cách mô hình có thể học từ dữ liệu mới liên tục (học trực tuyến) hoặc chỉ được đào tạo một lần (học offline).**

**- Chức Năng Mất Mát (Loss Functions): Lý thuyết giải thích về cách xây dựng các hàm mất mát để đo lường sự chênh lệch giữa dự đoán và giá trị thực tế.**

**- Overfitting và Underfitting: Nghiên cứu cách ngăn chặn hiện tượng quá mức hóa và quá đơn giản hóa trong quá trình đào tạo mô hình.**

**Tóm lại:**

**Lý thuyết học máy là nền tảng toán học và thống kê giúp hiểu rõ cơ chế hoạt động và đảm bảo tính chất của các mô hình học máy. Đồng thời, nó cung cấp cơ sở cho việc phát triển các phương pháp mới và cải thiện hiệu suất của các hệ thống học máy.**

**2.3. Thuật Toán và Mô Hình trong Học Máy**

**Học máy sử dụng một loạt các thuật toán và mô hình để giải quyết nhiều vấn đề khác nhau. Dưới đây là một trình bày về một số thuật toán và mô hình quan trọng trong lĩnh vực này:**

**Học Có Giám Sát (Supervised Learning):**

**- Regresion (Hồi Quy): Dùng để dự đoán giá trị số, ví dụ như giá nhà, dựa trên dữ liệu đã được gán nhãn.**

**- Classification (Phân Loại): Sử dụng để phân loại dữ liệu vào các lớp hoặc nhóm, chẳng hạn như phân loại email là spam hay không.**

**- Support Vector Machines (SVM): Phương pháp tìm ra ranh giới phân loại tối ưu giữa các lớp dữ liệu.**

**Học Không Giám Sát (Unsupervised Learning):**

**- Clustering (Gom Nhóm): Gom nhóm dữ liệu thành các nhóm có tính chất tương tự mà không cần nhãn.**

**- Dimensionality Reduction (Giảm Chiều Dữ Liệu): Giảm số lượng biến đầu vào trong dữ liệu giữ nguyên tính chất quan trọng.**

**- Generative Adversarial Networks (GANs): Một mô hình tạo ra dữ liệu mới thông qua sự cạnh tranh giữa hai mạng nơ-ron.**

**Học Tăng Cường (Reinforcement Learning):**

**- Q-Learning: Một phương pháp học tăng cường cơ bản, tập trung vào việc tối ưu hóa hàm giá trị Q.**

**- Deep Q Network (DQN): Sử dụng mạng nơ-ron sâu để học chính sách tối ưu.**

**- Policy Gradient Methods: Tìm cách tối ưu chính sách trực tiếp, thường sử dụng mạng nơ-ron.**

**Học Sâu:**

**- Convolutional Neural Networks (CNN): Được thiết kế đặc biệt cho xử lý hình ảnh, sử dụng lớp tích chập.**

**Học Máy Dựa Trên Instance:**

**- K-Nearest Neighbors (KNN): Phân loại hoặc dự đoán dựa trên các láng giềng gần nhất.**

**- Decision Trees (Cây Quyết Định): Quyết định dựa trên các luật logic được học từ dữ liệu.**

**2.4. Công cụ và Ngôn Ngữ Lập Trình**

Các công cụ, thư viện, và ngôn ngữ lập trình đóng vai trò quan trọng trong phát triển các ứng dụng AI và học máy. Dưới đây là một giới thiệu về một số công cụ và ngôn ngữ quan trọng trong lĩnh vực này:

**Ngôn Ngữ Lập Trình:**

- Python: Là ngôn ngữ lập trình phổ biến nhất trong cộng đồng AI và học máy. Python có cú pháp đơn giản, nhiều thư viện mạnh mẽ như NumPy, Pandas, và Scikit-learn, làm cho nó trở thành sự lựa chọn chính cho phát triển mô hình.

- R: Được sử dụng chủ yếu trong phân tích thống kê và học máy. R cung cấp nhiều gói mở rộng để thực hiện các tác vụ phức tạp và là một lựa chọn tốt cho việc xử lý và khám phá dữ liệu.

- Java: Sử dụng trong các ứng dụng lớn và có tính chất phân tán. Java được ưa chuộng trong các dự án AI đòi hỏi tính ổn định và hiệu suất cao.

**Công Cụ Phát Triển và Môi Trường:**

- Jupyter Notebooks: Môi trường phát triển tương tác cho Python, giúp hiển thị mã nguồn, kết quả, và visualizations một cách linh hoạt và trực quan.

- Google Colab: Dịch vụ cung cấp môi trường Jupyter Notebook trực tuyến, hỗ trợ GPU miễn phí, làm cho việc thử nghiệm mô hình và chia sẻ dễ dàng.

-Pycharm: PyCharm là một phần mềm được phát triển bởi JetBrains, cung cấp các công cụ cần thiết giúp các lập trình viên Python tăng năng suất làm việc. Ngoài ra, PyCharm còn được tích hợp nhiều yếu tố mở rộng khác như: biên dịch mã, tô sáng cú pháp, điều hướng project nhanh chóng, công cụ cơ sở dữ liệu và trình soạn thảo văn bản.

**Thư Viện và Framework:**

- Scikit-learn: Thư viện học máy phổ biến cho Python, cung cấp nhiều thuật toán học máy cơ bản và công cụ tiện ích cho việc tiền xử lý dữ liệu.

- Keras: Giao diện cao cấp cho việc xây dựng mô hình deep learning. Keras có thể chạy trên TensorFlow, Theano, hoặc Microsoft Cognitive Toolkit.

- NLTK (Natural Language Toolkit): Dành cho xử lý ngôn ngữ tự nhiên và được sử dụng rộng rãi trong các ứng dụng liên quan đến ngôn ngữ.

- OpenCV: Thư viện xử lý ảnh và thị giác máy tính, được sử dụng rộng rãi trong các dự án AI liên quan đến thị giác máy tính.

**Công Cụ Phân Tích Dữ Liệu:**

- Pandas: Thư viện mạnh mẽ cho phân tích và xử lý dữ liệu có cấu trúc, giúp làm cho việc làm việc với dữ liệu trở nên thuận lợi.

- Matplotlib và Seaborn: Thư viện đồ họa để tạo các biểu đồ và visualizations, hỗ trợ hiển thị thông tin một cách trực quan.

- Tableau: Công cụ phân tích dữ liệu và tạo bảng điều khiển trực quan, thích hợp cho việc tạo các báo cáo và biểu đồ phức tạp.

CHƯƠNG III: XÂY DỰNG DỮ LIỆU

**3.1. Mô tả Dữ liệu**

Trước tiên, chúng ta sẽ mô tả dữ liệu mà chúng ta thu thập và sử dụng trong dự án. Quy trình thu thập:

Phương Pháp Thu Thập: Chúng tôi đã sử dụng một phương pháp kết hợp để thu thập dữ liệu cho dự án của mình. Đầu tiên, chúng tôi đã thu thập một tập dữ liệu chứa các hình ảnh chứa các chữ số viết tay từ nhiều nguồn khác nhau trên internet, bao gồm cả cơ sở dữ liệu như MNIST và SVHN, cũng như ảnh được thu thập từ các nguồn khác như website, ứng dụng di động, v.v. Sau đó, chúng tôi đã sử dụng các công cụ và kỹ thuật xử lý ảnh để tiền xử lý và chuẩn bị dữ liệu cho việc huấn luyện mô hình.

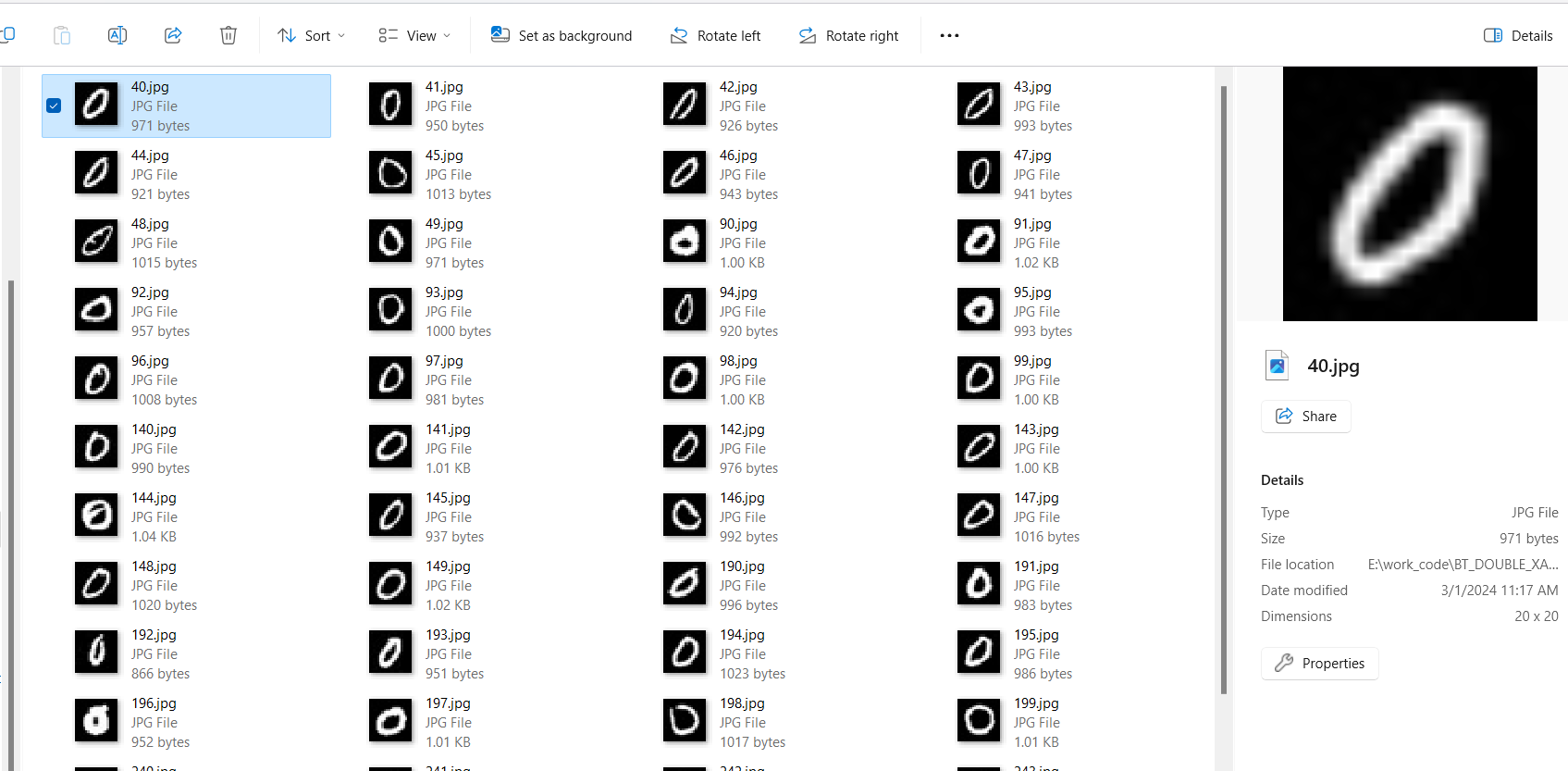
Dữ liệu thu thập qua website: https://www.kaggle.com/datasets/scolianni/mnistasjpg

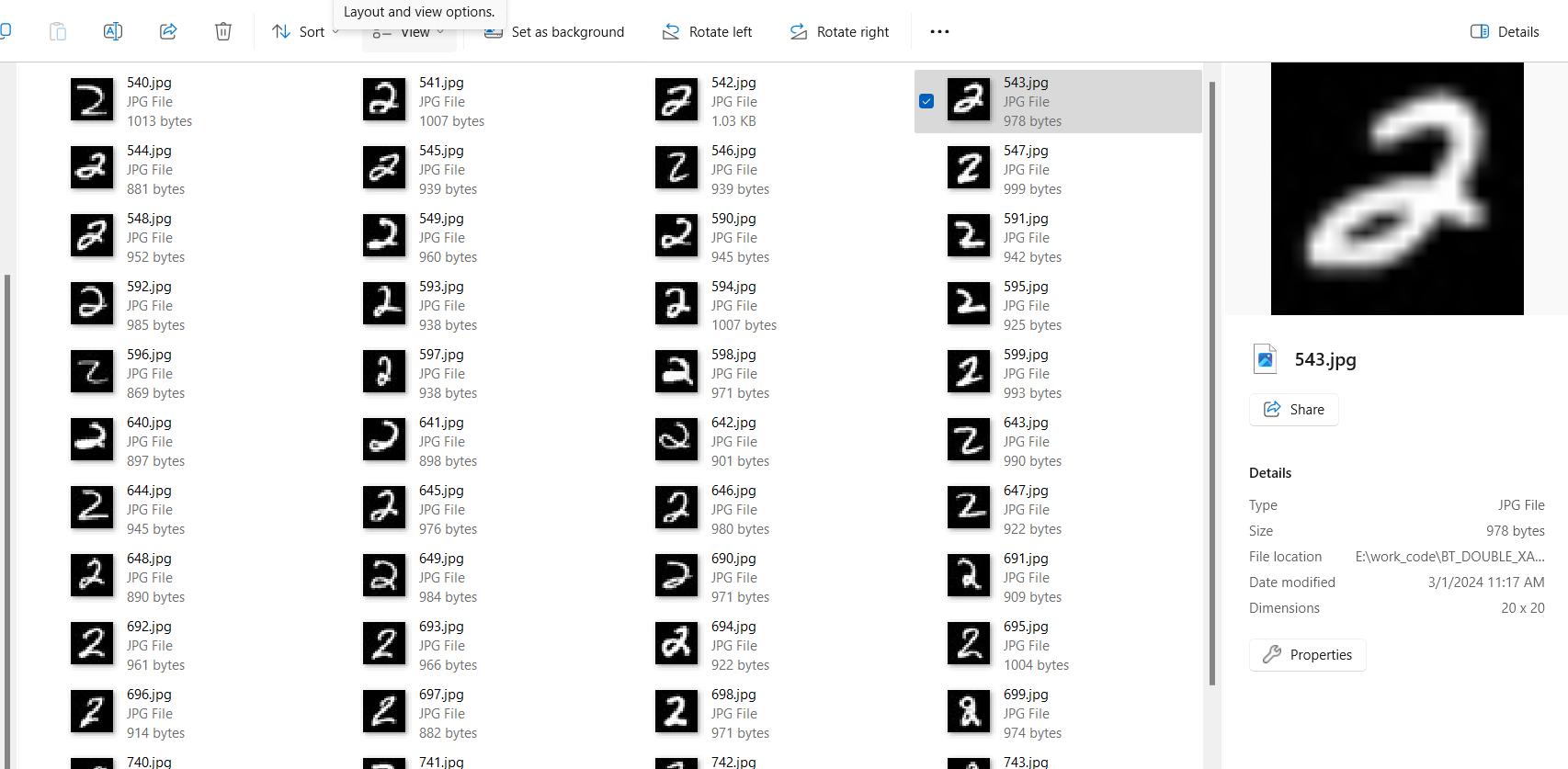
Chất Lượng Dữ liệu: Dữ liệu thu thập được đã được kiểm tra và đánh giá sơ bộ về chất lượng. Chúng tôi đã xem xét tính đầy đủ của dữ liệu (bao gồm cả số lượng mẫu và phạm vi của chữ số), độ chính xác của các nhãn được gán cho mỗi hình ảnh, và sự liên quan của dữ liệu đến vấn đề được nghiên cứu.

Kích Thước và Đặc Điểm Dữ liệu

Kích Thước: Tập dữ liệu của chúng tôi bao gồm khoảng 40.000 hình ảnh chứa chữ số viết tay. Mỗi hình ảnh có kích thước 28x28 pixel và được biểu diễn dưới dạng một ma trận số (grayscale). Tổng số lớp (chữ số từ 0 đến 9) trong tập dữ liệu là 10.

Đặc Điểm: Dữ liệu của chúng tôi chủ yếu là hình ảnh số viết tay, với mỗi hình ảnh chứa một chữ số từ 0 đến 9. Mỗi hình ảnh được biểu diễn dưới dạng giá trị pixel grayscale, trong đó giá trị pixel của mỗi điểm ảnh biểu thị mức độ sáng tối của điểm ảnh đó. Các hình ảnh này có thể chứa nhiễu, biến dạng và độ phức tạp khác nhau, phản ánh sự đa dạng của dữ liệu thực tế.





**3.2 Tiền Xử Lý Dữ Liệu**

*3.2.1 Làm sạch dữ liệu*

**Xử lý giá trị khuyết thiếu:** Có thể sử dụng các phương pháp như loại bỏ các mẫu dữ liệu có giá trị thiếu, điền giá trị thiếu bằng giá trị trung bình, median hoặc sử dụng các phương pháp ước lượng phức tạp như K-nearest neighbors.

**Xử lý nhiễu:** Các kỹ thuật như bỏ qua outliers, sử dụng các phương pháp lọc như Median Filter hoặc Gaussian Filter để giảm thiểu nhiễu.

*3.2.2 Chuyển Đổi Dữ liệu:*

**Bình thường hóa:** Chuẩn hóa dữ liệu để đảm bảo các đặc trưng có cùng phạm vi giá trị, giúp mô hình học tốt hơn.

Chuẩn hóa: Đưa dữ liệu về 1 định dạng hình ảnh đúng với kích thước, chế độ màu nhằm tối ưu

*3.2.3 Tích hợp dữ liệu*

**Hợp Nhất Dữ liệu:** Kết hợp dữ liệu từ nhiều nguồn khác nhau để tạo thành một tập dữ liệu đồng nhất. Do dữ liệu mới thu thập sẽ có sự xáo trộn không đủ hoặc thừa hay không đúng vị trí nên cần có quá trình hợp nhất lại

**Định Dạng Lại Dữ liệu:** Cấu trúc lại dữ liệu để đáp ứng với yêu cầu của mô hình hoặc phân tích, bao gồm việc chọn lọc và chuyển đổi các đặc trưng.

*3.2.4 Mô tả tập dữ liệu cuối cùng*

**Chuẩn Bị Đặc Trưng:** Lựa chọn và chuẩn bị các đặc trưng để đưa vào mô hình, bao gồm việc rút trích, biến đổi và kết hợp các đặc trưng.

**Chia Tập Dữ Liệu:** Phân chia tập dữ liệu thành tập huấn luyện, tập xác thực và tập kiểm tra. Tập huấn luyện được sử dụng để huấn luyện mô hình, tập xác thực được sử dụng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện và tập kiểm tra được sử dụng để đánh giá hiệu suất cuối cùng của mô hình.

Đảm bảo dữ liệu tốt nhất: Dữ liệu sau khi xử lý trước khi đưa vào mô hình cần đảm bảo đủ và đúng, phân loại tập dữ liệu huấn luyện mô hình và tập dữ liệu kiểm tra mô hình sau này.

**CHƯƠNG IV: PHÁT TRIỂN MÔ HÌNH**

**4.1. Lựa chọn mô hình**

Tiêu chí lựa chọn:

* **Độ chính xác:** Mô hình cần có khả năng đưa ra dự đoán chính xác trên tập dữ liệu kiểm tra.
* **Thời gian huấn luyện:** Mô hình cần phải có thời gian huấn luyện hợp lý và không quá lớn.
* **Hiểu biết của mô hình:** Mô hình cần phải có khả năng học được các đặc trưng phức tạp trong dữ liệu.

Danh Sách Mô Hình Được Cân Nhắc:

* **Mô hình tuyến tính:** Linear Regression, Logistic Regression.
* **Mô hình cây quyết định:** Decision Trees, Random Forests.
* **Mô hình máy vector hỗ trợ:** Support Vector Machines.
* **Mạng nơ-ron sâu (DNN):** Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), Gated Recurrent Units (GRU).

Lý Do Chọn Mô Hình:

* **Chọn mô hình CNN:** Vì dữ liệu đầu vào là hình ảnh, và CNN đã được chứng minh là hiệu quả trong việc xử lý dữ liệu ảnh.
* **Ưu điểm của CNN:** CNN có khả năng tự động rút trích các đặc trưng từ ảnh, giúp giảm thiểu công việc tiền xử lý dữ liệu.
* **Khả năng học đặc trưng phức tạp:** CNN có khả năng học các đặc trưng phức tạp và không gian đặc trưng của ảnh, điều này có thể giúp cải thiện hiệu suất của mô hình trong việc nhận diện và phân loại ảnh.

**4.2 Hoàn thành mô hình và huấn luyện**

Sau khi hoàn thành quá trình chuẩn bị dữ liệu và lựa chọn mô hình CNN thì cần bắt đầu viết mã nguồn nhằm tạo quá trình trainning từ những dữ liệu đã có, dưới đây là mã chương trình python train mô hình:

**Mô hình trên được xây dựng với các phần như sau:**

* **Import các thư viện và module cần thiết**

|  |
| --- |
| import os  os.environ["TF\_ENABLE\_ONEDNN\_OPTS"] = "0"  from keras.models import Sequential  from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense  from keras.preprocessing.image  import ImageDataGenerator  import matplotlib.pyplot as plt |

**os**: Thư viện os trong Python cung cấp các hàm cho việc tương tác với hệ thống máy tính, chẳng hạn như đọc/ghi tệp, thao tác với thư mục, v.v.

**os.environ**: Biến môi trường trong Python được sử dụng để truy cập các biến môi trường của hệ thống. Trong đoạn mã này, os.environ được sử dụng để thiết lập một biến môi trường có tên TF\_ENABLE\_ONEDNN\_OPTS thành '0', với mục đích tắt tính năng OneDNN khi sử dụng framework Keras. Điều này có thể làm giảm một số lỗi khi sử dụng Keras trên một số hệ thống.

**keras.models**: Từ thư viện keras.models, chúng ta import class Sequential, cho phép chúng ta xây dựng một mô hình mạng nơ-ron tuần tự bằng cách thêm các layer lần lượt.

**keras.layers**: Từ thư viện keras.layers, chúng ta import các lớp mạng nơ-ron cơ bản như Conv2D, MaxPooling2D, Flatten, và Dense. Các lớp này được sử dụng để xây dựng các kiến trúc mạng nơ-ron CNN.

**keras.preprocessing.image**: Thư viện keras.preprocessing.image cung cấp các công cụ để tiền xử lý ảnh trong quá trình huấn luyện mô hình, bao gồm ImageDataGenerator để tạo dữ liệu tăng cường và chuẩn hóa.

**matplotlib.pyplot**: Thư viện matplotlib.pyplot được sử dụng để vẽ biểu đồ và trực quan hóa dữ liệu, trong trường hợp này, là biểu đồ loss và accuracy của mô hình qua các epoch trong quá trình huấn luyện.

Các thư viện này kết hợp lại với nhau để xây dựng, huấn luyện, và đánh giá một mô hình CNN sử dụng framework Keras.

* **Xây dựng mô hình CNN**

|  |
| --- |
| # Khởi tạo mô hình CNN  model = Sequential()  model.add(Conv2D(32,(3,3),activation="relu",  input\_shape=(20, 20, 3)))  model.add(MaxPooling2D((2, 2)))  model.add(Conv2D(64, (3, 3), activation="relu"))  model.add(MaxPooling2D((2, 2)))  model.add(Conv2D(64, (3, 3), activation="relu"))  model.add(Flatten())  model.add(Dense(64, activation="relu"))  model.add(Dense(10, activation="softmax")) # 10 lớp đầu ra cho các số từ 0 đến 9 |

Định nghĩa một mô hình Convolutional Neural Network (CNN) sử dụng framework Keras.

**Conv2D**: Lớp convolutional với 32 bộ lọc kích thước 3x3 và hàm kích hoạt ReLU.

**MaxPooling2D**: Lớp max pooling với kích thước cửa sổ 2x2.

**Conv2D**: Lớp convolutional tiếp theo với 64 bộ lọc kích thước 3x3 và hàm kích hoạt ReLU.

**MaxPooling2D**: Lớp max pooling tiếp theo với kích thước cửa sổ 2x2.

**Conv2D**: Lớp convolutional tiếp theo với 64 bộ lọc kích thước 3x3 và hàm kích hoạt ReLU.

**Flatten**: Lớp chuyển đổi đầu ra của các lớp convolutional thành một vectơ 1 chiều.

**Dense**: Lớp kết nối đầy đủ với 64 nơ-ron và hàm kích hoạt ReLU.

**Dense**: Lớp kết nối đầy đủ cuối cùng với 10 nơ-ron và hàm kích hoạt softmax, tạo ra xác suất phân loại cho 10 lớp đầu ra (từ 0 đến 9).

Mô hình này bao gồm các lớp convolutional để trích xuất đặc trưng từ ảnh, các lớp max pooling để giảm kích thước của đặc trưng, sau đó sử dụng các lớp dense để phân loại ảnh vào các lớp đầu ra tương ứng.

* **Compile mô hình**

|  |
| --- |
| # Compile mô hình  model.compile(optimizer="adam", loss="categorical\_crossentropy", metrics=["accuracy"]) |

"biên dịch" một mô hình trong deep learning, thường là thông qua việc sử dụng hàm compile() trong Keras hoặc tương tự trong các framework khác, nó thiết lập cách mô hình sẽ được huấn luyện, cụ thể là:

* Optimizer: Thuật toán tối ưu hóa được sử dụng để điều chỉnh các tham số của mô hình trong quá trình huấn luyện.
* Loss function: Hàm mất mát được sử dụng để đo lường sự khác biệt giữa dự đoán của mô hình và giá trị thực tế.
* Metrics: Các chỉ số được sử dụng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện.

Việc này chỉ là bước chuẩn bị trước khi bắt đầu quá trình huấn luyện. Sau khi mô hình đã được biên dịch, bạn cần gọi phương thức fit() hoặc fit\_generator() để thực sự huấn luyện mô hình trên dữ liệu.

* **Tạo dữ liệu tăng cường (data augmentation) cho việc huấn luyện**

|  |
| --- |
| # Tạo các đối tượng ImageDataGenerator để tăng cường dữ liệu huấn luyện  train\_datagen = ImageDataGenerator(rescale=1.0 / 255) # Tái tổ chức lại giá trị pixel về khoảng 0-1 |

Trong deep learning, việc chuẩn hóa dữ liệu là một phần quan trọng của quá trình tiền xử lý dữ liệu. Trong trường hợp của ảnh, việc chuẩn hóa thường bao gồm tái tổ chức lại giá trị pixel từ khoảng 0 đến 255 (phạm vi giá trị pixel thường từ 0 đến 255) về khoảng 0 đến 1.

Lý do chính để chuẩn hóa dữ liệu như vậy là để đảm bảo rằng các giá trị đầu vào có phạm vi tương tự và nằm trong khoảng nhất định, giúp cho quá trình huấn luyện mô hình diễn ra hiệu quả hơn. Nếu không chuẩn hóa, các giá trị pixel có thể có phạm vi khác nhau và có thể gây ra hiện tượng học không ổn định hoặc hội tụ chậm trong quá trình huấn luyện.

Trong đoạn mã bạn đưa ra, ImageDataGenerator được sử dụng để tạo ra dữ liệu tăng cường (data augmentation) cho quá trình huấn luyện. Điều này có thể bao gồm việc xoay, phóng to, thu nhỏ, lật ảnh, và các biến đổi khác để tăng cường dữ liệu và giúp mô hình học được các đặc trưng tổng quát hơn. Cùng lúc đó, việc tái tổ chức lại giá trị pixel về phạm vi 0-1 cũng là một phần của tiền xử lý dữ liệu trước khi đưa vào mô hình huấn luyện.

* **Định nghĩa thông tin về dữ liệu huấn luyện**

|  |
| --- |
| train\_generator = train\_datagen.flow\_from\_directory(  "images/train",  target\_size=(20, 20),  batch\_size=batch\_size,  class\_mode="categorical",  ) |

Hàm flow\_from\_directory() của lớp ImageDataGenerator trong Keras được sử dụng để tạo ra một dữ liệu tăng cường từ các ảnh được lưu trữ trong các thư mục phân loại theo lớp. Dưới đây là các tham số chính của hàm này:

* directory: Đường dẫn đến thư mục chứa dữ liệu huấn luyện. Trong trường hợp này, là "images/train", nơi chứa các thư mục con chứa ảnh của các lớp khác nhau.
* target\_size: Kích thước mà ảnh sẽ được resize về trước khi đưa vào mô hình. Trong trường hợp này, các ảnh sẽ được resize về kích thước 20x20 pixels.
* batch\_size: Số lượng mẫu trong mỗi batch khi huấn luyện mô hình.
* class\_mode: Chế độ lớp của dữ liệu. Trong trường hợp này, được chỉ định là "categorical", cho biết rằng các nhãn của dữ liệu sẽ được mã hóa one-hot.

Khi gọi flow\_from\_directory(), nó sẽ tự động quét các thư mục trong "images/train", phân loại các ảnh theo tên thư mục, và tạo ra các batch dữ liệu có kích thước đã chỉ định. Điều này giúp cho việc sử dụng dữ liệu huấn luyện trở nên dễ dàng và linh hoạt trong quá trình huấn luyện mô hình.

* **Huấn luyện mô hình**

|  |
| --- |
| history = model.fit(  train\_generator,  steps\_per\_epoch=steps\_per\_epoch, # Số lượng batch cần phải tạo ra  epochs=10,  ) # Số lần lặp lại huấn luyện |

Thực hiện việc huấn luyện mô hình với dữ liệu được tạo ra từ train\_generator, cùng các thông số cụ thể như sau:

* train\_generator: Đây là luồng dữ liệu dùng để cung cấp dữ liệu huấn luyện cho mô hình. Trong mỗi lần lặp, một batch dữ liệu sẽ được lấy từ train\_generator và được sử dụng để huấn luyện mô hình.
* steps\_per\_epoch: Đây là số lượng batch dữ liệu cần được tạo ra từ train\_generator trong mỗi epoch. Nếu steps\_per\_epoch không được chỉ định, Keras sẽ tự động tính toán số lượng batch dựa trên số lượng dữ liệu huấn luyện và kích thước batch đã được chỉ định trước đó.
* epochs: Đây là số lần lặp lại huấn luyện trên toàn bộ tập dữ liệu. Mỗi epoch là một vòng lặp qua toàn bộ tập dữ liệu huấn luyện một lần. Mục tiêu của việc lặp lại này là để cải thiện hiệu suất của mô hình theo thời gian.

Quá trình huấn luyện sẽ diễn ra trong 10 epochs theo cách đã mô tả, với mỗi epoch, mô hình sẽ được cập nhật các tham số sao cho giảm thiểu hàm mất mát và cải thiện độ chính xác trên dữ liệu huấn luyện. Kết quả của quá trình huấn luyện sẽ được lưu lại trong biến history để sau này có thể truy cập và phân tích.

* **Vẽ biểu đồ loss và accuracy**

|  |
| --- |
| loss = history.history["loss"]  accuracy = history.history["accuracy"]  plt.plot(loss, label="Loss", color="#ff3300")  plt.plot(accuracy, label="Accuracy", color="#00b300")  plt.title("Training Loss and Accuracy")  plt.xlabel("Epoch")  plt.ylabel("Rate")  plt.legend()  epochs = range(10)  plt.xticks(epochs, [str(i + 1) for i in epochs])  plt.show() |

Sử dụng thư viện matlotlib vẽ biểu đồ biểu diễn sự thay đổi của loss và accuracy qua các epoch trong quá trình huấn luyện mô hình.

* **Lưu mô hình đã huấn luyện**

|  |
| --- |
| model.save("litterTrain.keras")  print("Đã lưu: litterTrain.keras") |

Sử dụng phương thức save() để lưu mô hình đã huấn luyện vào file với đuôi mở rộng .keras

* **Kết quả huấn luyện**



Đây là kết quả thu được sau quá trình huấn luyện 10 epoch, có thể quan sát rằng chỉ số sai(loss) đã giảm dần và thành công(accurary) đã tăng dần cho thấy mô hình đang đã huấn luyện rất tốt

**4.3 Thử Nghiệm** **và đánh giá mô hình**

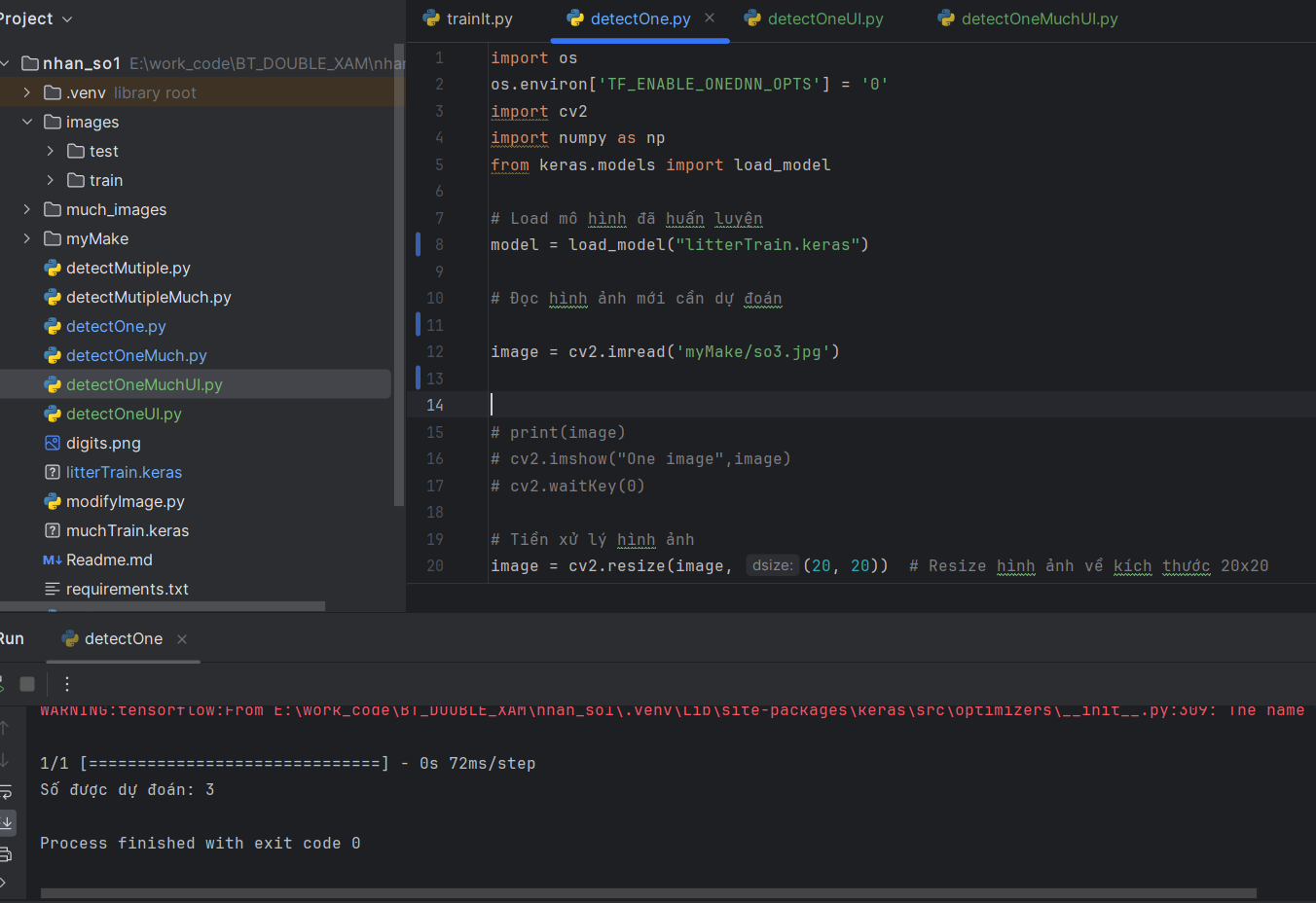
*4.3.1 Thử nghiệm*

Sau khi đã hoàn thành quá trình huấn luyện, tạo ra file modal thì bước tiếp theo là cần sử dụng model này nhận diện các ảnh trong tập test đã chuẩn bị ở bước trước

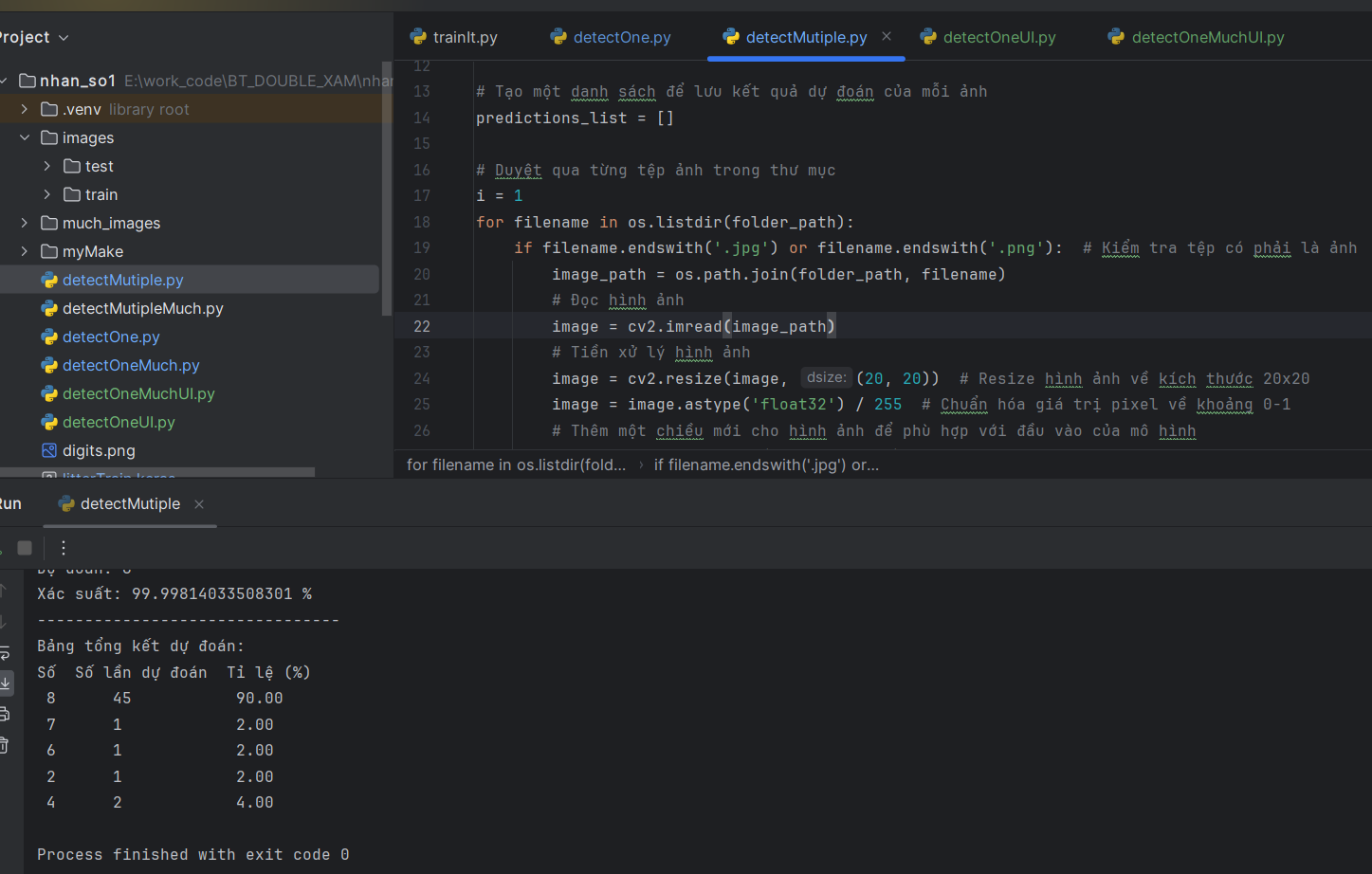
Dưới đây là mã code sử dụng mô hình nhận diện:

|  |
| --- |
| import os  os.environ["TF\_ENABLE\_ONEDNN\_OPTS"] = "0"  import cv2  import numpy as np  from keras.models import load\_model  # Load mô hình đã huấn luyện  model = load\_model("number\_recognition\_model.keras")  # Đọc hình ảnh mới cần dự đoán  image = cv2.imread("myMake/so3.jpg")  # print(image)  # cv2.imshow("One image",image)  # cv2.waitKey(0)  # Tiền xử lý hình ảnh  image = cv2.resize(image, (20, 20)) # Resize hình ảnh về kích thước 20x20  image = image.astype("float32") / 255 # Chuẩn hóa giá trị pixel về khoảng 0-1  # Thêm một chiều mới cho hình ảnh để phù hợp với đầu vào của mô hình  image = np.expand\_dims(image, axis=0)  # Dự đoán số trên hình ảnh mới  predictions = model.predict(image)  # Lấy số có xác suất cao nhất  predicted\_number = np.argmax(predictions)  print("Số được dự đoán:", predicted\_number) |

Đoạn code trên đọc modal đã huấn luyện sau đó đưa đường dẫn hình ảnh mới cần nhận diện vào, với đầu vào là số ảnh số 3, thì kết quả đã cho ra chính xác như sau:



Ngoài ra để quá trình kiểm thử nghiệm modal qua nhiều lần test hơn và có kết quả trực quan ta cũng có thể nhận diện nhiều ảnh cùng 1 lúc và cho ra thống kê kết quả. Dưới đây là kết quả modal nhận diện số 8 qua 50 mẫu thử:



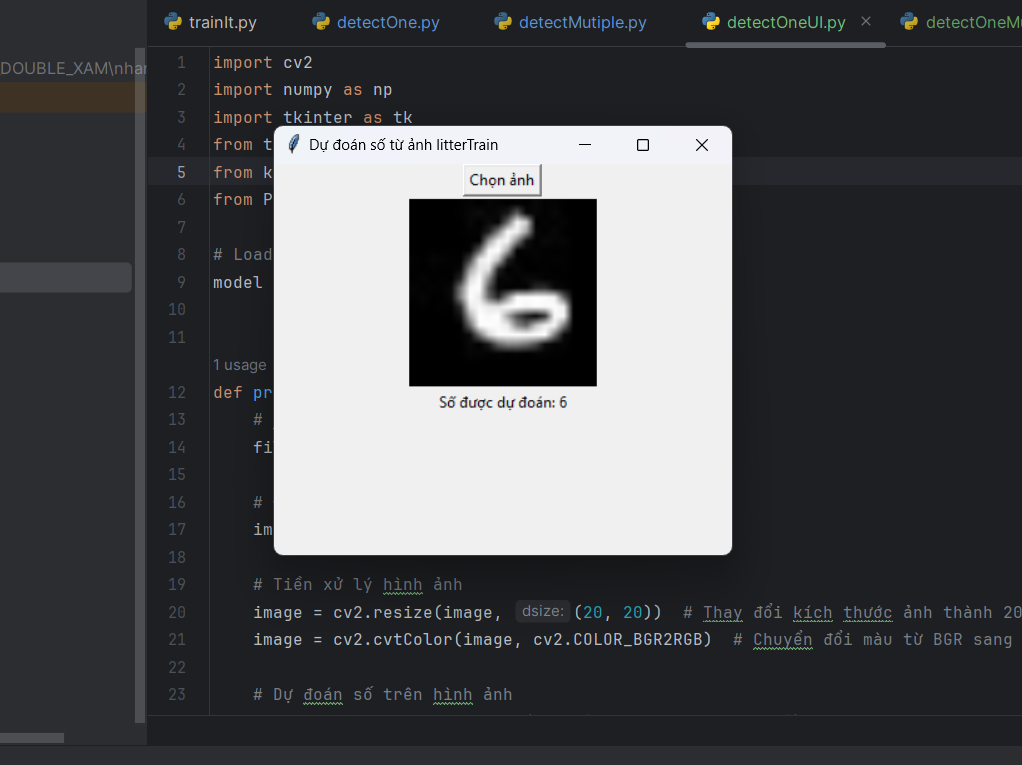
Có thể thấy tỉ lệ nhận diện đúng của chương trình đã đến 90% qua huấn luyện 10 epoch

*4.3.2: Xây dựng giao diện*

Xây dựng giao diện người dùng trong các ứng dụng là quan trọng vì nó tạo ra một giao diện tương tác giữa người dùng và ứng dụng, cho phép người dùng tương tác với ứng dụng một cách dễ dàng và trực quan.

Việc xây dựng giao diện cho phép người dùng chọn một hình ảnh từ hệ thống tệp của họ và sau đó hiển thị hình ảnh đó cùng với kết quả dự đoán số từ mô hình đã được huấn luyện. Điều này tạo ra một cách tương tác giữa người dùng và mô hình machine learning, giúp người dùng hiểu được cách mô hình hoạt động và cách nó có thể được áp dụng trong thực tế.

Ứng dụng thư viện tkinter trong python, chúng ta có thể xây dựng 1 giao diện đơn giản nhằm trực quan hóa, tăng tương tác, hiệu quả trong quá trình đánh giá mô hình:



*4.3.3: Đánh giá mô hình*

Mô hình đã tạo và qua quá trình thử nghiệm cho thấy đáp ứng yêu cầu nhận diện chữ số qua hình ảnh, tỉ lệ chính xác là 90% trở lên.

Nhưng vẫn còn những sai sót trong mô hình dẫn đến khả năng nhận diện vẫn chưa tốt nhất là do:

* Dữ liệu nhiễu: Dữ liệu chứa nhiễu hoặc thông tin không chính xác có thể ảnh hưởng đến khả năng học của mô hình.
* Không phù hợp với mô hình: Sự không phù hợp giữa mô hình và dữ liệu có thể làm giảm độ chính xác. Đôi khi một mô hình đơn giản hơn hoặc phức tạp hơn có thể phù hợp hơn với dữ liệu cụ thể.
* Thiếu huấn luyện đúng: Quá trình huấn luyện mô hình không được thực hiện đúng cách, bao gồm việc chọn siêu tham số không chính xác, việc sử dụng một thuật toán không phù hợp, hoặc việc áp dụng các kỹ thuật huấn luyện không hiệu quả.

**CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

Trong bài tập lớn này, em đã thực hiện xây dựng và đánh giá một mô hình máy học để nhận diện và dự đoán các số từ hình ảnh. Quá trình này bao gồm các bước như chuẩn bị dữ liệu, xây dựng mô hình CNN, huấn luyện mô hình trên dữ liệu, và đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

Kết quả của bài tập lớn đã cho thấy một mô hình có khả năng dự đoán số từ hình ảnh với độ chính xác đáng kể. Qua đó, mô hình này có thể được ứng dụng trong nhiều lĩnh vực như nhận diện ký tự số trong ứng dụng quét mã vạch, hệ thống tự động nhận dạng biển số xe, hoặc các ứng dụng trong lĩnh vực chẩn đoán hình ảnh y tế.

Việc áp dụng mô hình CNN để nhận diện và dự đoán số từ hình ảnh là một ứng dụng phổ biến và có ý nghĩa trong thực tế. Bằng việc xây dựng và đánh giá một mô hình như vậy, chúng ta có thể hiểu sâu hơn về cách thức hoạt động của các mô hình học sâu và cách áp dụng chúng trong các bài toán cụ thể.

Để phát triển và cải tiến cho tương lai, có một số hướng nghiên cứu và cải tiến có thể được đề xuất:

* Nâng cao độ chính xác: Tìm kiếm các kỹ thuật mới để cải thiện độ chính xác của mô hình, bao gồm việc tinh chỉnh siêu tham số, sử dụng các kiến trúc mạng phức tạp hơn, và kết hợp các kỹ thuật mới như transfer learning.
* Xử lý dữ liệu không cân bằng: Phát triển các phương pháp để xử lý dữ liệu mất cân bằng trong quá trình huấn luyện mô hình, bao gồm việc sử dụng các kỹ thuật tái cân bằng lớp hoặc phương pháp tạo dữ liệu nhân tạo.
* Tăng cường tính tổng quát hóa: Nghiên cứu các phương pháp để cải thiện khả năng tổng quát hóa của mô hình, bao gồm việc sử dụng kỹ thuật regularization, dropout, và data augmentation.
* Tối ưu hóa hiệu suất: Tìm kiếm các phương pháp tối ưu hóa hiệu suất của mô hình, bao gồm việc sử dụng các kỹ thuật tối ưu hóa khác nhau và tinh chỉnh siêu tham số.
* Mở rộng ứng dụng: Áp dụng mô hình vào các bài toán thực tế khác nhau và nghiên cứu các phương pháp để tối ưu hóa và mở rộng ứng dụng của mô hình trong các lĩnh vực như y tế, ô tô tự lái, hoặc công nghệ thông tin.
* Bằng cách tiếp tục nghiên cứu và phát triển các phương pháp và ứng dụng của học máy và AI, chúng ta có thể đạt được những tiến bộ lớn đối với các vấn đề thực tế và tạo ra những giá trị mới cho xã hội.