TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN : XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH

ĐỀ TÀI SỐ 06: XÂY DỰNG HỆ THỐNG NHẬN DIỆN CHỮ SỐ VIẾT TAY

Giảng viên hướng dẫn: Lương Thị Hồng Lan

TT	Mã sinh viên	Sinh viên thực hiện	Lớp hành chính
1	20211127	Nguyễn Hữu Cường	DCCNTT12.10.4
2	20210933	Nguyễn Văn Mạnh	DCCNTT12.10.4
3	20210921	Phạm Hải Quốc	DCCNTT12.10.4
4	20210969	Bùi Việt Anh	DCCNTT12.10.4
5	20211016	Ngô Văn Thuyết	DCCNTT12.10.4

Bắc Ninh, năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN : XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH

ĐỀ TÀI SỐ 06: XÂY DỰNG HỆ THỐNG NHẬN DIỆN CHỮ SỐ VIẾT TAY

Giảng viên hướng dẫn: Lương Thị Hồng Lan

TT	Mã sinh viên	Sinh viên thực hiện	Lớp hành chính
1	20211127	Nguyễn Hữu Cường	DCCNTT12.10.4
2	20210933	Nguyễn Văn Mạnh	DCCNTT12.10.4
3	20210921	Phạm Hải Quốc	DCCNTT12.10.4
4	20210969	Bùi Việt Anh	DCCNTT12.10.4
5	20211016	Ngô Văn Thuyết	DCCNTT12.10.4

Bắc Ninh, năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA CÔNG NGHỆ THÔNG TIN

KỲ THI KẾT THÚC HỌC PHẦN
HỌC KỲ 1, NĂM HỌC 2024 – 2025

PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN
TÊN HỌC PHẦN :XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH
ĐỀ TÀI SỐ 06: XÂY DỰNG HỆ THỐNG NHẬN DIỆN CHỮ SỐ VIẾT TAY

Lớp Tín chỉ: DCCNTT12.10.4

Cán bộ chấm thi 1 **Cán bộ chấm thi 2**

Lương thị Hồng Lan

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Hữu Cường 20211127	Nguyễn Văn Mạnh 20210933	Phạm Hải Quốc 20210921	Bùi Việt Anh 20210969	Ngô Văn Thuyết 20211016
1	**Nội dung báo cáo trên Word đầy đủ**	**3.5**					
1.1	Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận).	0,5					
1.2	Nội dung phân tích rõ ràng, logic.	0,5					
1.3	Có dẫn chứng, số liệu minh họa đầy đủ.	0,5					
1.4	Ngôn ngữ và trình bày chuẩn, không lỗi chính tả.	0,5					
1.5	Có trích dẫn tài liệu tham khảo đúng quy cách.	0,5					
1.6	Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý).	0,5					
1.7	Tài liệu đầy đủ, bám sát yêu cầu của đề bài.	0,5					

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Hữu Cường	Nguyễn Văn Mạnh	Phạm Hải Quốc	Bùi Việt Anh	Ngô Văn Thuyết
			20211127	20210933	20210921	20210969	20211016
2	**Nội dung thuyết trình đầy đủ**	**1.0**					
2.1	Trình bày tự tin, phát âm rõ ràng, mạch lạc.	0,5					
2.2	Nội dung thuyết trình đúng trọng tâm, không lan man.	0,5					
3	**Slides báo cáo đầy đủ nội dung + Hỏi đáp**	**3.0**					
3.1	Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận).	0,5					
3.2	Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa).	0,5					
3.3	Nội dung trên slides ngắn gọn, dễ hiểu, súc tích.	0,5					
3.4	Nội dung slides phù hợp với nội dung báo cáo.	0,5					
3.5	Trả lời câu hỏi đầy đủ, chính xác.	0,5					
3.6	Trả lời câu hỏi tự tin, thuyết phục.	0,5					
4	**Code đầy đủ**	**2.5**					
1.1	Code được trình bày rõ ràng, có chú thích đầy đủ.	0,5					
1.2	Code chạy đúng, không lỗi.	0,5					
1.3	Code tối ưu, không dư thừa.	0,5					
1.4	Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài.	0,5					
1.5	Có tính sáng tạo hoặc cải thiện so với yêu cầu.	0,5					
TỔNG ĐIỂM BẰNG SỐ:		**10**					

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Hữu Cường	Nguyễn Văn Mạnh	Phạm Hải Quốc	Bùi Việt Anh	Ngô Văn Thuyết
			20211127	20210933	20210921	20210969	20211016
TỔNG ĐIỂM BẰNG CHỮ:		*Mười tròn*					

Mục lục

Danh mục viết tắt

STT	Chữ viết tắt	Giải thích
1	tv	Thư viện
2	cnn	Convolutional Neural Network
3	nn	Neural Network
4	OCR	Optical Character Recognition)

Danh sách bảng biểu hình ảnh

Phân công nhiệm vụ

STT	Tên thành viên	Vai trò	Nhiệm vụ	Thời gian hoàn thành dự kiến
1	Nguyễn Hữu Cường	Nhóm trưởng	Code	
2	Nguyễn Văn Mạnh	Thành viên	Word chương 1	
3	Phạm Hải Quốc	Thành viên	Word chương 2	
4	Bùi Việt Anh	Thành viên	Word chương 3, Slide báo cáo	
5	Ngô Văn Thuyết	Thành viên	Word chương 3, slide báo cáo	

Lời nói đầu

Nhận dạng chữ số viết tay là một trong những lĩnh vực quan trọng trong xử lý ảnh và trí tuệ nhân tạo, với ứng dụng rộng rãi trong nhiều lĩnh vực như nhận dạng văn bản, quản lý tài liệu, và tự động hóa các quy trình xử lý dữ liệu. Khả năng chuyển đổi chữ viết tay thành văn bản số không chỉ giúp tiết kiệm thời gian mà còn tăng độ chính xác và hiệu quả trong xử lý thông tin.

Đề tài "Nhận dạng chữ số viết tay trong xử lý ảnh" nhằm nghiên cứu và áp dụng các kỹ thuật xử lý ảnh, trích xuất đặc trưng và sử dụng các thuật toán học máy để phân loại và nhận diện chữ số. Trong quá trình thực hiện, đề tài tập trung vào các bước như tiền xử lý ảnh, phát hiện và tách chữ số, xây dựng mô hình học máy, và đánh giá hiệu quả hệ thống.

Với sự phát triển nhanh chóng của công nghệ và dữ liệu, nhận dạng chữ số viết tay không chỉ là một vấn đề kỹ thuật mà còn mở ra cơ hội cải tiến nhiều ứng dụng thực tế trong cuộc sống. Chúng tôi hy vọng rằng nghiên cứu này sẽ mang lại giá trị thiết thực và là nền tảng cho các nghiên cứu sâu hơn trong tương lai.

Chương 1 Cơ sở lý thuyết

1.1 Nhận dạng

1.1.1 Nhận dạng là gì?

Nhận dạng là một quá trình hoặc hệ thống nhằm xác định và phân loại một đối tượng, sự kiện, hoặc hiện tượng dựa trên các đặc điểm, dữ liệu thu thập được từ môi trường. Khái niệm này được ứng dụng rộng rãi trong nhiều lĩnh vực khoa học và công nghệ, như trí tuệ nhân tạo (AI), học máy (machine learning), xử lý ảnh, xử lý ngôn ngữ tự nhiên (NLP), và các hệ thống an ninh.

Nhận dạng có thể hiểu như việc hệ thống mô phỏng khả năng nhận thức của con người, giúp máy tính hoặc các thiết bị thông minh "hiểu" và "phân tích" các yếu tố như hình ảnh, âm thanh, văn bản, hoặc chuyển động.

Hình 1. Nhận dạng chữ viết

❖ Các loại nhận dạng chi tiết

- Nhận dạng hình ảnh: Phân tích và xác định nội dung trong ảnh hoặc video, ví dụ nhận dạng vật thể, khuôn mặt, hoặc cảnh quan.
- Nhận dạng giọng nói: Xác định nội dung hoặc nhận diện người nói từ tín hiệu âm thanh.
- Nhận dạng văn bản (Text Recognition): Tự động phát hiện và trích xuất thông tin từ văn bản, kể cả chữ viết tay hoặc tài liệu scan.

- Nhận dạng hành động và cử chỉ: Phân tích và nhận diện các chuyển động, hành động hoặc cử chỉ của con người từ video hoặc cảm biến.
- Nhận dạng sinh trắc học: Sử dụng các đặc điểm sinh học duy nhất của con người để nhận diện cá nhân, ví dụ vân tay, võng mạc, hoặc đặc điểm khuôn mặt.

1.1.2 Ứng dụng

❖ Nhận dạng khuôn mặt (Face Recognition)

- Ngân hàng: Xác thực khách hàng khi thực hiện giao dịch trực tuyến hoặc tại ATM.
- Lực lượng pháp luật: Tìm kiếm đối tượng truy nã trong đám đông, xác minh danh tính tại hiện trường.
- Sân bay: Kiểm soát an ninh, xác minh hành khách một cách nhanh chóng và chính xác.
- Quản lý truy cập: Kiểm soát ra vào các khu vực hạn chế, theo dõi sự hiện diện của nhân viên.
- Marketing: Phân tích cảm xúc khách hàng dựa trên biểu cảm khuôn mặt để cải thiện trải nghiệm mua sắm.

❖ Nhận diện biển số xe (ANPR)

- Thu phí tự động: Tính phí dựa trên biển số xe, loại bỏ các trạm thu phí truyền thống.
- Giám sát tốc độ: Phát hiện các phương tiện vượt quá tốc độ cho phép.
- Tìm kiếm xe mất cắp: Xây dựng cơ sở dữ liệu biển số xe để hỗ trợ công tác điều tra.
- Quản lý bãi đỗ xe: Tự động hóa quá trình vào/ra bãi đỗ xe, tính phí chính xác.

❖ Phân loại đối tượng trong ảnh (Object Detection)

- An ninh: Phát hiện các đối tượng lạ, hành vi đáng ngờ trong các khu vực công cộng.
- Tự lái xe: Nhận biết người đi bộ, xe đạp, các phương tiện khác, biển báo giao thông để đưa ra quyết định lái xe an toàn.
- Sản xuất: Kiểm soát chất lượng sản phẩm, phát hiện lỗi sản xuất.

❖ OCR (Optical Character Recognition)

- Lưu trữ dữ liệu: Số hóa tài liệu, tạo cơ sở dữ liệu văn bản dễ tìm kiếm.
- Dịch vụ khách hàng: Tự động hóa quá trình xử lý các đơn hàng, yêu cầu của khách hàng.

- Giáo dục: Tạo ra các công cụ học tập tương tác, hỗ trợ người khiếm thị.

❖ Ứng dụng trong y tế (Medical Imaging)

- Chẩn đoán hình ảnh: Hỗ trợ bác sĩ trong việc phát hiện các bệnh lý sớm, tăng độ chính xác của chẩn đoán.
- Phẫu thuật: Hỗ trợ phẫu thuật bằng cách cung cấp hình ảnh 3D chi tiết về cơ thể bệnh nhân.
- Nghiên cứu y học: Phân tích một lượng lớn dữ liệu hình ảnh để tìm ra các mối liên hệ mới, phát triển các phương pháp điều trị hiệu quả.

❖ Phân tích và nhận dạng cử chỉ (Gesture Recognition)

- Giao diện người máy: Điều khiển robot, máy tính bằng cử chỉ.
- Trò chơi điện tử: Tạo ra các trò chơi tương tác cao, trải nghiệm người dùng chân thực hơn.
- Hỗ trợ người khuyết tật: Điều khiển các thiết bị thông minh bằng cử chỉ.

❖ Nhận diện sản phẩm trong thương mại điện tử

- Tìm kiếm hình ảnh: Cho phép người dùng tìm kiếm sản phẩm bằng hình ảnh thay vì nhập từ khóa.
- So sánh giá: So sánh giá của cùng một sản phẩm trên các trang web khác nhau.
- Tư vấn mua sắm: Đề xuất các sản phẩm tương tự hoặc bổ sung dựa trên sản phẩm mà người dùng đang xem.

❖ Phát hiện và phân loại hình ảnh vệ tinh

- Giám sát môi trường: Theo dõi sự thay đổi của rừng, biển, đô thị.
- Quản lý thiên tai: Phát hiện các khu vực bị ảnh hưởng bởi thiên tai, hỗ trợ công tác cứu hộ.
- Quản lý đô thị: Lập bản đồ đô thị, theo dõi sự phát triển của các khu vực.

❖ Tiềm năng phát triển

- AI và Deep Learning: Sự phát triển của trí tuệ nhân tạo và học sâu giúp các hệ thống nhận dạng hình ảnh trở nên chính xác và thông minh hơn.
- Ứng dụng trong thực tế ảo (VR) và tăng cường thực tế (AR): Tạo ra các trải nghiệm tương tác sống động hơn.
- An ninh quốc gia: Phân tích hình ảnh vệ tinh để phát hiện các hoạt động bất hợp pháp, bảo vệ biên giới.

1.2 Các phương pháp sử dụng trong nhận dạng

1.2.1 Sử dụng đặc trưng biên

Sử dụng đặc trưng đường biên (Edge Features) là một trong những phương pháp quan trọng trong xử lý ảnh và nhận diện đối tượng. Đặc trưng đường biên tập trung vào việc trích xuất thông tin liên quan đến ranh giới giữa các vùng trong ảnh, giúp xác định hình dạng và cấu trúc của đối tượng.

Biên ảnh là những điểm mà tại đó hàm độ sáng của ảnh liên tục có bước nhảy hoặc biến thiên nhanh. Cơ sở toán học của việc phát hiện và tách biên là phép toán đạo hàm, phương pháp này còn được gọi là phương pháp phát hiện biên trực tiếp. Tập hợp các điểm biên tạo thành đường biên(edge) hay đường bao (boundary) của ảnh. Ví dụ trong một ảnh nhị phân một điểm có thể được gọi là biên nếu đó là điểm đen và có ít nhất một điểm trắng lân cận.

Đặc trưng đường biên chủ yếu được sử dụng trong các bài toán nhận diện đối tượng, phân đoạn ảnh và phân tích hình ảnh. Cụ thể:

- Nhận diện đối tượng: Đặc trưng đường biên giúp phát hiện các đối tượng bằng cách nhận diện ranh giới giữa chúng và nền.
- Phân đoạn ảnh: Dùng để tách các đối tượng ra khỏi nền dựa trên sự thay đổi cường độ sáng.
- Phân tích hình dạng: Phát hiện các hình dạng trong ảnh như hình tròn, hình vuông, giúp phân loại và nhận diện hình ảnh.
- Quản lý ảnh y tế: Dùng trong các ứng dụng như nhận diện tổn thương hoặc các khối u trên ảnh X-quang hoặc CT.

Ưu điểm:

- Nhận diện chính xác các ranh giới:
  - Đặc trưng đường biên rất hiệu quả trong việc xác định ranh giới giữa các đối tượng và nền trong ảnh, điều này cực kỳ quan trọng trong các bài toán phân đoạn ảnh và nhận diện đối tượng.
- Tăng cường tính chính xác trong nhận diện hình dạng:
  - Việc phát hiện đường biên giúp mô hình hiểu rõ cấu trúc và hình dạng của đối tượng, nâng cao khả năng nhận diện hình ảnh.
- Giảm dung lượng tính toán:

- Thay vì xử lý toàn bộ ảnh, việc sử dụng đặc trưng đường biên giúp giảm thiểu số lượng dữ liệu cần xử lý, từ đó giúp tăng tốc độ xử lý và tiết kiệm tài nguyên.

➢ Có thể kết hợp với các phương pháp khác:

- Đặc trưng đường biên có thể dễ dàng kết hợp với các đặc trưng khác như màu sắc và kết cấu để cải thiện hiệu quả nhận diện đối tượng trong ảnh.

➢ Hiệu quả trong phân tích các ảnh có độ tương phản cao:

- Đặc trưng đường biên hoạt động tốt khi ảnh có sự thay đổi mạnh về cường độ sáng giữa các đối tượng và nền.

Nhược điểm:

➢ Nhạy cảm với nhiễu:

- Đặc trưng đường biên có thể bị nhiễu bởi các yếu tố không mong muốn như hạt nhiễu trong ảnh, gây ra những đường biên giả hoặc sai lệch. Điều này cần phải xử lý qua các bước tiền xử lý, ví dụ như làm mịn ảnh.

➢ Không hiệu quả trong ảnh có độ tương phản thấp:

- Khi sự thay đổi giữa các đối tượng và nền không rõ ràng (ảnh có độ tương phản thấp), đặc trưng đường biên có thể không phát hiện được các đường biên rõ ràng.

➢ Khó phát hiện đối tượng nhỏ hoặc phức tạp:

- Đặc trưng đường biên khó nhận diện các đối tượng nhỏ hoặc các đối tượng có nhiều chi tiết phức tạp, như tóc hoặc các họa tiết nhỏ trong ảnh.

➢ Không chứa thông tin nội dung đối tượng:

- Đặc trưng đường biên chỉ cung cấp thông tin về ranh giới của đối tượng mà không mang lại thông tin về nội dung bên trong đối tượng, như màu sắc hoặc kết cấu.

➢ Phụ thuộc vào ngưỡng và tham số:

- Các thuật toán phát hiện đường biên như Canny hay Sobel phụ thuộc vào việc chọn đúng ngưỡng (threshold). Nếu ngưỡng không phù hợp, có thể dẫn đến việc phát hiện quá ít hoặc quá nhiều đường biên.

1.2.2 Học sâu

❖ Neural network là gì

Con chó có thể phân biệt được người thân trong gia đình và người lạ hay đứa trẻ có thể phân biệt được các con vật. Những việc tưởng chừng như rất đơn giản nhưng lại cực kì khó để thực hiện bằng máy tính. Vậy sự khác biệt nằm ở đâu? Câu trả lời nằm ở cấu trúc bộ não với lượng lớn các nơ-ron thần kinh liên kết với nhau.

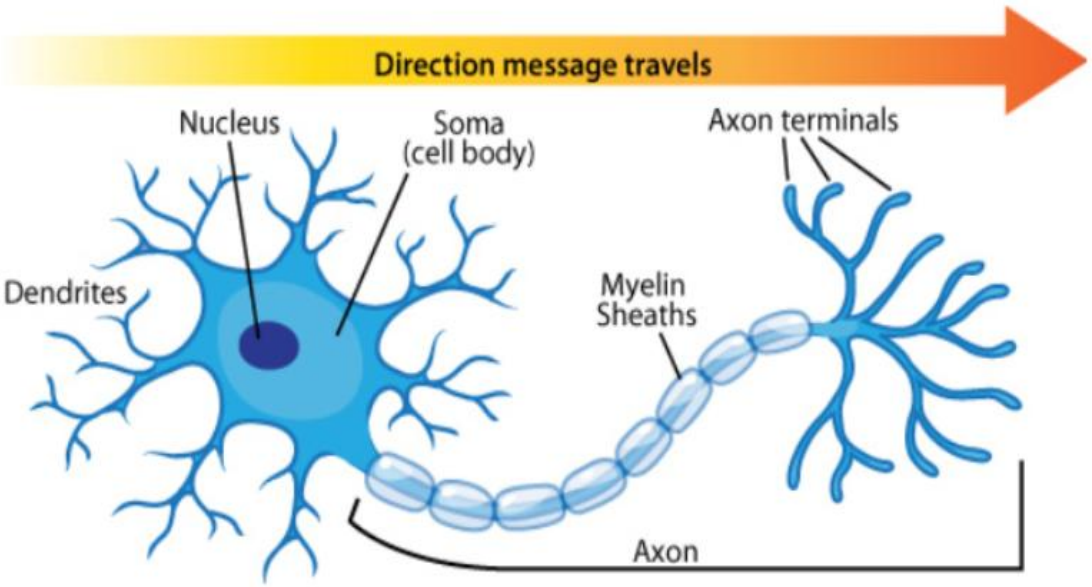
Neural là tính từ của neuron (nơ-ron), network chỉ cấu trúc, cách các nơ-ron đó liên kết với nhau, nên neural network (NN) là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ-ron trong hệ thần kinh.

Nơ-ron là đơn vị cơ bản cấu tạo hệ thống thần kinh và là thành phần quan trọng nhất của não. Đầu chúng ta gồm khoảng 10 triệu nơ-ron và mỗi nơ-ron lại liên kết với tầm 10.000 nơ-ron khác.

Ở mỗi nơ-ron có phần thân (soma) chứa nhân, các tín hiệu đầu vào qua sợi nhánh (dendrites) và các tín hiệu đầu ra qua sợi trục (axon) kết nối với các nơ-ron khác. Hiểu đơn giản mỗi nơ-ronnhận dữ liệu đầu vào qua sợi nhánh và truyền dữ liệu đầu ra qua sợi trục, đến các sợi nhánh của các nơ-ron khác.

Mỗi nơ-ron nhận xung điện từ các nơ-ron khác qua sợi nhánh. Nếu các xung điện này đủ lớn để kích hoạt nơ-ron, thì tín hiệu này đi qua sợi trục đến các sợi nhánh của các nơ-ron khác.

Mô hình neural network



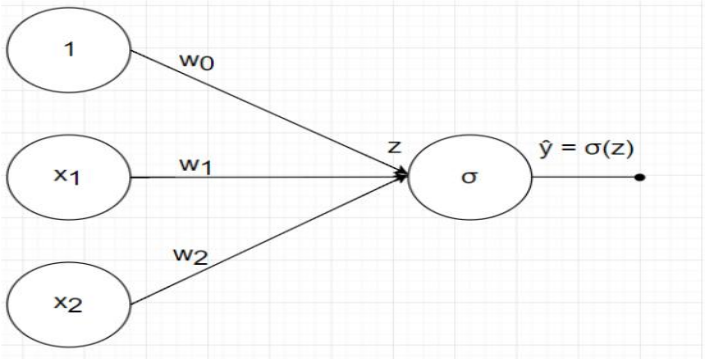
Hình 2. Mạng neuron

❖ Logistic regression

Logistic regression là mô hình neural network đơn giản nhất chỉ với input layer và output layer. Mô hình của logistic regression từ bài trước là: $\hat{y} = s(w_0 + w_1 * x_1 + w_2 * x_2)$. Có 2 bước:

• Tính tổng linear: $z = 1 * w_0 + x_1 * w_1 + x_2 * w_2$

• Áp dụng sigmoid function: $\hat{y} = \sigma(z)$



Hình 3. Mô hình Logistic regression

Hệ số w0 được gọi là bias. Để ý từ những bài trước đến giờ dữ liệu khi tính toán luôn được thêm 1 để tính hệ số bias w_0 . Tại sao lại cần hệ số bias? Quay lại với bài 1, phương trình đường thẳng sẽ thế nào nếu bỏ w_0, phương trình giờ có dạng: $y = w1 * x$, sẽ luôn đi qua gốc tọa độ và nó không tổng quát hóa phương trình đường thẳng nên có thể không tìm được phương trình mong muốn.

❖ Mô hình tổng quát

Layer đầu tiên là input layer, các layer ở giữa được gọi là hidden layer, layer cuối cùng được gọi là output layer. Các hình tròn được gọi là node. Mỗi mô hình luôn có 1 input layer, 1 output layer, có thể có hoặc không các hidden layer. Tổng số layer trong mô hình được quy ước là số layer - 1 (không tính input layer).

Ví dụ như ở hình trên có 1 input layer, 2 hidden layer và 1 output layer. Số lượng layer của mô hình là 3 layer.

Mỗi node trong hidden layer và output layer :

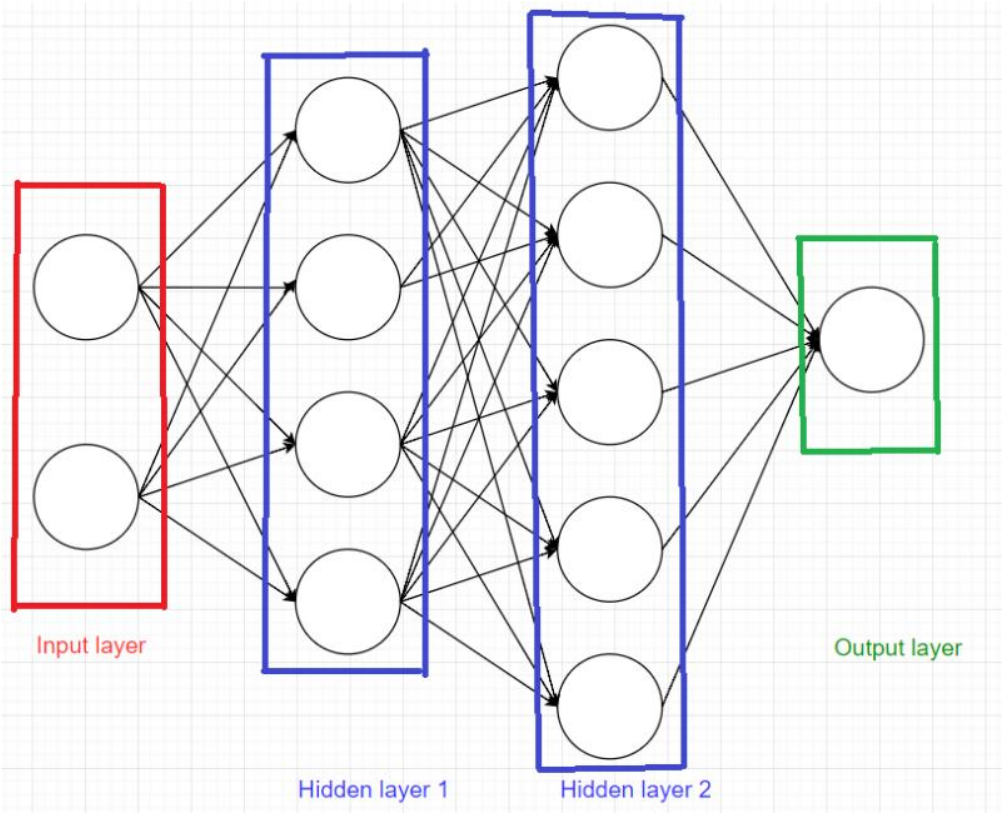
• Liên kết với tất cả các node ở layer trước đó với các hệ số w riêng.

• Mỗi node có 1 hệ số bias b riêng.

• Diễn ra 2 bước: tính tổng linear và áp dụng activation function.

Số node trong hidden layer thứ i là l(i).

Ma trận W(k) kích thước l(k-1) ∗ l(k) là ma trận hệ số giữa layer (k-1) và layer k, trong đó w(i j k) là hệ số kết nối từ node thứ i của layer k-1 đến node thứ j của layer k.

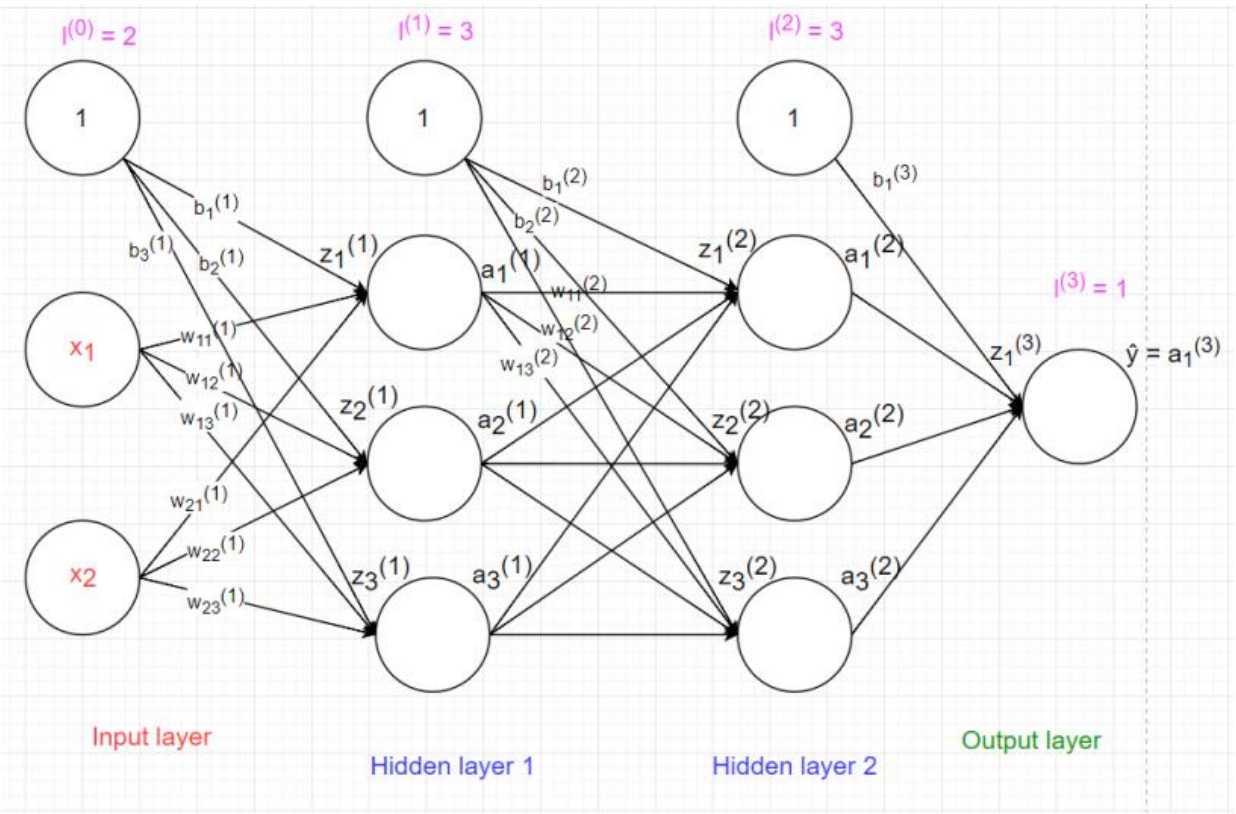


Hình 4.Mô hình tổng quát

Vector $b^{(k)}$kích thước $l^k * 1$ là hệ số bias của các node trong layer k, trong đó $b_i^{(k)}$ là bias của node thứ i trong layer k. Với node thứ i trong layer l có bias b(il)thực hiện 2 bước:

- Tính tổng linear: $z_i^{(l)} = \sum_{j=1}^{l^{(l-1)}} a_j^{(l-1)} * w_{ji}^{(l)} + b_i^{(l)}$, là tổng tất cả các node trong layer trước nhân với hệ số w tương ứng, rồi cộng với bias b.
- Áp dụng activation function: $a_i^{(l)} = \sigma(z_i^{(l)})$

Vector $z^{(k)}$ kích thước $l^{(k)} * 1$ là giá trị các node trong layer k sau bước tính tổng linear. Vector $z^{(k)}$ kích thước $l^{(k)} * 1$ là giá trị của các node trong layer k sau khi áp dụng hàm activation function.



Hình 5. Mô hình minh họa

Mô hình neural network trên gồm 3 layer. Input layer có 2 node ($l^{(0)} = 2$), hidden layer 1 có 3 node, hidden layer 2 có 3 node và output layer có 1 node. Do mỗi node trong hidden layer và output layer đều có bias nên trong input layer và hidden layer cần thêm node 1 để tính bias (nhưng không tính vào tổng số node layer có).

Tại node thứ 2 ở layer 1, ta có:

- $z_2^{(1)} = x_1 * w_{12}^{(1)} + x_2 * w_{22}^{(1)} + b_2^{(1)}$
- $a_2^{(1)} = \sigma(z_2^{(1)})$

Hay ở node thứ 3 layer 2, ta có:

- $z_3^{(2)} = a_1^{(1)} * w_{13}^{(2)} + a_2^{(1)} * w_{23}^{(2)} + a_3^{(1)} * w_{33}^{(2)} + b_3^{(2)}$
- $a_3^{(2)} = \sigma(z_3^{(2)})$

Ưu điểm:

➢ Khả năng học từ dữ liệu phức tạp:

- Mạng nơ-ron có khả năng học các mối quan hệ phức tạp và phi tuyến giữa các đặc trưng trong dữ liệu, điều mà các thuật toán học máy truyền thống như hồi quy tuyến tính không thể làm được. Nhờ vào lớp ẩn (hidden layers), nó có thể phát hiện ra những mẫu và đặc trưng không rõ ràng trong dữ liệu

➢ Độ chính xác cao:

- Khi được huấn luyện đúng cách với dữ liệu đủ lớn, mạng nơ-ron có thể đạt được độ chính xác rất cao, đặc biệt trong các bài toán như nhận diện ảnh, xử lý ngôn ngữ tự nhiên, và các dự đoán phức tạp

➢ Khả năng tự động phát hiện đặc trưng:

- Mạng nơ-ron có khả năng tự động học các đặc trưng từ dữ liệu mà không cần phải thiết kế các đặc trưng thủ công, điều này giúp tiết kiệm thời gian và công sức trong việc chuẩn bị dữ liệu

➢ Ứng dụng đa dạng:

- Mạng nơ-ron có thể được áp dụng trong nhiều lĩnh vực khác nhau như nhận diện hình ảnh, phân loại văn bản, dự đoán thị trường tài chính, y tế, và tự lái xe

Nhược điểm:

➢ Cần dữ liệu lớn để huấn luyện:

- Mạng nơ-ron thường yêu cầu một lượng dữ liệu lớn để huấn luyện và đạt được kết quả tốt. Khi thiếu dữ liệu, mô hình có thể gặp vấn đề overfitting (học quá kỹ các đặc trưng của dữ liệu huấn luyện và không tổng quát tốt trên dữ liệu mới)

➢ Khó giải thích và kiểm soát:

- Mạng nơ-ron hoạt động như một hộp đen, nghĩa là khó để hiểu được quá trình mà mô hình sử dụng để đưa ra quyết định. Điều này đặc biệt quan trọng trong các ứng dụng cần giải thích mô hình, chẳng hạn như trong y tế hoặc tài chính .

➢ Yêu cầu tính toán và tài nguyên cao:

- Để huấn luyện mạng nơ-ron, đặc biệt là với các mô hình sâu (deep learning), yêu cầu tài nguyên tính toán lớn như GPU và thời gian huấn luyện kéo dài

➢ Quá trình huấn luyện có thể lâu dài:

- Việc huấn luyện một mạng nơ-ron có thể mất nhiều thời gian, đặc biệt khi dữ liệu huấn luyện lớn và mạng phức tạp, điều này có thể gây khó khăn trong việc triển khai nhanh chóng

➢ Dễ bị overfitting nếu không có đủ dữ liệu hoặc kỹ thuật điều chỉnh:

- Nếu không áp dụng các kỹ thuật điều chỉnh như Dropout, Regularization, hoặc Data Augmentation, mạng nơ-ron có thể học quá kỹ dữ liệu huấn luyện và không tổng quát tốt với dữ liệu mới

❖ Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) là một trong những loại mạng nơ-ron phổ biến và mạnh mẽ nhất trong lĩnh vực thị giác máy tính và học sâu. CNN đặc biệt hiệu quả trong việc xử lý và phân tích dữ liệu có cấu trúc lưới như ảnh, video, và dữ liệu không gian 2D. Điều này giúp CNN trở thành một công cụ cực kỳ mạnh mẽ cho các bài toán như nhận diện đối tượng, phân loại ảnh, phân đoạn ảnh, và nhiều ứng dụng khác trong thị giác máy tính.

Convolutional Neural Network (CNN) bắt nguồn từ các nghiên cứu về mạng neuron và thị giác sinh học. Năm 1980, Kunihiko Fukushima phát minh Neocognitron, mô hình đầu tiên sử dụng các lớp tích chập và pooling, mô phỏng cơ chế thị giác tự nhiên. Sau đó, vào năm 1989, Yann LeCun và đồng nghiệp phát triển LeNet-5, CNN hiện đại đầu tiên, ứng dụng trong nhận diện chữ viết tay, đánh dấu bước ngoặt quan trọng trong ứng dụng thực tế của mạng neuron.

CNN thực sự bùng nổ vào năm 2012 với sự ra đời của AlexNet, kiến trúc mạng sâu sử dụng GPU để huấn luyện và đạt hiệu suất vượt trội trong cuộc thi ImageNet. Từ đó, hàng loạt cải tiến như VGGNet, ResNet, và EfficientNet ra đời, giúp CNN được áp dụng rộng rãi trong nhiều lĩnh vực như nhận diện hình ảnh, xử lý video, y học, và robot. CNN đã trở thành một trụ cột quan trọng của trí tuệ nhân tạo nhờ sự kết hợp giữa lý thuyết, dữ liệu lớn và tiến bộ phần cứng.

CNN là một mô hình học sâu được thiết kế để tự động và hiệu quả học từ dữ liệu hình ảnh, trích xuất các đặc trưng không gian quan trọng mà không cần sự can thiệp thủ công. Với khả năng nhận diện các mẫu cơ bản như cạnh, góc, và các đối tượng phức tạp, CNN có thể sử dụng các đặc trưng này để đưa ra dự đoán chính xác.

Từ "Convolutional" trong CNN xuất phát từ thao tác tích chập (convolution) mà mô hình này sử dụng để xử lý dữ liệu hình ảnh. Convolution giúp mạng học và trích xuất các đặc trưng của hình ảnh, chẳng hạn như cạnh và hình dạng, thông qua các bộ lọc (filters) hoặc kernel. Đây là bước đầu tiên giúp mạng "nhìn" và hiểu được cấu trúc không gian của hình ảnh.

Khi bạn đưa một ảnh vào CNN, mỗi bộ lọc (kernel) sẽ "di chuyển" qua ảnh và áp dụng phép toán tích chập để tạo ra các đặc trưng. Sau đó, các đặc trưng này sẽ được đưa vào các lớp tiếp theo của mạng để học thêm các mẫu phức tạp hơn. Quá trình này

giúp CNN phát hiện các đặc trưng từ đơn giản đến phức tạp, từ cạnh cơ bản đến đối tượng phức tạp như con người, xe hơi, động vật, v.v.

Convolution neural network gồm những lớp cơ bản sau:

❖ Convolutional layer

Đây chính là lớp đóng vai trò mấu chốt của CNN, khi layer này đảm nhiệm việc thực hiện mọi tính toán. Stride, padding, filter map, feature map là những yếu tố quan trọng nhất của convolutional layer.

- Cơ chế của CNN là tạo ra các filter áp dụng vào từng vùng hình ảnh. Các filter map này được gọi là ma trận 3 chiều, bên trong chứa các parameter dưới dạng những con số.
- Stride là sự dịch chuyển filter map theo pixel dựa trên giá trị từ trái sang phải.
- Padding: Là các giá trị 0 được thêm cùng lớp input.
- Feature map: Sau mỗi lần quét, một quá trình tính toán sẽ được thực hiện. Feature map sẽ thể hiện kết quả sau mỗi lần filter map quét qua input.

Relu layer

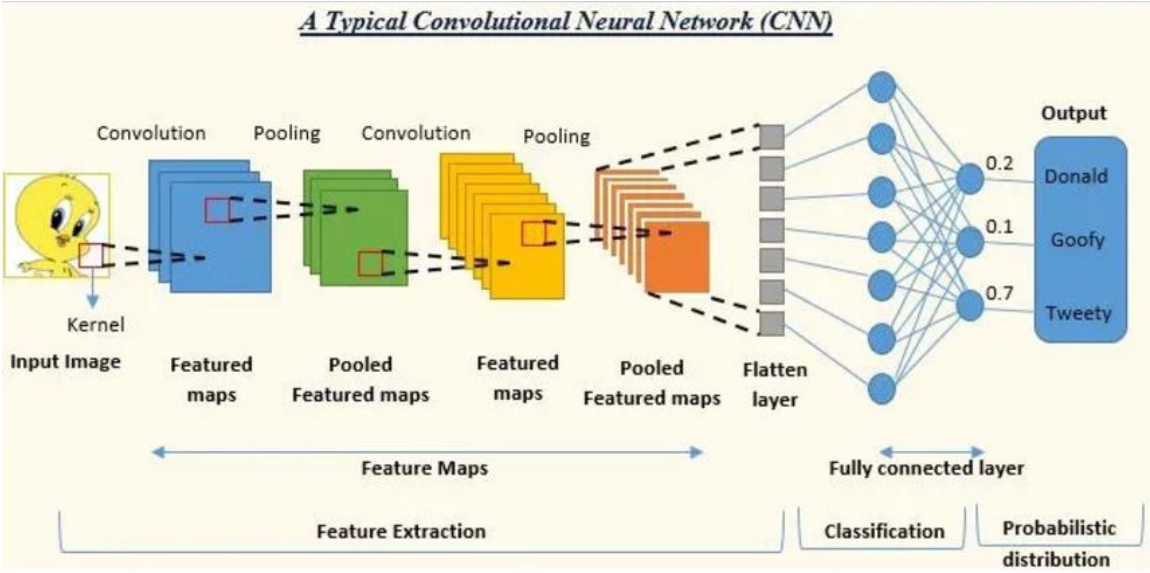
Còn có tên gọi khác là activation function, đây là một hàm được kích hoạt trong neural network. Có tác dụng mô phỏng các neuron có tỷ lệ truyền xung qua axon. Trong activation function chúng còn có hàm nghĩa là: Relu, Tanh, Sigmoid, Maxout, Leaky,… Relu layer được ứng dụng phổ biến trong việc huấn luyện nơ-ron do sở hữu nhiều ưu điểm tiên tiến.

❖ Pooling layer

Khi nhận phải đầu vào quá lớn, các lớp pooling layer sẽ được xếp giữa những lớp Convolutional layer nhằm mục đích giảm parameter. Pooling layer được chia thành 2 loại phổ biến là max pooling và average.

❖ Fully connected layer

Khi 2 lớp convolutional layer và pooling layer nhận được ảnh truyền, lớp này sẽ có nhiệm vụ xuất kết quả. Khi ta nhận được kết quả là model đọc được thông tin ảnh, ta cần phải tạo sự liên kết để cho ra nhiều output hơn. Đây chính là lúc các lập trình viên sử dụng fully connected layer. Hơn nữa, nếu fully connected layer có dữ liệu về hình ảnh thì chúng sẽ chuyển thành mục chưa được phân chia chất lượng.



Hình 6. Mô hình CNN

Suốt quá trình huấn luyện, CNN sẽ tự động học hỏi các giá trị thông qua lớp filter với "mẫu" là cách thức người dùng thực hiện. Điều này khá giống với cách bộ não con người nhận diện những vật thể trong tự nhiên.

Một cấu trúc cơ bản nhất của CNN sẽ bao gồm 3 phần chủ yếu, đó là:

- Local receptive field (trường cục bộ): Nhiệm vụ của trường cục bộ là phân tách và lọc dữ liệu cũng như thông tin ảnh, sau đó chọn ra các vùng ảnh có giá trị sử dụng cao nhất.
- Shared weights and bias (trọng số chia sẻ): Trong mạng CNN, thành phần này có tác dụng giảm thiểu tối đa lượng tham số có tác dụng lớn. Trong mỗi convolution sẽ chứa nhiều feature map khác nhau, mỗi feature lại có khả năng giúp nhận diện một số feature trong ảnh.
- Pooling layer (lớp tổng hợp): Pooling layer là lớp cuối cùng, với khả năng đơn giản hóa thông tin đầu ra. Khi đã hoàn tất tính toán và quét qua các lớp, pooling layer sẽ được tạo ra nhằm mục đích lược bớt các thông tin không cần thiết và tối ưu đầu ra. Điều này giúp người dùng nhận được kết quả ưng ý và đúng với yêu cầu hay mong muốn.

❖ Mạng CNN có gì đặc biệt?

Tự động trích xuất đặc trưng

Trái ngược với các phương pháp học máy truyền thống, nơi người ta phải trích xuất đặc trưng thủ công từ dữ liệu (ví dụ: thông qua histogram, SIFT, hoặc HOG), CNN tự động học và trích xuất các đặc trưng từ dữ liệu đầu vào. Điều này không chỉ giúp giảm bớt công việc thủ công mà còn mang lại khả năng nhận diện chính xác hơn khi mạng có thể tự phát hiện các đặc trưng quan trọng mà không bị ảnh hưởng bởi các quyết định của người lập trình.

Khả năng nhận diện đối tượng mạnh mẽ

CNN có khả năng nhận diện các đối tượng trong ảnh một cách mạnh mẽ và hiệu quả. Mạng này có thể phát hiện các đối tượng ngay cả khi chúng thay đổi về kích thước, vị trí hoặc góc độ trong ảnh, nhờ vào các lớp pooling và convolution giúp mạng học được các đặc trưng bất biến đối với các biến đổi này.

Khả năng xử lý ảnh có kích thước lớn

CNN có thể xử lý và học từ các ảnh có độ phân giải lớn mà không gặp phải vấn đề về việc trích xuất đặc trưng thủ công, làm cho nó đặc biệt hữu ích cho các bài toán yêu cầu phân tích hình ảnh quy mô lớn (ví dụ: nhận diện khuôn mặt hoặc phân loại ảnh với hàng triệu ảnh).

Các ứng dụng phổ biến của CNN

CNN đã đạt được nhiều thành tựu nổi bật trong các bài toán liên quan đến thị giác máy tính. Một số ứng dụng điển hình bao gồm:

- Nhận diện và phân loại đối tượng: CNN có thể phân loại ảnh thành các nhóm hoặc phân biệt các đối tượng trong ảnh (ví dụ: nhận diện mèo, chó, xe hơi).
- Nhận diện khuôn mặt: CNN được sử dụng trong nhiều ứng dụng nhận diện khuôn mặt, từ bảo mật cho đến ứng dụng phân tích cảm xúc.
- Phân đoạn ảnh: CNN có thể chia ảnh thành các vùng có ý nghĩa, ví dụ trong phân đoạn y tế, nơi mỗi vùng có thể đại diện cho một bộ phận cơ thể cụ thể.
- Nhận diện chữ viết tay: CNN cũng được áp dụng để nhận diện chữ viết tay, ví dụ trong việc quét các tài liệu.

1.3 Ngôn ngữ lập trình và các thư viện sử dụng

1.3.1 Ngôn ngữ lập trình

❖ Python

Python là một ngôn ngữ lập trình bậc cao, thông dịch, và đa năng, được thiết kế để dễ học và dễ sử dụng. Nó được tạo ra bởi Guido van Rossum vào năm 1991 và ngày nay là một trong những ngôn ngữ lập trình phổ biến nhất thế giới.

Đặc điểm chính của Python

- Cú pháp dễ đọc và học: Cú pháp của Python đơn giản, gần gũi với ngôn ngữ tự nhiên, giúp lập trình viên dễ dàng nắm bắt ngay cả khi mới bắt đầu.
- Đa nền tảng: Python hoạt động trên hầu hết các hệ điều hành phổ biến như Windows, macOS, Linux, và có thể được sử dụng trên các thiết bị IoT.
- Hướng đối tượng: Python hỗ trợ lập trình hướng đối tượng (OOP), nhưng cũng rất linh hoạt với các phong cách lập trình thủ tục hoặc hàm.
- Thư viện phong phú: Python cung cấp hàng nghìn thư viện tích hợp sẵn và có thể mở rộng với các thư viện bên thứ ba như:
  - NumPy, Pandas, Matplotlib: Xử lý dữ liệu.
  - TensorFlow, PyTorch, scikit-learn: Học máy và AI.
  - Django, Flask: Phát triển web.
  - OpenCV, Pillow: Xử lý ảnh.
- Cộng đồng lớn và hỗ trợ mạnh mẽ: Python có một cộng đồng người dùng và nhà phát triển lớn, tài liệu phong phú, giúp việc học và giải quyết vấn đề trở nên dễ dàng.
- Đa năng: Python có thể được sử dụng trong nhiều lĩnh vực, từ phát triển phần mềm, phân tích dữ liệu, học máy, đến lập trình web, nhúng và tự động hóa.

1.3.2 Các thư viện sử dụng

❖ OpenCv (Open Source Computer Vision Library)

OpenCV là một thư viện mã nguồn mở nổi tiếng dành cho xử lý ảnh và thị giác máy tính (Computer Vision). Được phát triển bởi Intel vào năm 2000 và hiện nay được hỗ trợ bởi một cộng đồng lớn, OpenCV là một công cụ mạnh mẽ cho các ứng dụng xử lý ảnh, phát hiện đối tượng, nhận dạng khuôn mặt, và nhiều hơn nữa.

Đặc điểm nổi bật của OpenCV

- Xử lý ảnh nhanh chóng và hiệu quả:Hỗ trợ các thao tác xử lý ảnh cơ bản và phức tạp, từ làm mờ ảnh, chuyển đổi màu, đến phát hiện cạnh và đối tượng.

- Tương thích đa nền tảng:Hoạt động trên nhiều hệ điều hành như Windows, macOS, Linux, và Android/iOS.
- Hỗ trợ nhiều ngôn ngữ:OpenCV có thể được sử dụng với Python, C++, Java, và Matlab.
- Tích hợp mạnh mẽ với thư viện AI:OpenCV hỗ trợ tích hợp với các framework học máy và học sâu như TensorFlow, PyTorch, ONNX.
- Hỗ trợ video và xử lý thời gian thực:Xử lý ảnh từ camera và video trực tiếp, phù hợp với các ứng dụng giám sát hoặc phân tích hành vi.

Ứng dụng của OpenCV

- Nhận diện khuôn mặt: Sử dụng phương pháp Haar Cascades hoặc HOG + SVM để phát hiện và nhận diện khuôn mặt trong ảnh hoặc video.
- Phát hiện và theo dõi đối tượng: Xác định các đối tượng chuyển động hoặc tĩnh từ camera, video giám sát.
- Phân loại và nhận diện đối tượng: Kết hợp với các mô hình học sâu để phân loại và phát hiện đối tượng trong ảnh/video.
- Xử lý ảnh y tế: Phân tích hình ảnh từ X-quang, MRI để phát hiện bệnh.
- Nhận diện ký tự (OCR): Kết hợp với Tesseract để nhận diện chữ viết trong ảnh.
- Xử lý ảnh nghệ thuật và hiệu ứng:Áp dụng các bộ lọc, tạo hiệu ứng như tranh sơn dầu, vẽ phác thảo.
- Thị giác máy tính trong robot: Sử dụng OpenCV trong các hệ thống tự động như xe tự hành, robot công nghiệp.

Ưu điểm:

- Thư viện mã nguồn mở và miễn phí.
- Hỗ trợ xử lý thời gian thực.
- Cộng đồng lớn và nhiều tài liệu học tập.

Nhược điểm:

- Một số tính năng cần tích hợp thêm thư viện bên ngoài.
- Đôi khi cần kiến thức cơ bản về xử lý ảnh để sử dụng hiệu quả.

❖ NumPy(Numerical Python)

NumPy là một thư viện mạnh mẽ trong Python, được thiết kế để làm việc với mảng dữ liệu lớn (arrays) và thực hiện các tính toán số học hiệu quả. Đây là công cụ cơ bản cho các lĩnh vực như khoa học dữ liệu, học máy, và xử lý tín hiệu, thường được sử dụng kết hợp với các thư viện khác như Pandas, Matplotlib, và TensorFlow.

Đặc điểm nổi bật của NumPy

- Mảng đa chiều (ndarray): NumPy cung cấp đối tượng chính là ndarray, hỗ trợ lưu trữ và xử lý dữ liệu nhiều chiều hiệu quả hơn so với danh sách (list) trong Python.
- Tính toán nhanh chóng: Các phép toán được thực hiện trực tiếp trên mảng NumPy được tối ưu hóa, nhanh hơn so với sử dụng vòng lặp Python thông thường.
- Các hàm toán học tích hợp: NumPy cung cấp nhiều hàm toán học như cộng, trừ, nhân, chia, căn bậc hai, logarit, lượng giác, v.v.
- Hỗ trợ đại số tuyến tính: Bao gồm các phép tính như nhân ma trận, tìm định thức, tính eigenvalues, và giải hệ phương trình.
- Tương thích với dữ liệu từ bên ngoài: NumPy có thể đọc và ghi dữ liệu từ file CSV, txt, hoặc thậm chí từ các định dạng nhị phân.
- Tích hợp tốt với các thư viện khác: NumPy là nền tảng của nhiều thư viện Python phổ biến như Pandas, Scikit-learn, và Matplotlib.

Ứng dụng của NumPy

- Khoa học dữ liệu: Làm việc với dữ liệu lớn, xử lý dữ liệu nhanh chóng trước khi phân tích.
- Học máy: Là nền tảng cho các thư viện học máy như TensorFlow, Scikit-learn.
- Đại số tuyến tính: Giải quyết các bài toán liên quan đến ma trận và vector.
- Xử lý tín hiệu: Áp dụng các phép toán và hàm số học để phân tích tín hiệu.

Chương 2 Xây dựng hệ thống

2.1 Bài toán

Bài toán nhận dạng chữ số viết tay là một trong những bài toán kinh điển trong lĩnh vực xử lý ảnh và học máy. Mục tiêu của bài toán là xây dựng một hệ thống có khả năng nhận dạng chính xác các chữ số viết tay (0-9) từ hình ảnh đầu vào.

Cụ thể, bài toán bao gồm các bước sau:

❖ Đầu vào:

- Một hình ảnh chứa chữ số viết tay (có thể là đơn chữ số hoặc nhiều chữ số).
- Hình ảnh có thể được chụp từ giấy viết tay, bảng trắng, hoặc dữ liệu số hóa.

❖ Xử lý ảnh:

- Tiền xử lý ảnh để cải thiện chất lượng, bao gồm:
- Chuyển ảnh sang thang độ xám.
- Lọc nhiễu và làm mịn.
- Chuẩn hóa kích thước ảnh để đồng nhất dữ liệu.

❖ Trích xuất đặc trưng: Phân tích các đặc trưng quan trọng của chữ số như: hình dạng, đường viền, điểm ảnh đậm nhạt, hoặc sử dụng các phương pháp như PCA (Principal Component Analysis).

❖ Mô hình phân loại: Sử dụng thuật toán học máy (như KNN, SVM, hoặc mạng nơ-ron nhân tạo) để xây dựng mô hình nhận dạng.

❖ Đầu ra: Hệ thống trả về nhãn chữ số (0-9) tương ứng với hình ảnh đầu vào.

❖ Yêu cầu bài toán

- Độ chính xác cao: Hệ thống phải nhận dạng đúng đa số các chữ số viết tay, kể cả khi chúng có kiểu chữ hoặc kích thước khác nhau.
- Xử lý nhanh: Đảm bảo hệ thống nhận diện trong thời gian ngắn, phù hợp với các ứng dụng thời gian thực.
- Tính tổng quát: Khả năng nhận dạng được nhiều kiểu chữ số khác nhau, kể cả khi bị nghiêng, méo, hoặc thiếu nét.

❖ Ứng dụng thực tế

- Đọc và xử lý văn bản viết tay.
- Nhận dạng mã bưu điện, số tài liệu, hoặc biển số xe.
- Sử dụng trong các hệ thống tự động hóa, quản lý hồ sơ hoặc chấm bài thi.

2.2 Xây dựng hệ thống

2.2.1 Tiền xử lý dữ liệu

❖ Chuẩn hóa dữ liệu

- Kỹ thuật: Chuyển dữ liệu từ dạng nguyên (uint8, giá trị từ 0-255) về dạng float32 và chuẩn hóa dữ liệu trong khoảng [0, 1] bằng cách chia mỗi điểm ảnh cho 255.
- Ý nghĩa:
  - Giúp giảm độ phức tạp tính toán và cải thiện tốc độ hội tụ của mô hình.
  - Dữ liệu chuẩn hóa giúp tránh vấn đề về độ lớn giá trị khác nhau, làm giảm nguy cơ mô hình bị lệch.

❖ Tăng cường dữ liệu (Data Augmentation)

- Kỹ thuật: Tạo ra các biến thể của dữ liệu huấn luyện bằng cách sử dụng ImageDataGenerator, bao gồm:
  - Xoay ảnh (rotation_range): Giúp mô hình học cách nhận dạng chữ số ở các góc độ khác nhau.
  - Dịch chuyển (width_shift_range, height_shift_range): Mô phỏng dữ liệu bị lệch vị trí trong thực tế.
  - Phóng to/thu nhỏ (zoom_range): Tạo các biến thể khác nhau về kích thước.
- Ý nghĩa:
  - Giảm nguy cơ overfitting bằng cách tăng độ đa dạng dữ liệu huấn luyện.
  - Cải thiện khả năng khái quát hóa của mô hình khi gặp dữ liệu mới.

2.2.2 Xây dựng mô hình học sâu

❖ Tăng cường dữ liệu (Data Augmentation)

- Kỹ thuật: Tạo ra các biến thể của dữ liệu huấn luyện bằng cách sử dụng ImageDataGenerator, bao gồm:
  - Xoay ảnh (rotation_range): Giúp mô hình học cách nhận dạng chữ số ở các góc độ khác nhau.
  - Dịch chuyển (width_shift_range, height_shift_range): Mô phỏng dữ liệu bị lệch vị trí trong thực tế.
  - Phóng to/thu nhỏ (zoom_range): Tạo các biến thể khác nhau về kích thước.
- Ý nghĩa:

- Giảm nguy cơ overfitting bằng cách tăng độ đa dạng dữ liệu huấn luyện.
- Cải thiện khả năng khái quát hóa của mô hình khi gặp dữ liệu mới.

❖ Tăng cường dữ liệu (Data Augmentation)

- Kỹ thuật: Tạo ra các biến thể của dữ liệu huấn luyện bằng cách sử dụng ImageDataGenerator, bao gồm:
  - Xoay ảnh (rotation_range): Giúp mô hình học cách nhận dạng chữ số ở các góc độ khác nhau.
  - Dịch chuyển (width_shift_range, height_shift_range): Mô phỏng dữ liệu bị lệch vị trí trong thực tế.
  - Phóng to/thu nhỏ (zoom_range): Tạo các biến thể khác nhau về kích thước.
- Ý nghĩa:
  - Giảm nguy cơ overfitting bằng cách tăng độ đa dạng dữ liệu huấn luyện.
  - Cải thiện khả năng khái quát hóa của mô hình khi gặp dữ liệu mới.

❖ Tăng cường dữ liệu (Data Augmentation)

- Kỹ thuật: Tạo ra các biến thể của dữ liệu huấn luyện bằng cách sử dụng ImageDataGenerator, bao gồm:
  - Xoay ảnh (rotation_range): Giúp mô hình học cách nhận dạng chữ số ở các góc độ khác nhau.
  - Dịch chuyển (width_shift_range, height_shift_range): Mô phỏng dữ liệu bị lệch vị trí trong thực tế.
  - Phóng to/thu nhỏ (zoom_range): Tạo các biến thể khác nhau về kích thước.
- Ý nghĩa:
  - Giảm nguy cơ overfitting bằng cách tăng độ đa dạng dữ liệu huấn luyện.
  - Cải thiện khả năng khái quát hóa của mô hình khi gặp dữ liệu mới.

❖ Tăng cường dữ liệu (Data Augmentation)

- Kỹ thuật: Tạo ra các biến thể của dữ liệu huấn luyện bằng cách sử dụng ImageDataGenerator, bao gồm:
  - Xoay ảnh (rotation_range): Giúp mô hình học cách nhận dạng chữ số ở các góc độ khác nhau.
  - Dịch chuyển (width_shift_range, height_shift_range): Mô phỏng dữ liệu bị lệch vị trí trong thực tế.
  - Phóng to/thu nhỏ (zoom_range): Tạo các biến thể khác nhau về kích thước.

- Ý nghĩa:
  - Giảm nguy cơ overfitting bằng cách tăng độ đa dạng dữ liệu huấn luyện.
  - Cải thiện khả năng khái quát hóa của mô hình khi gặp dữ liệu mới.

❖ Tăng cường dữ liệu (Data Augmentation)

- Kỹ thuật: Tạo ra các biến thể của dữ liệu huấn luyện bằng cách sử dụng ImageDataGenerator, bao gồm:
  - Xoay ảnh (rotation_range): Giúp mô hình học cách nhận dạng chữ số ở các góc độ khác nhau.
  - Dịch chuyển (width_shift_range, height_shift_range): Mô phỏng dữ liệu bị lệch vị trí trong thực tế.
  - Phóng to/thu nhỏ (zoom_range): Tạo các biến thể khác nhau về kích thước.
- Ý nghĩa:
  - Giảm nguy cơ overfitting bằng cách tăng độ đa dạng dữ liệu huấn luyện.
  - Cải thiện khả năng khái quát hóa của mô hình khi gặp dữ liệu mới.

❖ Tăng cường dữ liệu (Data Augmentation)

- Kỹ thuật: Tạo ra các biến thể của dữ liệu huấn luyện bằng cách sử dụng ImageDataGenerator, bao gồm:
  - Xoay ảnh (rotation_range): Giúp mô hình học cách nhận dạng chữ số ở các góc độ khác nhau.
  - Dịch chuyển (width_shift_range, height_shift_range): Mô phỏng dữ liệu bị lệch vị trí trong thực tế.
  - Phóng to/thu nhỏ (zoom_range): Tạo các biến thể khác nhau về kích thước.
- Ý nghĩa:
  - Giảm nguy cơ overfitting bằng cách tăng độ đa dạng dữ liệu huấn luyện.
  - Cải thiện khả năng khái quát hóa của mô hình khi gặp dữ liệu mới.

Chương 3 Kết quả thực nghiệm

3.1 Dữ liệu

Bộ dữ liệu

Các bộ dữ liệu phổ biến cho bài toán nhận dạng chữ viết tay:

MNIST: Gồm 70.000 hình ảnh (28×28 pixel) của các chữ số viết tay từ 0 đến 9.

EMNIST: Mở rộng từ MNIST, bao gồm các ký tự chữ cái (in hoa, in thường) và chữ số.

IAM Dataset: Bộ dữ liệu gồm các dòng văn bản viết tay, được sử dụng cho nhận dạng từ hoặc câu.

CIFAR-10: Có thể được điều chỉnh để nhận dạng chữ viết tay khi cần đa dạng hóa bài toán.

Tiền xử lý dữ liệu

Chuyển đổi ảnh về thang xám (Grayscale): Giảm kích thước và loại bỏ thông tin màu không cần thiết.

Chuẩn hóa ảnh: Đưa giá trị pixel về khoảng [0, 1] để tăng tốc độ hội tụ của mô hình.

Tăng cường dữ liệu (Data Augmentation):

Lật ngang, xoay ảnh để mô hình có khả năng tổng quát hóa tốt hơn.

Thêm nhiễu Gaussian hoặc điều chỉnh độ sáng để mô phỏng các điều kiện viết tay khác nhau.

Chuẩn hóa kích thước ảnh: Đưa tất cả ảnh về cùng kích thước, ví dụ: 28×28 pixel hoặc 32×32 pixel.

Chia tập dữ liệu:

Training set: 70-80%.

Validation set: 10-15%.

Test set: 10-15%.

3.2 Độ đo đánh giá

Đánh giá độ chính xác phát hiện đối tượng (Object Detection Metrics)

Dù nhận dạng chữ viết tay là bài toán phân loại, nhưng bạn vẫn có thể áp dụng các chỉ số Precision, Recall và F1-score để đánh giá kết quả dự đoán chữ cái.

Precision (P)

Định nghĩa: Đo lường độ chính xác của các dự đoán về chữ viết tay.

Công thức:

$$\text{Precision} = \frac{\text{Số ký tự dự đoán đúng}}{\text{Số ký tự được dự đoán là đúng (đúng + sai)}}$$

True Positives (TP): Số ký tự được mô hình nhận diện chính xác.

False Positives (FP): Số ký tự bị mô hình dự đoán sai nhưng thực tế không tồn tại.

Recall (R)

Định nghĩa: Đo lường khả năng mà mô hình nhận diện tất cả các ký tự thực sự có trong ảnh.

Công thức:

$$\text{Recall} = \frac{\text{Số ký tự dự đoán đúng}}{\text{Số ký tự thực tế có trong ảnh (đúng + sai)}}$$

False Negatives (FN): Số ký tự thực sự có nhưng mô hình không nhận diện được.

F1-score

Định nghĩa: Là thước đo cân bằng giữa Precision và Recall, giúp đánh giá hiệu suất của mô hình trong cả hai khía cạnh (đúng và đầy đủ).

Công thức:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Đánh giá tốc độ xử lý (Processing Speed Metrics)

FPS (Frames Per Second)

Định nghĩa: FPS thể hiện khả năng của hệ thống trong việc nhận diện các chữ viết tay trong các ảnh hoặc video thời gian thực.

Công thức:

$$\text{FPS} = \frac{\text{Số ảnh đã xử lý}}{\text{Thời gian xử lý}}$$

Đo tốc độ xử lý trong việc nhận diện chữ viết tay qua mỗi khung hình.

Độ trễ (Latency)

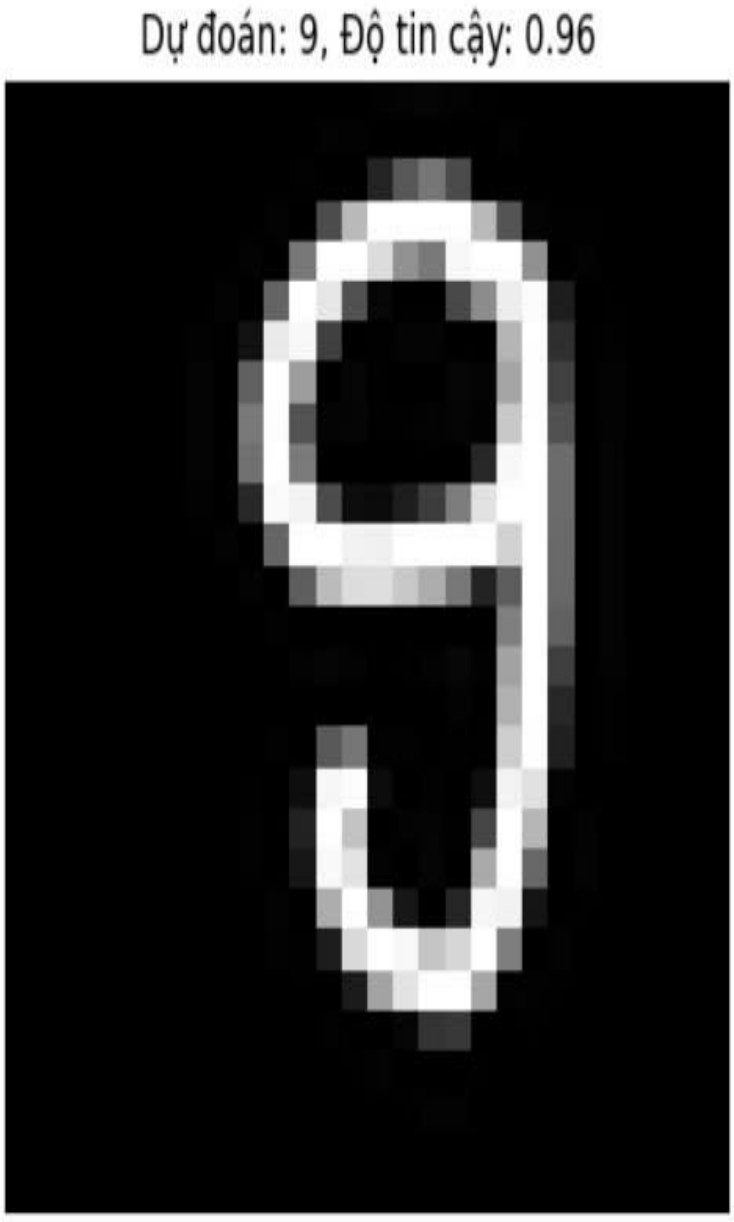
Định nghĩa: Đo thời gian mà hệ thống mất từ lúc nhận được ảnh chữ viết tay đến khi đưa ra kết quả nhận diện.

Công thức:

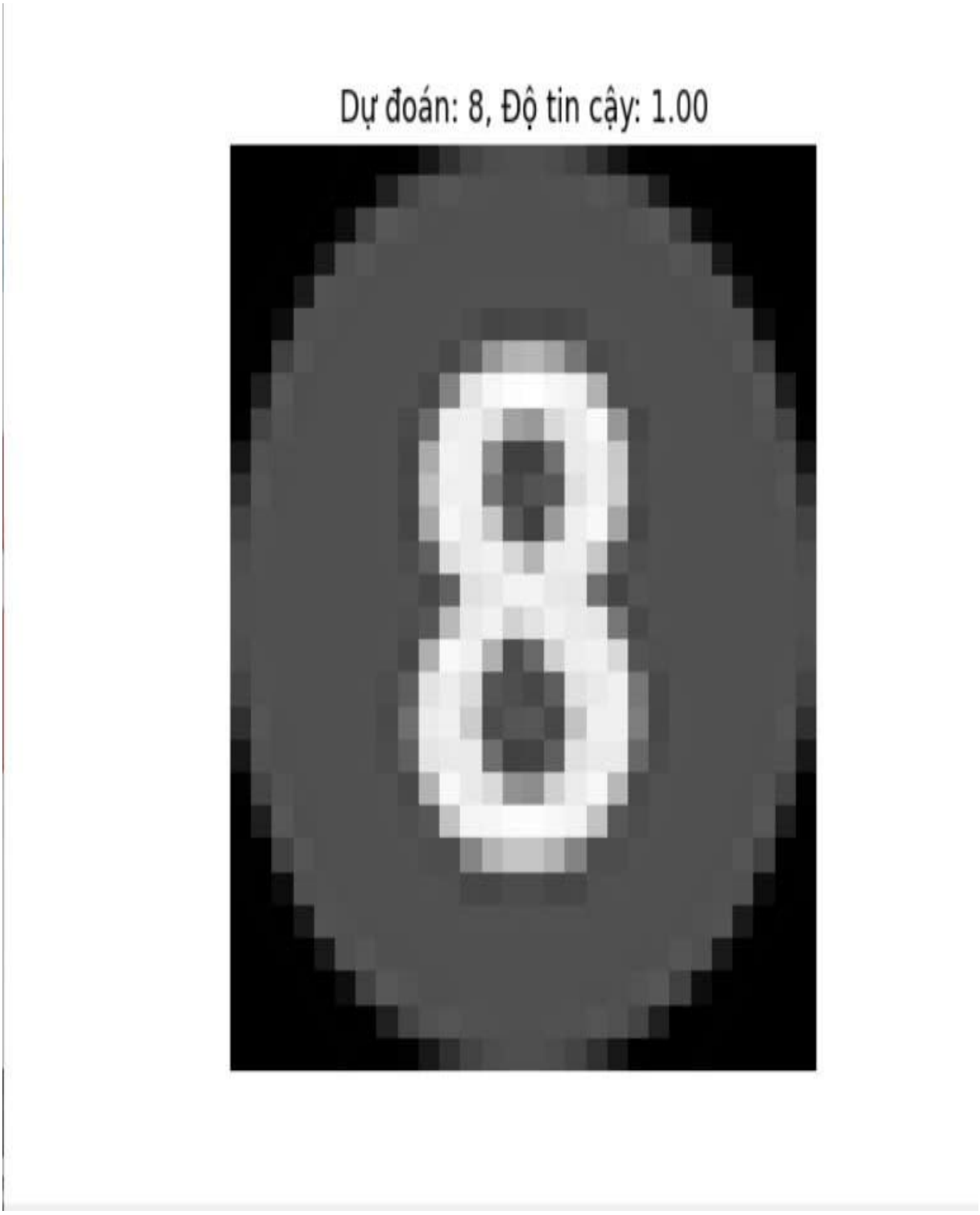
$$\text{Latency} = \text{Thời gian xử lý chữ viết tay (giây)} - \text{Thời gian nhận ảnh đầu vào}$$

3.3 Kết quả thực nghiệm

Hình 7. Dự đoán số 3



Hình 8.Dự đoán số 9



Hình 9. Dự đoán số 9

Hình 10. Dự đoán số 2

Kết luận

Bài toán nhận dạng chữ số viết tay là một ứng dụng điển hình của học sâu, mang lại những kết quả ấn tượng nhờ vào khả năng xử lý và trích xuất đặc trưng tự động từ dữ liệu hình ảnh. Qua việc xây dựng và huấn luyện mô hình CNN trên tập dữ liệu MNIST, hệ thống đã đạt được độ chính xác cao cùng khả năng dự đoán chính xác trên cả dữ liệu vẽ tay lẫn ảnh bên ngoài.

Kỹ thuật tiền xử lý dữ liệu như chuẩn hóa, tăng cường dữ liệu và xử lý ảnh đầu vào đã đóng vai trò quan trọng trong việc chuẩn bị dữ liệu đầu vào đồng nhất, tăng cường tính đa dạng và giảm lỗi hệ thống. Kiến trúc CNN với các lớp Convolutional, MaxPooling, Dropout và Batch Normalization đã giúp mô hình học và khái quát hóa tốt hơn. Đồng thời, việc tích hợp giao diện tương tác với Tkinter giúp ứng dụng trở nên thân thiện và dễ sử dụng.

Tuy nhiên, hệ thống vẫn có tiềm năng cải thiện:

- Mở rộng dữ liệu huấn luyện: Sử dụng các tập dữ liệu lớn và đa dạng hơn để tăng khả năng khái quát hóa.
- Tối ưu hiệu năng: Sử dụng mô hình nhẹ hơn hoặc triển khai trên GPU để cải thiện tốc độ dự đoán.
- Tích hợp thực tế: Triển khai ứng dụng trên các nền tảng như web hoặc di động để ứng dụng vào các lĩnh vực như giáo dục hoặc kiểm tra tự động.

Tài liệu tham khảo

[1] Nguyễn Thanh Tuấn - Deep learning cơ bản - (2019)

[2] Valliappa Lakshmanan, Sara Robinson, Michael Munn - Machine Learning Design Patterns - (5/10/2020) - O'Reilly Media

[3] Nguyễn Đình Cường - Image processing Lecture - (2/2020)

[4] Slide bài giảng của giảng viên Lương Thị Hồng Lan -(20/11/2024)

[5] Website: https://cuuduongthancong.com/s/xu-ly-anh#google_vignette - (3/12/2024)

: https://sachbaovn.vn/doc-truc-tuyen/sach/Giao-trinh-thi-giac-may-tinh-va-ung-dung-MUIwQzRBMjc - (4/12/2024 - 20h40)

: https://users.soict.hust.edu.vn/ductq/XLA%20Lecture.pdf - (5/12/2024)

[6} OpenAI: chatgpt.com -(20/11/2024)