

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**XÂY DỰNG APP ANDROID QUẢN LÝ
ĐIỂM THI SINH VIÊN**

Giảng viên hướng dẫn	:	ThS. PHAN THANH HY
Sinh viên thực hiện	:	NGUYỄN HOÀNG NAM
Mã số sinh viên	:	N17DCCN102
Lớp	:	D17CQCP02-N
Khóa	:	2017 – 2022
Hệ	:	ĐẠI HỌC CHÍNH QUY

TP. HCM, tháng 12/2021

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**XÂY DỰNG APP ANDROID QUẢN LÝ
ĐIỂM THI SINH VIÊN**

Giảng viên hướng dẫn	:	ThS. PHAN THANH HY
Sinh viên thực hiện	:	NGUYỄN HOÀNG NAM
Mã số sinh viên	:	N17DCCN102
Lớp	:	D17CQCP02-N
Khóa	:	2017 – 2022
Hệ	:	ĐẠI HỌC CHÍNH QUY

TP. HCM, tháng 12/2021

LỜI CẢM ƠN

Trước tiên, em xin gửi lời tri ân sâu sắc đến các thầy cô trường Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM đã tận tình dẫn dắt và truyền đạt cho em rất nhiều kiến thức quý báu trong các học kỳ vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến Ths Phan Thanh Hy. Thầy đã hướng dẫn tận tình, truyền đạt kiến thức, chỉ bảo cho em trong suốt thời gian học tập và thực hiện đề tài thực tập, đề tài tốt nghiệp. Trong quá trình làm đề tài thực sự có rất nhiều khó khăn đối với em, thầy đã luôn nhiệt hướng dẫn, động viên em. Kính chúc thầy và gia đình nhiều sức khỏe và thành công trong cuộc sống.

Con xin gửi lòng biết ơn và sự kính trọng đến cha mẹ, gia đình đã luôn động viên, tạo điều kiện tốt nhất cho con học tập. Và xin chân thành cảm ơn tất cả các bạn đã luôn sát cánh, giúp đỡ, động viên mình trong những thời điểm khó khăn, tiếp thêm động lực và ý chí giúp vượt qua khó khăn trong suốt quá trình học tập tại trường.

Tuy nhiên, vì thời gian và kiến thức còn hạn chế em đã cố gắng rất nhiều để hoàn thành đề tài này, nhưng trong quá trình làm bài em không tránh khỏi những thiếu sót, kính mong được lời nhận xét và góp ý của thầy cô.

Kính chúc tất cả Thầy cô, và các Anh/Chị lời chúc sức khỏe và thành công trong công việc cũng như trong cuộc sống.

Một lần nữa, em xin chân thành cảm ơn !

Sinh viên thực hiện

NGUYỄN HOÀNG NAM

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Giới thiệu chung.....	1
1.2 Tìm hiểu “Xây dựng app android quản lý điểm thi sinh viên”	1
1.3 Công cụ sử dụng.....	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	3
2.1 Sơ lược về công cụ.....	3
2.1.1 Giới thiệu về Jupyter Notebook.....	3
2.1.2 Giới thiệu Android Studio	3
2.1.3 Giới thiệu SQLiteDatabase BroswerPortable	4
2.2 Phân tích và thiết kế cơ sở dữ liệu	5
2.2.1 Thiết kế mô hình thực thể.....	5
2.2.2 Thiết kế dữ liệu	6
2.3 Cơ sở lý thuyết	9
2.3.1 Một số khái niệm liên quan	9
2.3.2 AI (Trí tuệ nhân tạo)	10
2.3.3 Machine learning.	11
2.3.4 Deep Learning.....	12
2.3.5 Mạng nơ ron.....	13
2.3.6 Mạng Nơ ron tích chập	14
CHƯƠNG 3. KỸ THUẬT NHẬN DẠNG KÝ TỰ	25
3.1 Nhận dạng ký tự sử dụng mạng Nơ ron lan truyền ngược.....	25
3.2 Nhận dạng ký tự sử dụng công nghệ OCR	26
3.3 Nhận dạng ký tự sử dụng công nghệ ICR	27
3.4 Công nghệ nhận dạng đánh dấu OMR	28
CHƯƠNG 4. DEMO CHƯƠNG TRÌNH.....	30
4.1 Mô tả	30
4.2 Xây dựng Model	30
4.3 Demo chương trình “Quản lý điểm thi sinh viên”	34
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	42
PHỤ LỤC	43
TÀI LIỆU THAM KHẢO.....	44

DANH MỤC HÌNH

Hình 2.1 Giao diện Jupyter Notebook.....	3
Hình 2.2 Giao diện Android Studio.....	4
Hình 2.3 Giao diện SQLite Database Browser Portable	4
Hình 2.4 Mô hình ERD	5
Hình 2.5 Các điểm gần mặt phân cách nhất của 2 classes	11
Hình 2.6 Tập giá trị của dữ liệu.....	12
Hình 2.7 Cấu tạo 1 Nơ ron	13
Hình 2.8 Mô hình mạng CNN	15
Hình 2.9 Minh họa tích chập	15
Hình 2.10 Ảnh mờ sau khi chập	16
Hình 2.11 Hình ảnh phát hiện biên sau khi chập.....	18
Hình 2.12 Mô hình mạng CNN Lenet 5	19
Hình 2.13 Đổi kích thước ảnh đầu vào lớp tích chập đầu tiên	20
Hình 2.14 Các loại lớp lấy mẫu con	21
Hình 2.15 Feature Map.....	21
Hình 2.16 Kích thước ảnh khi đi qua lớp tích chập thứ hai	22
Hình 2.17 Ảnh sau khi thông qua lớp gộp thứ hai	22
Hình 2.18 Lớp kết nối đầy đủ với 84 đơn vị	23
Hình 2.19 Ảnh đầu ra	24
Hình 2.20 Bảng tóm tắt mô hình Lenet 5	24

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
CNN	Convolutional neural network
kNN	k-nearest neighbors
SVM	Support vector machine
GPU	Máy chủ
RGB	Ảnh màu Red Green Blue

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Giới thiệu chung

Trong những năm gần đây, chúng ta đã được chứng kiến nhiều thành tựu vượt bậc trong lĩnh vực xử lý ảnh (image processing). Các hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động. Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) là một trong những mô hình học sâu (Deep Learning) tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

Các bài toán nhận dạng đang được ứng dụng trong thực tế hiện nay tập trung vào nhận dạng mẫu, nhận dạng tiếng nói và nhận dạng ký tự, chữ viết... Nhận dạng ký tự là bài toán được quan tâm rất nhiều vì nó là một trong các yêu cầu trong nhiều ứng dụng thực tế. Các ứng dụng của nhận dạng ký tự đã và đang được ứng dụng vào đời sống như phục vụ cho công việc tự động hóa đọc tài liệu, tăng tốc độ và hiệu quả nhập thông tin vào máy tính. Nhận dạng ký tự có thể phục vụ cho các ứng dụng đọc và xử lý các chứng từ, hóa đơn, phiếu ghi, scan văn bản...

Các hệ thống nhận dạng trước đây như (LeCun et al., 1998) [1], (Simard et al., 2003) [2], (Kégl & Busa Fekete, 2009) [3] đều sử dụng các đặc trưng cơ bản từ ảnh ký tự như đường biên, cạnh, độ dày, giá trị mức xám, haar-like, với các xử lý đặc thù như lấy mẫu, dao động các điểm ảnh, biến đổi ảnh, thêm dữ liệu ảo. Sau đó hệ thống nhận dạng huấn luyện các mô hình học tự động như k láng giềng (kNN) [4], mạng Nơ ron, máy vector hỗ trợ (SVM) [5], boosting. Các hệ thống này đã cơ bản đáp ứng yêu cầu của người sử dụng.

Bằng việc thực hiện đề tài “Xây dựng App android quản lý điểm thi sinh viên”, em muốn tìm hiểu rõ về mô hình học sâu CNN trong việc nhận dạng ký tự số và áp dụng mô hình này vào việc nhập điểm thi cho sinh viên tại học viện.

1.2 Tìm hiểu “Xây dựng app android quản lý điểm thi sinh viên”

a, Lý thuyết

- Tìm hiểu về nhận dạng số.
- Các phương pháp và giải thuật về nhận dạng.
- Tìm hiểu các mô hình deep learning.

- Tìm hiểu mạng neuron.
- Tìm hiểu mô hình học sâu Lenet-5.
- Tìm hiểu về model: Jupyter NoteBook.
- Tìm hiểu công nghệ xây dựng app Android.

b, Thực hành

- Phân tích và thiết kế cơ sở dữ liệu.
- Xây dựng app trên hệ điều hành Android thực hiện:
 - Quản lý danh sách sinh viên.
 - Quản lý điểm thi sinh viên ở học viện.
 - Thực hiện thêm danh sách sinh viên, danh sách môn học.
 - Thực hiện việc áp dụng mô hình học sâu để nhận dạng ký tự số viết tay từ đó hỗ trợ phòng khảo thí ở Học viện áp dụng đọc điểm từ bảng điểm viết tay của các giảng viên.
 - Hỗ trợ báo cáo, thống kê điểm thi sinh viên (các sinh viên rớt môn, qua môn,...).

c, Mục tiêu và phạm vi đề tài

- Mục tiêu: nghiên cứu hệ thống nhận dạng ký tự số sử dụng mạng CNN (mô hình học sâu lenet-5).
- Phạm vi: Các chữ số tự nhiên từ 0 đến 9 và ứng dụng vào việc quản lý điểm thi của sinh viên học viện.

1.3 Công cụ sử dụng

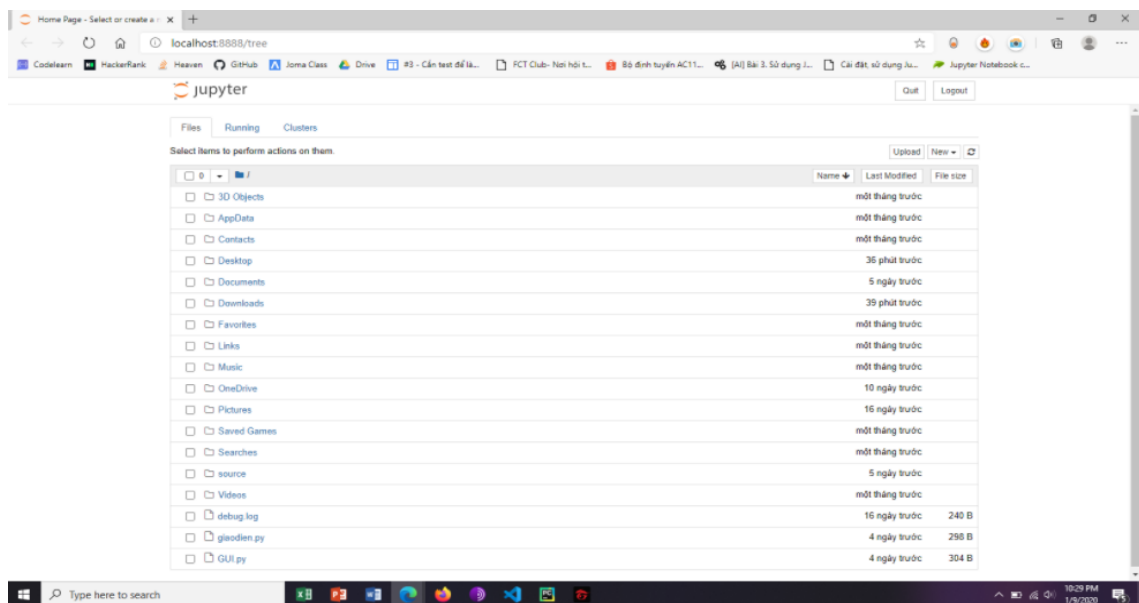
- Xây dựng Model: Jupyter Notebook.
- Xây dựng App: Android Studio, SQLiteDatabaseBrowserPortable.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Sơ lược về công cụ

2.1.1 Giới thiệu về Jupyter Notebook

Jupyter Notebook là công cụ cho phép bạn đưa cả code Python và các thành phần văn bản phức tạp như hình ảnh, công thức, video, biểu thức... vào trong cùng một file giúp cho việc trình bày trở lên dễ hiểu, giống như một file trình chiếu nhưng lại có thể thực hiện chạy code tương tác trên đó. Các file notebook này có thể được chia sẻ với mọi người và có thể thực hiện lại các công đoạn một cách nhanh chóng và chính xác như những gì tác giả của file này tạo ra.



Hình 2.1 Giao diện Jupyter Notebook

2.1.2 Giới thiệu Android Studio

Android Studio là một phần mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập emulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

Việc xây dựng một ứng dụng mobile (hoặc tablet) bao gồm rất nhiều các công đoạn khác nhau. Đầu tiên chúng ta sẽ viết code ứng dụng sử dụng máy tính cá nhân

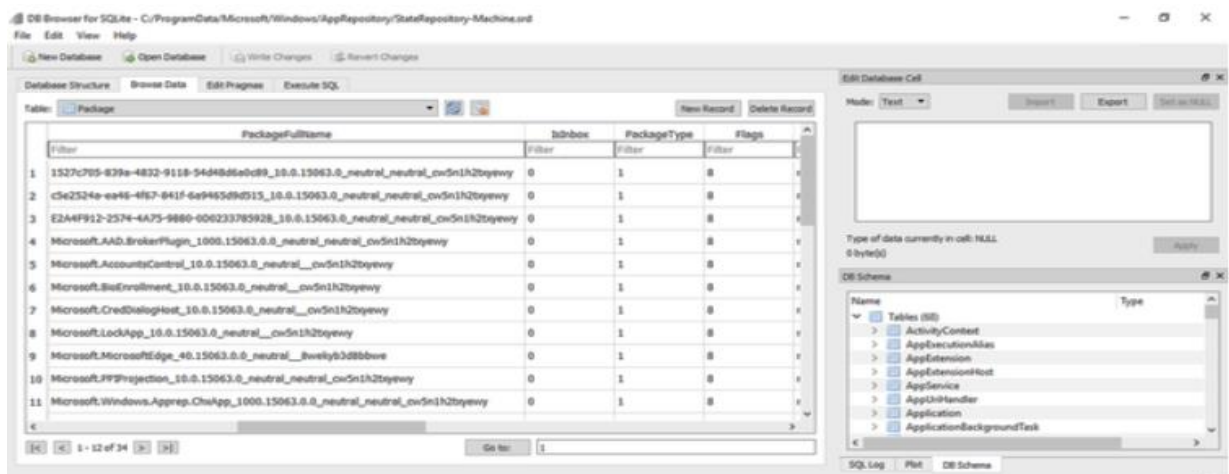
hoặc laptop. Sau đó chúng ta cần build ứng dụng để tạo file cài đặt. Sau khi build xong thì chúng ta cần copy file cài đặt này vào thiết bị mobile (hoặc table) để tiến hành cài đặt ứng dụng và chạy kiểm thử (testing). Bạn thử tưởng tượng nếu với mỗi lần viết một dòng code bạn lại phải build ứng dụng, cài đặt trên điện thoại hoặc tablet và sau đó chạy thử thì sẽ vô cùng tốn thời gian và công sức. Android Studio được phát triển để giải quyết các vấn đề này. Với Android Studio tất cả các công đoạn trên được thực hiện trên cùng một máy tính và các quy trình được tinh gọn tới mức tối giản nhất.



Hình 2.2 Giao diện Android Studio

2.1.3 Giới thiệu SQLiteDatabase BrowserPortable

SQLite Database Browser Portable là một công cụ mã nguồn mở, được thiết kế trực quan và chỉnh sửa các tệp cơ sở dữ liệu tương thích với SQLite. Đây là công cụ dành cho người dùng và nhà phát triển muốn tạo ra cơ sở dữ liệu, tìm kiếm và chỉnh sửa dữ liệu. Nó sử dụng một giao diện giống như bảng tính quen thuộc, và bạn không cần phải học những lệnh SQL phức tạp.



Hình 2.3 Giao diện SQLite Database Browser Portable

2.2 Phân tích và thiết kế cơ sở dữ liệu

2.2.1 Thiết kế mô hình thực thể

a, Xác định các thực thể

Khoa (MaKhoa, TenKhoa).

Lop (MaLop, TenLop, MaKhoa).

Mon (MaMon, TenMon).

SinhVien (MaSV, HoTen, NgaySinh, MaLop).

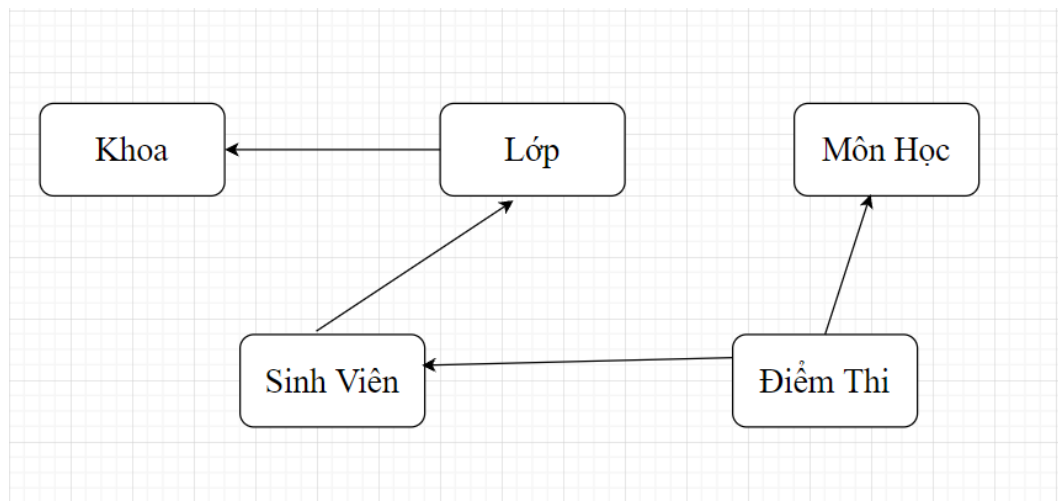
DiemThi (MaSV, MaMon, LanThi, Diem).

b, Xác định mối quan hệ giữa các thực thể

Giữa 2 thực thể Khoa và Lop có quan hệ 1-n. Một khoa có nhiều lớp và một lớp thì chỉ thuộc một khoa.

Giữa 2 thực thể Lop và SinhVien có quan hệ 1-n. Một lớp có nhiều sinh viên và một sinh viên thì chỉ thuộc một lớp.

c, Sơ đồ ERD



Hình 2.4 Mô hình ERD

2.2.2 Thiết kế dữ liệu

a, Từ điển dữ liệu

➤ Bảng Khoa

ST T	Tên trường	Kiểu dữ liệu	Độ dài	Diễn giải	Khóa chính	Khóa ngoại	Not null
1	MaKhoa	VARCHAR	10	Mã khoa	X		X
2	TenKhoa	VARCHAR	200	Tên khoa			X

➤ Bảng Lớp

ST T	Tên trường	Kiểu dữ liệu	Độ dài	Diễn giải	Khóa chính	Khóa ngoại	Not null
1	MaLop	VARCHA R	15	Mã lớp	X		X
2	TenLop	VARCHA R	200	Tên lớp			X
3	MaKhoa	VARCHA R	10	Mã khoa		X	X

➤ Bảng Môn Học

ST T	Tên trường	Kiểu dữ liệu	Độ dài	Diễn giải	Khóa chính	Khóa ngoại	Not null
1	MaMon	VARCHAR	20	Mã môn	X		X
2	TenMon	VARCHAR	200	Tên môn			X

➤ Bảng Sinh Viên

ST T	Tên trường	Kiểu dữ liệu	Độ dài	Diễn giải	Khóa chính	Khóa ngoại	Not null
1	MaSV	VARCHAR	20	Mã sinh viên	X		X
2	HoTen	VARCHAR	200	Tên sinh viên			X
3	NgaySin h	VARCHAR	200	Ngày sinh			X
4	MaLop	VARCHAR	15	Mã lớp		X	X

➤ Bảng Điểm Thi

ST T	Tên trường	Kiểu dữ liệu	Độ dài	Diễn giải	Khóa chính	Khóa ngoại	Not null
1	MaSV	VARCHAR	10	Mã sinh viên	X	X	X
2	MaMon	VARCHAR	20	Mã môn	X	X	X
3	LanThi	VARCHAR	5	Lần thi	X		X
4	Diem	INTEGER		Điểm			X

b, Ràng buộc toàn vẹn

➤ Ràng buộc 1:

Bối cảnh: Khoa.

Mô tả: MaKhoa là duy nhất.

Nội dung: $\forall k1, k2 \in \text{Khoa } k1 \neq k2 \Rightarrow k1.\text{MaKhoa} \neq k2.\text{MaKhoa}$.

Quan hệ	Thêm	Xóa	Sửa
Khoa	+	+	-

Tầm ảnh hưởng:

➤ Ràng buộc 2:

Bối cảnh: Lớp.

Mô tả: MaLop là duy nhất.

Nội dung: $\forall l1, l2 \in \text{Lop } l1 \neq l2 \Rightarrow l1.\text{MaLop} \neq l2.\text{MaLop}$.

Tầm ảnh hưởng:

Quan hệ	Thêm	Xóa	Sửa
Lớp	+	+	-

➤ Ràng buộc 3:

Bối cảnh: MonHoc

Mô tả: MaMon là duy nhất.

Nội dung: $\forall m1, m2 \in \text{MonHoc } m1 \neq m2 \Rightarrow m1.\text{MaMon} \neq m2.\text{MaMon}$.

Tầm ảnh hưởng:

Quan hệ	Thêm	Xóa	Sửa
MonHoc	+	+	-

➤ Ràng buộc 4:

Bối cảnh: SinhVien

Mô tả: MaSV là duy nhất.

Nội dung: $\forall m1, m2 \in \text{SinhVien } m1 \neq m2 \Rightarrow m1.\text{MaSV} \neq m2.\text{MaSV}$.

Tầm ảnh hưởng:

Quan hệ	Thêm	Xóa	Sửa
SinhVien	+	+	-

➤ Ràng buộc 5:

Bối cảnh: DiemThi.

Mô tả: MaSV, MaMon, LanThi là duy nhất.

Nội dung: $\forall m1, m2 \in \text{DiemThi } m1 \neq m2 \Rightarrow m1.(\text{MaSV}, \text{MaMon}, \text{LanThi}) \neq m2.(\text{MaSV}, \text{MaMon}, \text{LanThi})$

Tầm ảnh hưởng:

Quan hệ	Thêm	Sửa
DiemThi	+	-

2.3 Cơ sở lý thuyết**2.3.1 Một số khái niệm liên quan**

Ảnh và điểm ảnh: Gốc của ảnh (ảnh tự nhiên) là ảnh liên tục về không gian và độ sáng. Để xử lý bằng máy tính (số), ảnh cần phải được số hóa. Số hóa ảnh là sự biến đổi gần đúng một ảnh liên tục thành một tập điểm phù hợp với ảnh thật về vị trí (không gian) và độ sáng (mức xám). Khoảng cách giữa các điểm ảnh đó được thiết lập sao cho mắt người không phân biệt được ranh giới giữa chúng. Mỗi một điểm như vậy gọi là điểm ảnh (pixel).

Độ phân giải của ảnh: là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị. Theo định nghĩa, khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân bố, đó chính là độ phân giải và được phân bố theo trục x và y trong không gian hai chiều.

Mức xám của ảnh: là kết quả của sự biến đổi tương ứng một giá trị độ sáng của một điểm ảnh với một giá trị nguyên dương. Thông thường nó xác định trong khoảng 0...255. Tùy thuộc vào giá trị xám mà mỗi điểm ảnh được biểu diễn trên 1, 4, 8, 24 hay 32 bit.

Chuyển đổi ảnh màu sang ảnh xám bằng phương pháp Grayscale: ảnh trước khi đưa vào model để nhận dạng cần được tiền xử lý trước. Model được train cho các ảnh xám có kích thước 28x28 nên ta phải đưa ảnh màu thành ảnh xám bằng phương pháp Grayscale. Grayscale là hệ thống màu có mô hình đơn giản nhất với 256 cấp độ xám biến thiên từ màu đen đến màu trắng. Trong không gian màu RGB, để có một ảnh xám cần có phải có giá trị kênh màu $\text{Red}(x, y) = \text{Green}(x, y) = \text{Blue}(x, y)$ (với x, y lần lượt là tọa độ của điểm ảnh). Để chuyển ảnh màu sang ảnh xám ta dùng công thức: $I(x, y) = (2 * \text{Red}(x, y) + 5 * \text{Green}(x, y) + 1 * \text{Blue}(x, y)) / 8$ (1)

2.3.2 AI (Trí tuệ nhân tạo)

AI có thể được định nghĩa như một ngành của khoa học máy tính liên quan đến việc tự động hóa các hành vi thông minh. AI là một bộ phận của khoa học máy tính và do đó nó phải được đặt trên những nguyên lý lý thuyết vững chắc, có khả năng ứng dụng được của lĩnh vực này. Nói nôm na cho dễ hiểu: đó là trí tuệ của máy móc được tạo ra bởi con người. Trí tuệ này có thể tư duy, suy nghĩ, học hỏi,... như trí tuệ con người. Xử lý dữ liệu ở mức rộng lớn hơn, quy mô hơn, hệ thống, khoa học và nhanh hơn so với con người.

Tuy nhiên hiện nay, công nghệ AI vẫn đang còn rất nhiều hạn chế. Đơn cử như Alexa - một quản gia tuyệt vời, một trong những biểu tượng phổ biến nhất về ứng dụng của trí thông minh nhân tạo nhưng vẫn không thể vượt qua bài kiểm tra Turing.

Tóm lại, những gì chúng ta đang thực hiện với AI hiện nay nằm trong khái niệm “AI hẹp” (Narrow AI). Công nghệ này có khả năng thực hiện các nhiệm vụ cụ thể một cách tương tự, hoặc tốt hơn con người. Ví dụ về “AI hẹp” trong thực tế như công nghệ phân loại hình ảnh của Pinterest hay nhận diện khuôn mặt để tag bạn bè trên Facebook.

Những công nghệ này thể hiện một số khía cạnh của trí thông minh con người, nhưng làm thế nào để được như vậy? Trí tuệ đó đến từ đâu? Hãy cùng đến với vòng tròn tiếp theo: học máy (machine learning).

2.3.3 Machine learning.

a, Khái niệm

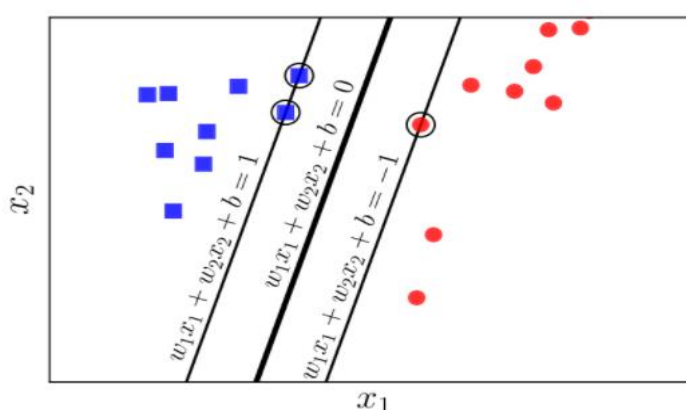
Machine Learning là một thuật ngữ rộng để chỉ hành động bạn dạy máy tính cải thiện một nhiệm vụ mà nó đang thực hiện. Cụ thể hơn, machine learning đề cập tới bất kỳ hệ thống mà hiệu suất của máy tính khi thực hiện một nhiệm vụ sẽ trở nên tốt hơn sau khi hoàn thành nhiệm vụ đó nhiều lần. Hay nói cách khác, khả năng cơ bản nhất của machine learning là sử dụng thuật toán để phân tích những thông tin có sẵn, học hỏi từ nó rồi đưa ra quyết định hoặc dự đoán về một thứ gì đó có liên quan. Thay vì tạo ra một phần mềm với những hành động, hướng dẫn chi tiết để thực hiện một nhiệm vụ cụ thể, máy tính được “huấn luyện” bằng cách sử dụng lượng dữ liệu và các thuật toán để học cách thực hiện nhiệm vụ.

Machine learning cần một tập dữ liệu với đầy đủ các đặc trưng và labels. Dựa vào các thuật toán của machine mà mô hình có thể học và ghi nhớ để có thể dự đoán được kết quả mong muốn dựa vào sự so sánh đặc trưng của đối tượng.

b, Một số thuật toán Machine Learning

➤ Support Vector Machines

Thuật toán cố gắng xây dựng một siêu mặt phẳng trong không gian nhiều chiều để phân biệt các đối tượng ở các lớp khác nhau; Làm sao cho khoảng cách giữa 2 đối tượng khác label gần nhau nhất có khoảng cách cực đại.

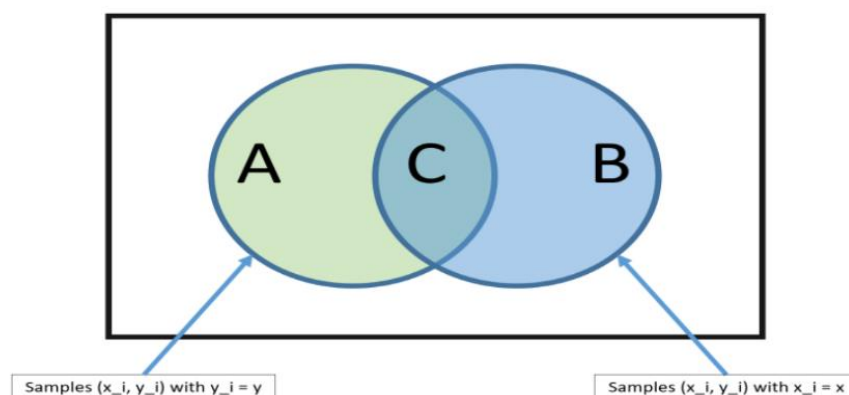


Hình 2.5 Các điểm gần mặt phân cách nhất của 2 classes

➤ Mô hình xác suất (Probabilistic Models)

Các mô hình này cố gắng giải quyết bài toán bằng phân bố xác suất. Một thuật toán phổ biến nhất là phân loại Naive Bayes; Nó sử dụng lý thuyết Bayes và giả thiết

các đặc trưng là độc lập. Điểm mạnh của mô hình xác suất là đơn giản nhưng hiệu quả. Đầu ra của nó không chỉ là label mà còn đi kèm xác suất thể hiện độ chính xác cho kết quả đó.



Hình 2.6 Tập giá trị của dữ liệu

2.3.4 Deep Learning

Có thể nói tính đến nay, AI đã gặt hái được khá nhiều bước tiến lớn. Hãy suy nghĩ về nó như là một loại machine learning với "mạng thần kinh - neural networks" sâu có thể xử lý dữ liệu theo cách tương tự như một bộ não con người có thể thực hiện. Điểm khác biệt chính ở đây là ta chỉ cần đưa cho deep learning một tập dữ liệu lớn, nó sẽ tự học, tự trích xuất đặc trưng, và đưa ra kết quả mong muốn.

Tóm lại: Deep learning là loại machine learning mà trong đó máy tự đào tạo chính nó. Deep learning đòi hỏi rất nhiều dữ liệu đầu vào và sức mạnh tính toán hơn là machine learning, nhưng nó đã bắt đầu được triển khai bởi các công ty công nghệ lớn như Facebook, Amazon. Trong đó, một trong những cái tên nổi tiếng nhất về machine learning là AlphaGo, một máy tính có thể chơi cờ vây với chính bản thân nó cho đến khi nó có thể dự đoán những đường đi nước bước chính xác nhất đủ để đánh bại nhiều nhà vô địch trên thế giới.

Deep Learning learning đã cho phép ứng dụng nhiều vấn đề thực tế của máy đồng thời mở rộng lĩnh vực tổng thể của trí tuệ nhân tạo. Deep learning phá vỡ các cách thức con người làm việc bằng cách làm cho tất cả các loại máy móc trợ giúp có thể thực hiện được, gần hoặc giống hệt con người. Ô tô không người lái, chăm sóc sức khỏe con người tốt hơn... Tất cả đều được hiện thực hóa trong thời đại ngày nay. AI là

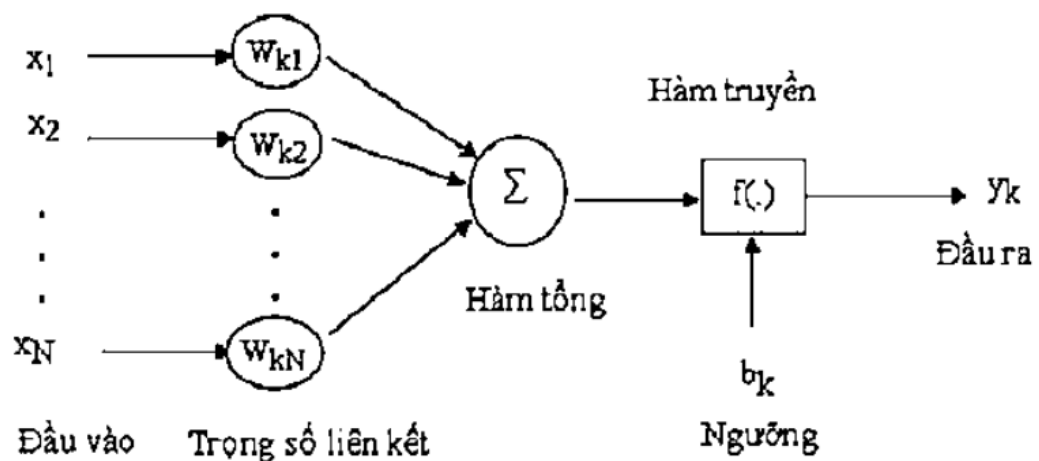
hiện tại và tương lai của thế giới. Với sự trợ giúp của deep Learning, AI có thể hiện thực hóa ước mơ khoa học giả tưởng mà chúng ta đã tưởng tượng từ rất lâu.

2.3.5 Mạng nơ ron

a, Giới thiệu về mạng Nơ ron

Mạng nơ ron nhân tạo là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu, ...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron.

Cấu trúc nơ ron nhân tạo



Hình 2.7 Cấu tạo 1 Nơ ron

b, Một số kiểu mạng Nơ ron

➤ **Mạng truyền thẳng:** Dòng dữ liệu từ đơn vị đầu vào đến đơn vị đầu ra chỉ được truyền thẳng. Việc xử lý dữ liệu có thể mở rộng ra nhiều lớp, nhưng không có các liên kết phản hồi. Nghĩa là, các liên kết mở rộng từ các đơn vị đầu ra tới các đơn vị đầu vào trong cùng một lớp hay các lớp trước đó là không cho phép.

➤ **Mạng hồi quy:** Có chứa các liên kết ngược. Khác với mạng truyền thẳng, các thuộc tính động của mạng mới quan trọng. Trong một số trường hợp, các giá trị kích hoạt của các đơn vị trải qua quá trình nói lỏng (tăng giảm số đơn vị và thay đổi các liên kết) cho đến khi mạng đạt đến một trạng thái ổn định và các giá trị kích hoạt

không thay đổi nữa. Trong các ứng dụng khác mà cách chạy động tạo thành đầu ra của mạng thì những sự thay đổi các giá trị kích hoạt là đáng quan tâm.

c, Các phương pháp huấn luyện mạng Nơ ron

➤ **Học có giám sát:** Mạng được huấn luyện bằng cách cung cấp cho nó các cặp mẫu đầu vào và các đầu ra mong muốn (target values). Các cặp được cung cấp bởi "thầy giáo", hay bởi hệ thống trên đó mạng hoạt động. Sự khác biệt giữa các đầu ra thực tế so với các đầu ra mong muốn được thuật toán sử dụng để thích ứng các trọng số trong mạng. Điều này thường được đưa ra như một bài toán xấp xỉ hàm số - cho dữ liệu huấn luyện bao gồm các cặp mẫu đầu vào x , và một đích tương ứng t , mục đích là tìm ra hàm $f(x)$ thỏa mãn tất cả các mẫu học đầu vào.

➤ **Học không giám sát:** Là việc học không có bất kỳ một sự giám sát nào. Trong bài toán học không giám sát, tập dữ liệu huấn luyện được cho dưới dạng: $D = \{(x_1, x_2, \dots, x_N)\}$, với (x_1, x_2, \dots, x_N) là vector đặc trưng của mẫu huấn luyện. Nhiệm vụ của thuật toán là phải phân chia tập dữ liệu D thành các nhóm con, mỗi nhóm chứa các vector đầu vào có đặc trưng giống nhau. Như vậy với học không giám sát, số lớp phân loại chưa được biết trước, và tùy theo tiêu chuẩn đánh giá độ tương tự giữa các mẫu mà ta có thể có các lớp phân loại khác nhau.

➤ **Học củng cố:** Là sự tổ hợp của cả 2 mô hình trên. Với vector đầu vào quan sát vector đầu ra do mạng tính được. Nếu kết quả được xem là "tốt" thì mạng sẽ được thưởng theo nghĩa tăng các trọng số kết nối lên; ngược lại mạng sẽ bị phạt, các trọng số kết nối không thích hợp sẽ được giảm xuống. Do đó học tăng cường là học theo nhà phê bình (*critic*), ngược với học có giám sát là học theo thầy giáo (*teacher*).

2.3.6 Mạng Nơ ron tích chập

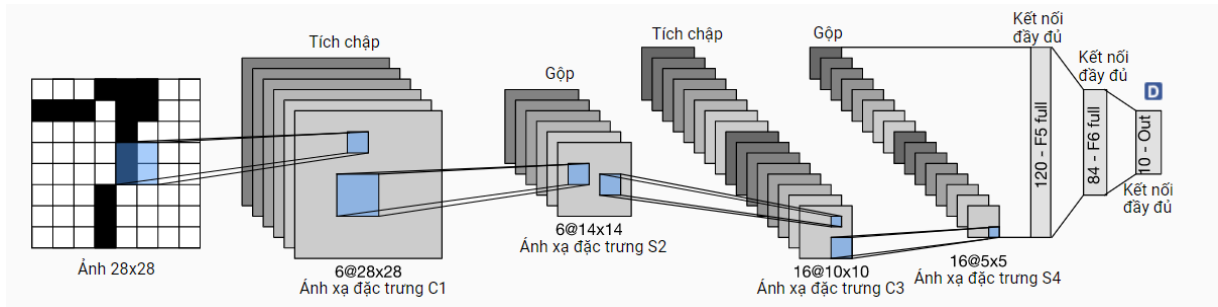
a, Định nghĩa

Những năm gần đây, ta đã chứng kiến được nhiều thành tựu vượt bậc trong ngành Thị giác máy tính (Computer Vision). Các hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động.

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Yann LeCun, Leon Bottou,

Yoshua Bengio và Patrick Haffner đã đề xuất kiến trúc mạng thần kinh để nhận dạng ký tự viết tay và in bằng máy vào năm 1990 mà họ gọi là LeNet-5. Kiến trúc đơn giản và dễ hiểu, đó là lý do tại sao nó chủ yếu được sử dụng như một bước đầu tiên để giảng dạy.

Một cách đơn giản, ta có thể xem LeNet gồm hai phần: (i) một khối các tầng tích chập; và (ii) một khối các tầng kết nối đầy đủ. Trước khi đi vào các chi tiết cụ thể, hãy quan sát tổng thể mô hình bên dưới.

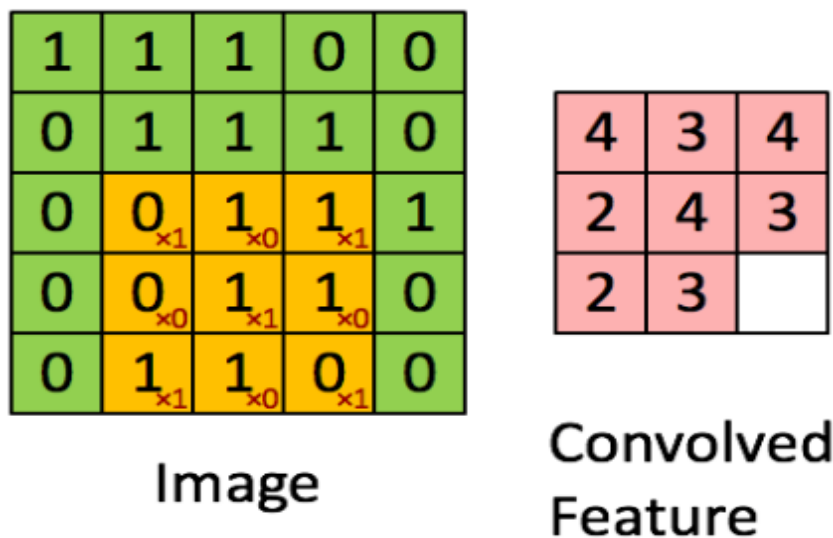


Hình 2.8 Mô hình mạng CNN

b, Tích chập

Tích chập được sử dụng đầu tiên trong xử lý tín hiệu số (Signal processing). Nhờ vào nguyên lý biến đổi thông tin, các nhà khoa học đã áp dụng kỹ thuật này vào xử lý ảnh và video số.

Để dễ hình dung, ta có thể xem tích chập như một cửa sổ trượt (sliding window) áp đặt lên một ma trận. Bạn có thể theo dõi cơ chế của tích chập qua hình minh họa bên dưới.



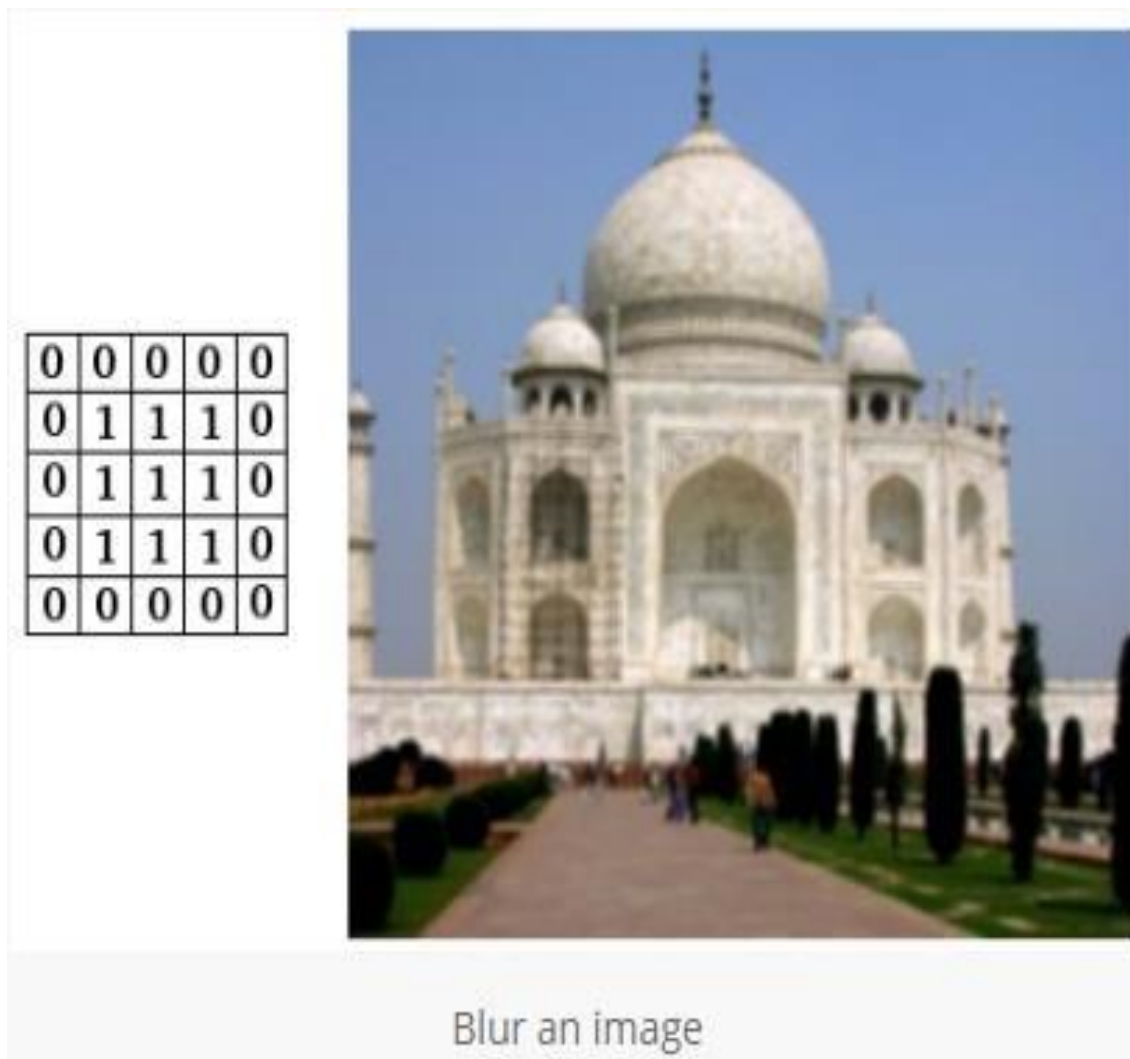
Hình 2.9 Minh họa tích chập

Ma trận bên trái là một bức ảnh đen trắng. Mỗi giá trị của ma trận tương đương với một điểm ảnh (pixel), 0 là màu đen, 1 là màu trắng (nếu là ảnh grayscale thì giá trị biến thiên từ 0 đến 255).

Sliding window còn có tên gọi là kernel, filter hay feature detector. Ở đây, ta dùng một ma trận filter 3×3 nhân từng thành phần tương ứng (element-wise) với ma trận ảnh bên trái. Giá trị đầu ra do tích của các thành phần này cộng lại.

Kết quả của tích chập là một ma trận (convolved feature) sinh ra từ việc trượt ma trận filter và thực hiện tích chập cùng lúc lên toàn bộ ma trận ảnh bên trái. Dưới đây là một vài ví dụ của phép toán tích chập.

Ta có thể làm mờ bức ảnh ban đầu bằng cách lấy giá trị trung bình của các điểm ảnh xung quanh cho vị trí điểm ảnh trung tâm.



Hình 2.10 Ảnh mờ sau khi chập

Về bản chất thực hiện làm mờ ảnh chính là tạo ra ảnh mới sao cho giá trị mức xám của mỗi pixel ở ảnh mới đúng bằng giá trị trung bình của điểm tương ứng và 8 điểm lân cận trên ảnh ban đầu. Nói cách khác, với mỗi điểm trên hình ban đầu, bạn tính giá trị trung bình của nó (tại hàng i cột j) với 8 điểm xung quanh rồi viết lại giá trị mức xám ở vị trí tương ứng (cũng tại hàng i cột j) lên ảnh mới, sau đó tương tự với các điểm tiếp theo.

2	6	8	6	3	4	7	9	
9	0	3	1	2	4	1	4	
1	3	5	2	5	1	3	3	tính trung bình các điểm xung quanh
9	0	8	7	8	9	0	3	
1	3	5	5	3	8	9	3	
6	7	3	2	4	2	1	4	$\frac{2+5+1+7+8+9+5+3+8}{9}$
3	5	5	6	5	4	3	5	bằng
3	6	7	1	2	4	5	6	$\frac{2}{9} + \frac{5}{9} + \frac{1}{9} + \frac{7}{9} + \frac{8}{9} + \frac{9}{9} + \frac{5}{9} + \frac{3}{9} + \frac{8}{9}$

Vậy có nghĩa là ta đang tính trung bình cộng của 9 pixel (pixel tại điểm đó và 8 pixel lân cận), vậy phép tính đó cũng giống như nhân từng giá trị mức sáng của các pixel lân cận với $1/9$ sau đó cộng lại với nhau. Vậy, nếu có một ma trận 3×3 với tất cả các con số trong ma trận đều là $1/9$, ta nhân từng phần tử của ma trận này với mức sáng của pixel tương ứng và cộng lại, ta sẽ có kết quả giống nhau (xem hình vẽ bên dưới).

2	6	8	6	3	4	7	9										
9	0	3	1	2	4	1	4										
1	3	5	2	5	1	3	3	nhân từng phần tử tương ứng và cộng lại									
9	0	8	7	8	9	0	3										
1	3	5	5	3	8	9	3										
6	7	3	2	4	2	1	4	<table border="0"> <tr> <td>2</td><td>5</td><td>1</td> </tr> <tr> <td>7</td><td>8</td><td>9</td> </tr> <tr> <td>5</td><td>3</td><td>8</td> </tr> </table>	2	5	1	7	8	9	5	3	8
2	5	1															
7	8	9															
5	3	8															
3	5	5	6	5	4	3	5	<table border="0"> <tr> <td>$\frac{1}{9}$</td><td>$\frac{1}{9}$</td><td>$\frac{1}{9}$</td> </tr> <tr> <td>$\frac{1}{9}$</td><td>$\frac{1}{9}$</td><td>$\frac{1}{9}$</td> </tr> <tr> <td>$\frac{1}{9}$</td><td>$\frac{1}{9}$</td><td>$\frac{1}{9}$</td> </tr> </table>	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$															
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$															
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$															
3	6	7	1	2	4	5	6										

Áp dụng tương tự cho mọi pixel trên ảnh ban đầu và lấy từng kết quả cho từng pixel của ảnh mới, ta sẽ được ảnh mới chính là ảnh mờ của ảnh ban đầu.

Ngoài ra, ta có thể phát hiện biên cạnh bằng cách tính vi phân (độ dị biệt) giữa các điểm ảnh lân cận.

	0	1	0	
	1	-4	1	
	0	1	0	



Edge detection

Hình 2.11 Hình ảnh phát hiện biên sau khi chụp

c, Kiến trúc mô hình học sâu Lenet 5

Phần này sẽ giới thiệu một trong những mạng sâu được sử dụng rộng rãi đó là mạng tích chập sâu (deep convolutional networks). Chúng ta sẽ làm việc cụ thể với mạng tích chập để giải quyết bài toán phân loại chữ số viết tay từ tập dữ liệu MNIST.

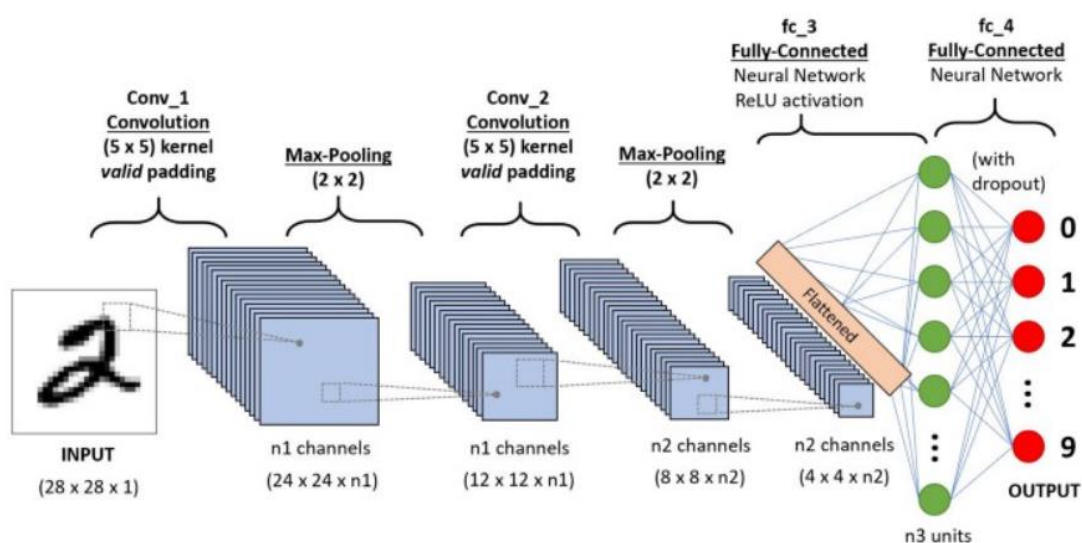
Chúng ta sẽ bắt đầu mạng tích chập với việc sử dụng mạng truyền thống để giải quyết bài toán này trong phần trước. Mặc dù nhiều phép toán lặp nhưng chúng ta sẽ xây dựng mạng hiệu quả hơn. Chúng ta sẽ khám phá ra rất nhiều kỹ thuật hiệu quả: Tích chập (convolution), giảm số chiều (pooling), sử dụng GPUs để huấn luyện được nhiều dữ liệu hơn chúng ta đã thực hiện trên mạng cũ, mở rộng giải thuật huấn luyện dữ liệu (để giảm quá khớp – overfitting), sử dụng kỹ thuật dropout để giảm overfitting, việc sử dụng tổng hợp các mạng và các kỹ thuật khác. Kết quả là hệ thống làm việc gần

như con người. Trong số 10.000 bức ảnh huấn luyện, hệ thống của chúng ta sẽ phân loại đúng 9.967 bức ảnh.

Phần còn lại của chương sẽ thảo luận về học sâu dưới góc độ tổng quan và chi tiết. Chúng ta sẽ tìm hiểu làm thế nào để các mô hình mạng nơron tích chập có thể ứng dụng để giải quyết các bài toán nhận dạng tiếng nói, xử lý ngôn ngữ tự nhiên và các lĩnh vực khác. Và chúng ta sẽ nghiên cứu về mạng nơron trong tương lai và học sâu (deep learning), từ các ý tưởng như giao diện người sử dụng hướng đích đến vai trò của học sâu trong trí tuệ nhân tạo.



Chúng ta đã sử dụng mạng nơron mà trong đó các tầng liên kết đầy đủ với nhau. Tức là mỗi nơron trong mạng liên kết với tất cả các nơron trong tầng liền kề.

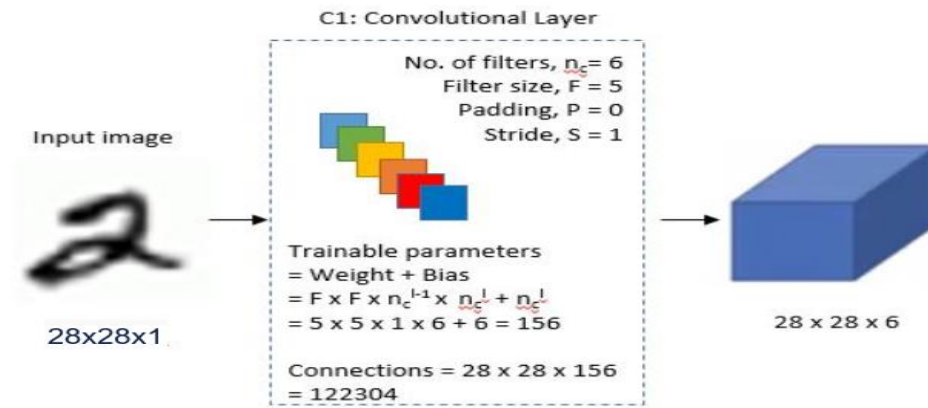


Hình 2.12 Mô hình mạng CNN Lenet 5

Chi tiết các lớp:

➤ **Lớp tích chập thứ nhất:**

Đầu vào cho LeNet-5 là ảnh xám 28x28 đi qua lớp tích đầu tiên với 6 feature map hoặc bộ lọc có kích thước 5×5 và một bước sải. Kích thước hình ảnh thay đổi từ 28x28x1 thành 28x28x6.



Hình 2.13 *Đổi kích thước ảnh đầu vào lớp tích chập đầu tiên*

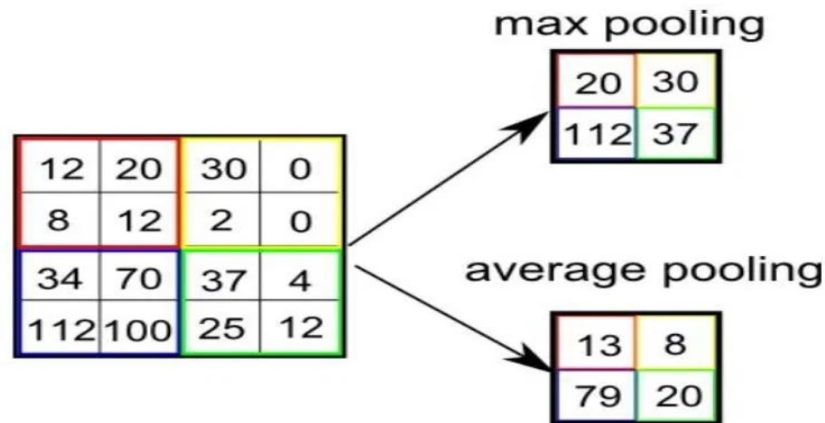
➤ **Lớp thứ hai: Lớp gộp (Pooling)**

Ngoài các lớp tích chập vừa mô tả, mạng nơron tích chập cũng chứa các lớp pooling. Lớp pooling thường được sử dụng ngay sau lớp tích chập. Những gì các lớp pooling làm là đơn giản hóa các thông tin ở đầu ra từ các lớp tích chập nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm việc tính toán trong model.

Mục đích của pooling rất đơn giản, nó làm giảm số hyperparameter mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh overfitting. Loại pooling ta thường gặp nhất là max pooling, lấy giá trị lớn nhất trong một pooling window. Pooling hoạt động gần giống với convolution, nó cũng có 1 cửa sổ trượt gọi là pooling window, cửa sổ này trượt qua từng giá trị của ma trận dữ liệu đầu vào (thường là các feature map trong convolutional layer), chọn ra một giá trị từ các giá trị nằm trong cửa sổ trượt (với max pooling ta sẽ lấy giá trị lớn nhất).

Gọi pooling size kích thước $K \times K$. Input của pooling layer có kích thước $H \times W \times D$, ta tách ra làm D ma trận kích thước $H \times W$. Với mỗi ma trận, trên vùng kích thước $K \times K$ trên ma trận ta tìm maximum hoặc average của dữ liệu rồi viết vào ma trận kết quả. Quy tắc về stride và padding áp dụng như phép tính convolution trên ảnh.

Có 2 loại pooling layer phổ biến là: max pooling và average pooling.



Hình 2.14 Các loại lớp lấy mẫu con

➤ **Lớp thứ ba: Lớp tích chập thứ hai**

Tiếp theo, có một lớp tích chập thứ hai với 16 feature map có kích thước 5×5 và sải bước là 1. Trong lớp này, chỉ 10 trong số 16 feature map được kết nối với 6 feature map của lớp trước như hình dưới đây:

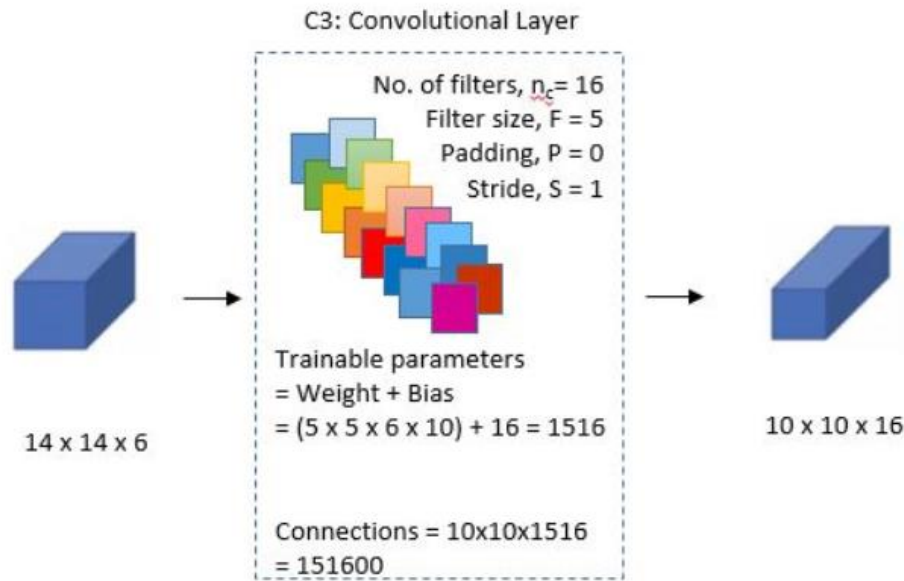
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED
BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

Hình 2.15 Feature Map

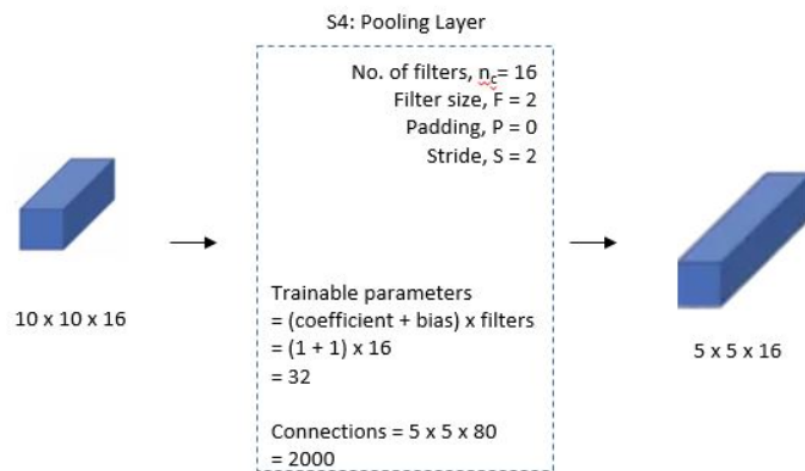
Nguyên nhân chính là để phá vỡ tính đối xứng trong mạng và giữ cho số lượng kết nối trong giới hạn hợp lý. Đó là lý do tại sao số lượng tham số đào tạo trong các lớp này là 1516 thay vì 2400 và tương tự, số lượng kết nối là 151600 thay vì 240000.



Hình 2.16 Kích thước ảnh khi đi qua lớp tích chập thứ hai

➤ **Lớp thứ tư: Lớp gộp (pooling)**

Là lớp max pooling với kích thước bộ lọc 2×2 và sải bước là 2. Lớp này giống với lớp thứ hai (S2) ngoại trừ nó có 16 feature map nên đầu ra sẽ giảm xuống còn $5 \times 5 \times 16$.



Hình 2.17 Ảnh sau khi thông qua lớp gộp thứ hai

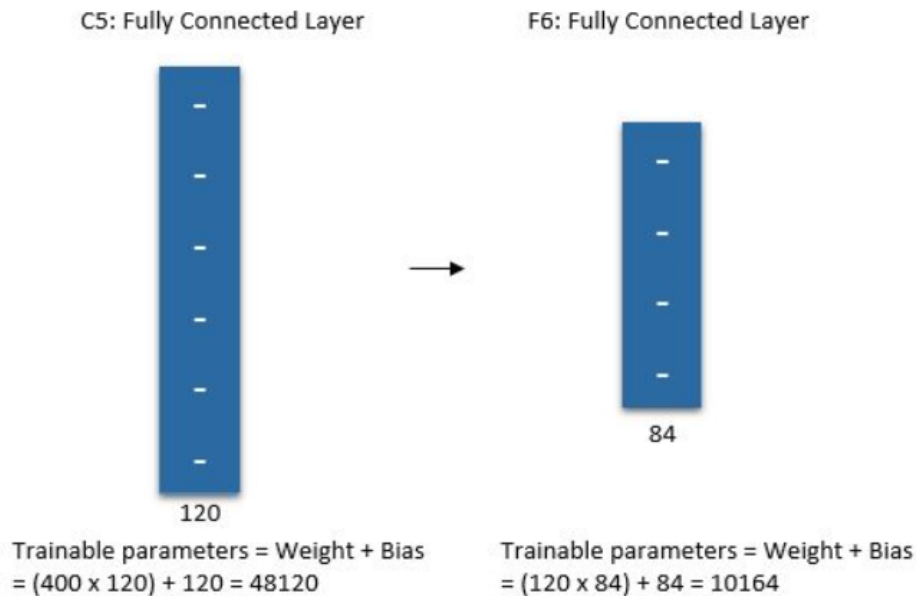
➤ **Lớp thứ năm: Lớp kết nối đầy đủ (fully connected convolutional)**

Sau khi ảnh được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của ảnh (ví dụ mắt, mũi, khung

mặt,...) thì tensor của output của layer cuối cùng, kích thước $H*W*D$, sẽ được chuyển về 1 vector kích thước $(H*W*D)$

Lớp thứ năm (C5) là fully connected convolutional với 120 feature map với kích thước $1*1$. Mỗi 120 đơn vị trong C5 được kết nối với tất cả 400 nút ($5*5*16$) trong lớp thứ tư S4.

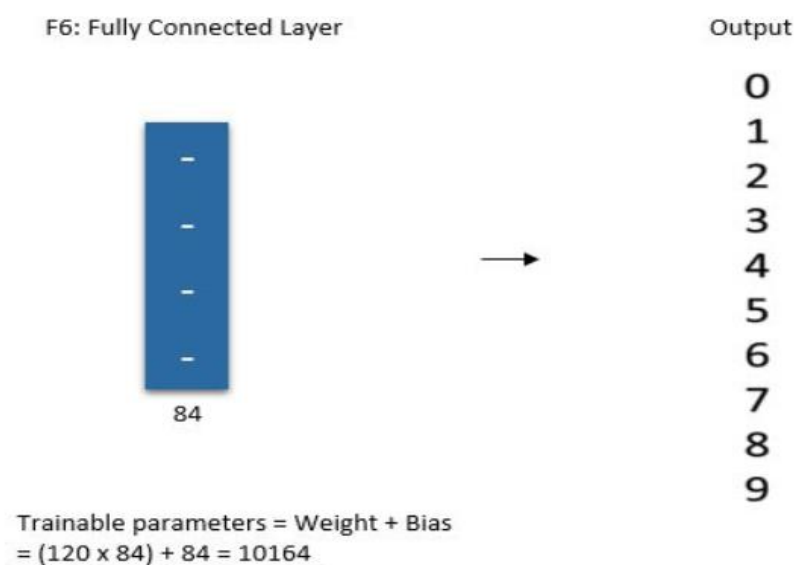
➤ **Lớp thứ sáu: Lớp kết nối đầy đủ với 84 đơn vị**



Hình 2.18 Lớp kết nối đầy đủ với 84 đơn vị

➤ **Đầu ra (Output):**

Có một lớp đầu ra softmax được kết nối đầy đủ với 10 giá trị có thể có tương ứng với các chữ số từ 0 đến 9.



Hình 2.19 Ảnh đầu ra

➤ Bảng tóm lược kiến trúc mô hình CNN Lenet 5

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

Hình 2.20 Bảng tóm tắt mô hình Lenet 5

CHƯƠNG 3. KỸ THUẬT NHẬN DẠNG KÝ TỰ

3.1 Nhận dạng ký tự sử dụng mạng Nơ ron lan truyền ngược

a, Cơ sở lý thuyết

Truyền ngược (back propagation) là một phương pháp phổ biến để huấn luyện các mạng thần kinh nhân tạo được sử dụng kết hợp với một phương pháp tối ưu hóa như gradient descent. Phương pháp này tính toán gradient của hàm tổn thất với tất cả các trọng số có liên quan trong mạng nơ ron đó. Gradient này được đưa vào phương pháp tối ưu hóa, sử dụng nó để cập nhật các trọng số, để cực tiểu hóa hàm tổn thất.

Truyền ngược yêu cầu một đầu ra mong muốn, đã biết cho mỗi giá trị đầu vào để tính toán gradient hàm tổn thất. Do đó, nó thường được xem là một phương pháp học có giám sát, mặc dù nó cũng được sử dụng trong một số mạng không có giám sát như autoencoders. Nó là một tổng quát hóa của quy tắc delta cho các mạng nuôi tiến đa tầng, có thể thực hiện bằng cách sử dụng quy tắc dây chuyền để tính toán lặp đi lặp lại các gradient cho mỗi lớp. Truyền ngược yêu cầu các hàm kích hoạt được sử dụng bởi các nơ-ron nhân tạo (hay "nút") khả vi.

b, Thuật toán

Bước 1: Xây dựng mạng tương ứng với mô hình tham số

Bước 2: Khởi tạo giá trị trọng số với giá trị ngẫu nhiên. Nạp file huấn luyện (cả ảnh đầu vào và ảnh đầu ra mong muốn)

Bước 3: Phân tích ảnh và ánh xạ tất cả ký tự tìm thấy vào các mảng một chiều.

Bước 4: Đọc giá trị đầu ra mong muốn từ file và chuyển đổi từng ký tự với giá trị nhị phân Unicode và lưu trữ riêng biệt

Bước 5: Với mỗi ký tự:

Tính toán giá trị đầu ra của mạng Feed Forward.

So sánh với giá trị đầu ra mong muốn tương ứng với từng ký tự và tính toán lỗi.

Truyền ngược giá trị từ đầu và với mỗi liên kết điều chỉnh trọng số liên kết.

Bước 6: Chuyển sang ký tự tiếp theo và lặp lại “6” cho đến khi hết ký tự.

Bước 7: Tính toán trung bình lỗi cho tất cả các ký tự.

Bước 8: Lặp lại từ bước 6 đến 8 cho tới khi đạt số đưa vào của số lần lặp lại tối đa.

c, Kết quả

Huấn luyện và nhận dạng được hai loại font: Arial và Times Roman với nhiều kích thước khác nhau đã đạt được độ chính xác 98,89%, nhưng còn một số tồn tại cần được phát triển để đạt kết quả cao hơn.

3.2 Nhận dạng ký tự sử dụng công nghệ OCR

OCR là viết tắt của từ Optical Character Recognition – nhận dạng ký tự quang học. Công nghệ OCR là loại phần mềm máy tính được tạo ra để chuyển đổi hình ảnh của chữ viết tay hoặc chữ đánh máy phép chuyển đổi tài liệu dạng ảnh (nhận diện chữ trên các ảnh đầu ra của máy quét, máy ảnh, nhận dạng chữ từ file PDF, nhận dạng chữ qua ảnh, nhận diện ảnh ...) thành tài liệu có thể biên tập được (file text, file Word...), xử lý ngôn ngữ tự nhiên và bóc tách thông tin. Việc áp dụng các công nghệ này vào hệ thống giúp công việc quản lý, biên soạn lại các tài liệu dạng ảnh, tìm kiếm tài liệu trở nên đơn giản và thuận tiện.

Công nghệ OCR hoạt động theo 3 bước sau:

Bước 1: Tiền xử lý: Ảnh văn bản được chuyển đổi thành ảnh nhị phân (pixel) để thể hiện chính xác nhất. Phần mềm OCR sẽ quét và xác nhận các vị trí cần xử lý ký tự.

Bước 2: Nhận dạng ký tự: Khi trích xuất thông tin từ bước 1, OCR quét các biểu tượng gập và phân chúng thành ký tự cơ bản.

Bước 3: Xử lý hậu kỳ: Nếu thiếu bước này, độ chính xác chuyển đổi thấp, OCR sẽ tạo ra nhiều từ vô nghĩa. Thuật toán OCR sẽ quét các từ trong bộ từ điển có trong output chứa danh sách các từ được phép xuất hiện trong văn bản.

Tính năng ưu việt của công nghệ OCR:

➤ Bóc tách và nhận dạng tự động thông tin: Trong nhập liệu, trường hợp cần sử dụng OCR nhất là công việc chuyển đổi tài liệu giấy in thành tài liệu văn bản có thể đọc được khi đưa lên hệ thống, người dùng chọn lựa loại tài liệu (công văn, báo cáo, quyết định, hoặc 1 số dạng khác ...), Tài liệu sau đó được nhận dạng OCR chuyển sang dạng text và tự động bóc tách các trường thông tin cần thiết.

➤ Bóc tách và nhận dạng theo form mẫu do người dùng định nghĩa: Người dùng định nghĩa các vùng cần bóc tách, sau đó lưu lại thành. Khi đưa tài liệu lên hệ thống người dùng chọn mẫu tài liệu tương ứng, sau đó hệ thống sẽ tự động nhận dạng và trích xuất thông tin tại các vùng được đánh dấu trong mẫu.

➤ Bóc tách và nhận dạng trực tiếp trên file tài liệu: Người dùng chọn file cần bóc tách thông tin sau đó lựa chọn vùng ảnh và gán vùng ảnh đó với trường thông tin cụ thể, hệ thống sẽ nhận dạng và tự động đưa vào các trường thông tin tương ứng đó.

OCR trong ứng dụng thực tế:

Có lẽ chúng ta đều đã từng một lần được sử dụng OCR nhưng không để ý, vì chúng ta thường chỉ hay sử dụng chúng một cách gián tiếp dựa trên máy scanner.

OCR được coi như một ứng dụng phần mềm có khả năng tích hợp trực tiếp vào máy tính hay một phần cứng bất kỳ. Ví dụ như: PDF OCR, IONE,....

3.3 Nhận dạng ký tự sử dụng công nghệ ICR

ICR là công nghệ mở rộng của Nhận dạng ký tự quang học (OCR) tập trung vào các ký tự văn bản in. Trong khi đó, các dịch vụ ICR cho phép hệ thống máy tính học các kiểu và phong chữ viết tay khác nhau để cải thiện độ chính xác và nhận dạng văn bản.

Công nghệ ICR ghi lại chữ viết tay từ các tệp hình ảnh. Đây là phiên bản tiên tiến của công nghệ OCR trong đó các ký tự in được chụp. Nhận dạng chữ viết tay đang trong giai đoạn chuyển đổi khi công nghệ không ngừng cải tiến để mang lại độ chính xác và tỷ lệ nhận dạng tốt hơn. Những thay đổi và điều chỉnh trong lĩnh vực này là không đổi, kết hợp các khả năng với thời gian để làm cho công nghệ trở nên đáng tin cậy hơn.

Trên thực tế khi so sánh với OCR, chúng ta có thể nói rằng kết quả của ICR vẫn còn thiếu tính đúng đắn. Lý do là phần mềm ICR hiện có đang liên tục học hỏi các mạng nơ-ron được phát triển bởi Trí tuệ nhân tạo. Dữ liệu càng nhiều thì phần mềm ICR càng có thể học và xử lý các tài liệu viết tay tốt hơn. Bất kỳ ngoại lệ nào cũng có thể dẫn đến việc giải thích dữ liệu sai.

Nhận dạng chữ viết tay có thể diễn giải một loạt các kiểu chữ và phong chữ viết tay mới bằng cách phát triển mô hình mạng thần kinh dựa trên AI, qua đó hệ thống đảm bảo khả năng tự học với từng kiểu và phong chữ tài liệu mới. Có nghĩa là bất cứ khi nào mô hình học một loại tài liệu mới, phần mềm ICR sẽ nâng cấp cơ sở dữ liệu của nó, điều này cuối cùng giúp phần mềm dự đoán chữ viết tay một cách hiệu quả với độ chính xác cao. Với mọi dữ liệu mới cho mô hình Mạng nơ ron nhân tạo (ANN), dữ

liệu mới cũng như dữ liệu trước đó được sử dụng để dự đoán chữ viết tay, phong chữ và kiểu.

Phần mềm ICR được tích hợp vào một hệ thống nhận đầu vào của một tài liệu viết tay. Nó quét thông tin và trích xuất dữ liệu từ các tài liệu.

Chúng ta hãy hiểu hoạt động của công nghệ nhận dạng chữ viết tay trong trường hợp sử dụng xác minh tài liệu số của một ngân hàng. Người dùng cuối sẽ hiển thị ID của họ trước máy ảnh. Phần mềm ICR sẽ quét tài liệu và trích xuất thông tin từ ID. Người dùng cũng có thể được yêu cầu tải lên bản chấp thuận viết tay. Văn bản sẽ được quét và trích xuất. Cả thông tin trích xuất sẽ được xác minh và dữ liệu sẽ được gửi đến ngân hàng.

Quá trình trích xuất dữ liệu tự động giúp doanh nghiệp loại bỏ các khâu nhập dữ liệu tốn kém thời gian và không hiệu quả. Không chỉ tiết kiệm thời gian mà còn tiết kiệm chi phí thuê nhân viên cho công việc nhập liệu. Dữ liệu được trích xuất có thể được sử dụng thêm trong một hệ thống luồng dữ liệu tự động có thể giúp các doanh nghiệp đảm bảo mô hình hóa dự đoán trên các tập dữ liệu lớn để làm cho hệ thống hiệu quả hơn và được tối ưu hóa.

3.4 Công nghệ nhận dạng đánh dấu OMR

Là công nghệ nhận biết dấu quang học trên trang giấy theo một định dạng nhất định. Công nghệ này có thể xác định các dấu hiệu đặc biệt đã được đánh dấu trên trang giấy tại các vị trí định trước bằng quang học.

Nó cho phép điều khiển một số loại máy quét ảnh, nhập tự động và số hóa các dữ liệu ảnh theo biểu mẫu được thiết kế phù hợp, kiểm tra, điều chỉnh điểm chọn trực quan trên ảnh quét và kết xuất ra báo cáo dạng text để truy nhập vào các phần mềm xử lý dữ liệu khác.

Công nghệ này thường được áp dụng để xử lý các dữ liệu từ phiếu điều tra hay các bài thi trắc nghiệm.

Tính năng ưu việt:

- Công nghệ có thể nhận biết các kích cỡ đánh dấu khác nhau với độ chính xác và linh hoạt cao.
- Điều chỉnh hình ảnh được scan để bù lại chất lượng thấp của máy scan.
- Sử dụng được với nhiều loại công cụ viết (bút chì, bút bi, bút nhớ, ...).

- Dễ dàng kiểm tra lại câu trả lời thông qua việc dùng xóa hoặc đánh dấu với kích cỡ lớn hơn.
- Để tránh việc đọc sai hình ảnh, công nghệ OMR đọc dựa trên đánh dấu thời gian nên bù đắp được các lỗi kỹ thuật của máy scan.
- Giúp tiết kiệm thời gian và chi phí.

CHƯƠNG 4. DEMO CHƯƠNG TRÌNH

4.1 Mô tả

Xây dựng chương trình để ứng dụng mô hình học sâu Lenet 5 (Mạng CNN) trong việc nhận dạng ký tự số áp dụng vào việc nhập điểm thi cho sinh viên tại học viện cơ sở.

Chương trình quản lý các chức năng liên quan như: quản lý khoa, quản lý lớp, quản lý sinh viên, quản lý môn học, quản lý điểm và thống kê. Ở chức năng quản lý điểm thi, chương trình đọc file ảnh điểm do người dùng vẽ trực tiếp vào để lưu điểm cho sinh viên theo từng môn.

4.2 Xây dựng Model

Bước 1: Import một số thư viện cần thiết.

```
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import math
```

Bước 2: Lấy dữ liệu từ tập MNIST bằng Tensorflow và chia dữ liệu thành 2 tập `train_set` và `test_set`.

```
mnist = tf.keras.datasets.mnist
(training_images, training_labels), (test_images, test_labels) =
mnist.load_data()
```

Bước 3: Chuẩn hóa lại dữ liệu bằng cách chia các giá trị của pixel cho 255.

```
training_images = training_images / 255.0
training_images = training_images.reshape(60000, 28,28, 1)
test_images = test_images / 255.0
test_images = test_images.reshape(10000, 28,28,1)
```

Bước 4: Mô hình Lenet 5

```
model = Sequential()
# C1 Convolutional Layer
```

```
model.add(Conv2D(6, kernel_size=(5, 5), activation='tanh',
input_shape=(28,28,1), padding='same'))
# S2 Pooling Layer
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
# C3 Convolutional Layer
model.add(Conv2D(16, kernel_size=(5, 5), activation='tanh'))
# S4 Pooling Layer
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
# C5 Fully Connected Convolutional Layer
model.add(Conv2D(120, kernel_size=(5, 5), activation='tanh'))
#Flatten the CNN output so that we can connect it with fully connected layers
model.add(Flatten())
# FC6 Fully Connected Layer
model.add(Dense(84, activation='tanh'))
#Output Layer with softmax activation
model.add(Dense(10, activation='softmax'))
model.summary()
```

Bước 5: Training model

```
# Compile the model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
hist = model.fit(training_images, training_labels, epochs=10, batch_size=128,
validation_data=(test_images, test_labels), verbose=1)
```

Bước 6: Đánh giá Model sau khi training

```
test_score = model.evaluate(test_images, test_labels)
print("Test loss {:.4f}, accuracy {:.2f}%".format(test_score[0], test_score[1] *
100))
```

Bước 7: Detect thử một vài hình ảnh trong tập test_set

```
# Helper libraries
import numpy as np
import matplotlib.pyplot as plt
```

```
import random

from sklearn.metrics import classification_report

# A helper function that returns 'red'/'black' depending on if its two input
# parameter matches or not.
def get_label_color(val1, val2):
    if val1 == val2:
        return 'black'
    else:
        return 'red'

# Predict the labels of digit images in our test dataset.
predictions = model.predict(test_images)

# As the model output 10 float representing the probability of the input image
# being a digit from 0 to 9, we need to find the largest probability value
# to find out which digit the model predicts to be most likely in the image.
prediction_digits = np.argmax(predictions, axis=1)
print(prediction_digits)

# Then plot 100 random test images and their predicted labels.
# If a prediction result is different from the label provided label in "test"
# dataset, we will highlight it in red color.
plt.figure(figsize=(18, 18))
for i in range(100):
    ax = plt.subplot(10, 10, i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    image_index = random.randint(0, len(prediction_digits))
    img_path = test_images[image_index]
    # print(img_path)
    # print(len(img_path.shape)) // return 3
    if(len(img_path.shape) == 3):
        plt.imshow(np.squeeze(img_path), cmap=plt.cm.gray)
```

```
elif(len(img_path.shape) == 2):  
    plt.imshow(img_path)  
else:  
    print("Higher dimensional data")  
    ax.xaxis.label.set_color(get_label_color(prediction_digits[image_index],\  
                                              test_labels[image_index]))  
    plt.xlabel('Predict: %d' % prediction_digits[image_index])  
plt.show()
```

Bước 8: Convert model sang file có đuôi tflite để phía mobile sử dụng

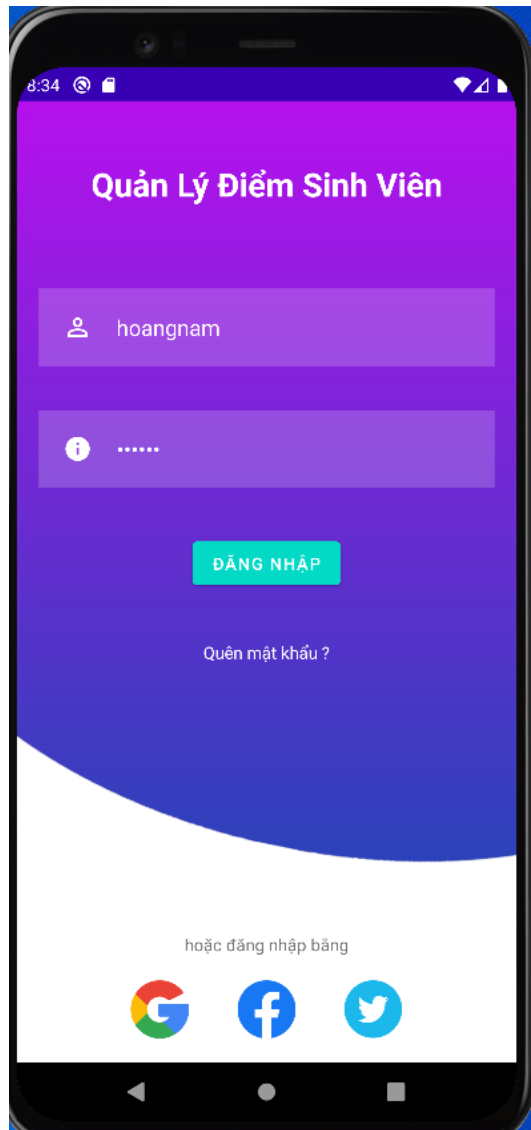
```
# Convert Keras model to TF Lite format.  
converter = tf.lite.TFLiteConverter.from_keras_model(model)  
tflite_model = converter.convert()  
# Save the TF Lite model as file  
f = open('lenet5.tflite', "wb")  
f.write(tflite_model)  
f.close()
```

4.3 Demo chương trình “Quản lý điểm thi sinh viên”

➤ **Giao diện đăng nhập**

Đăng nhập đúng => vào trang menu để chọn các chức năng liên quan.

Đăng nhập sai => báo lỗi.



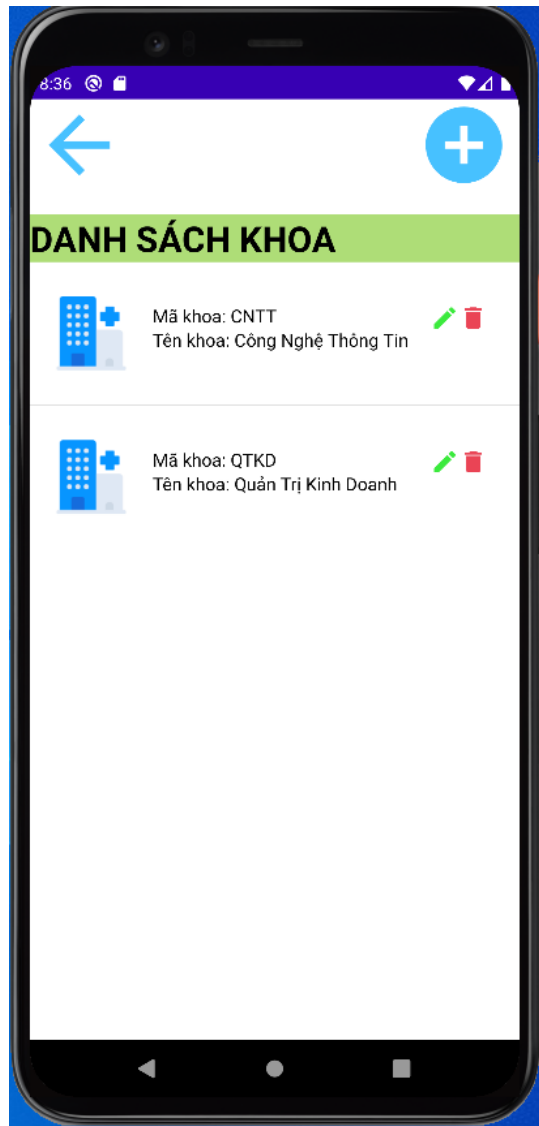
➤ Giao diện quản lý khoa

Bấm vào ô quản lý khoa thì in ra danh sách khoa.

Bấm dấu cộng để thêm thông tin của 1 khoa mới. Nếu nhập đủ và bấm lưu thì sẽ kiểm tra, không vi phạm lỗi liên quan đến khóa chính hay cách đặt tên các bản ghi thì thông báo lưu thành công. Ngược lại, báo lỗi tại vị trí bị sai.

Bấm icon sửa thì hiện ra thông tin của bản ghi đang được trỏ tới và cũng sẽ kiểm tra thông tin trước khi lưu.

Bấm icon xóa thì hiện ra hộp thoại hỏi có chắc chắn xóa hay không ? Nếu bấm có thì sẽ kiểm tra xem bản ghi đó có liên kết với bảng khác không? Nếu có in ra lỗi, nếu không thì thông báo xóa thành công. Nếu bấm không thì hủy quá trình xóa.



➤ Giao diện quản lý lớp

Bấm vào ô quản lý lớp thì in ra danh sách lớp.

Bấm dấu cộng để thêm thông tin của 1 lớp mới. Nếu nhập đủ và bấm lưu thì sẽ kiểm tra, không vi phạm lỗi liên quan đến khóa chính hay cách đặt tên các bản ghi thì thông báo lưu thành công. Ngược lại, báo lỗi tại vị trí bị sai.

Bấm icon sửa thì hiện ra thông tin của bản ghi đang được trỏ tới và cũng sẽ kiểm tra thông tin trước khi lưu.

Bấm icon xóa thì hiện ra hộp thoại hỏi có chắc chắn xóa hay không ? Nếu bấm có thì sẽ kiểm tra xem bản ghi đó có liên kết với bảng khác không? Nếu có in ra lỗi, nếu không thì thông báo xóa thành công. Nếu bấm không thì hủy quá trình xóa.

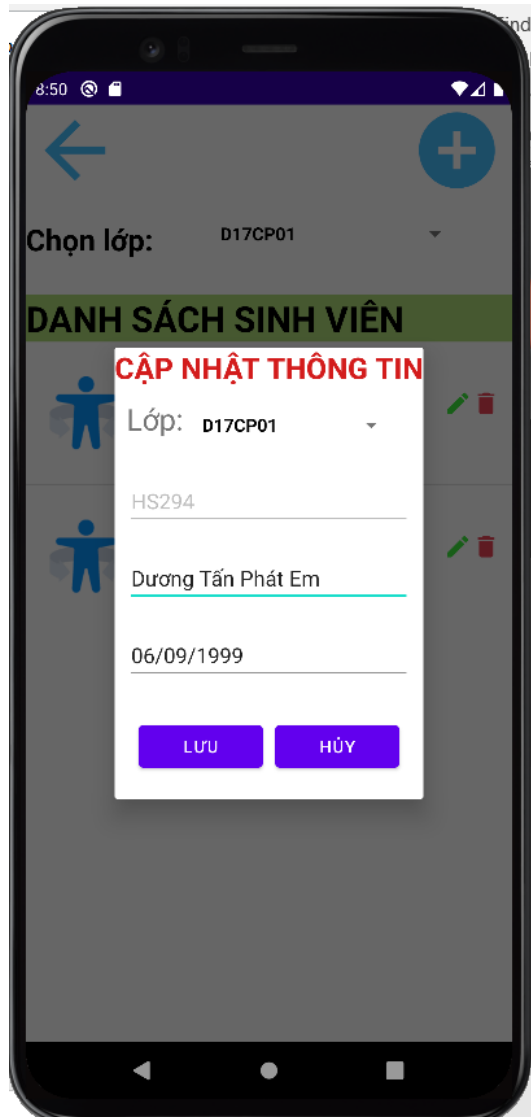


➤ **Giao diện quản lý sinh viên**

Xem danh sách sinh viên theo từng lớp: Chọn lớp và sẽ load thông tin sinh viên theo từng lớp để người dùng xem.



Thao tác thay đổi dữ liệu: thêm, sửa, xóa sinh viên: Bấm vào icon sửa của một sinh viên trong danh sách thì hiện ra các thông tin ban đầu có sẵn. Chỉ có thể thay đổi tên sinh viên và ngày tháng năm sinh. Bấm lưu sẽ thông báo lưu thành công và trả lại danh sách sinh viên sau khi được cập nhật.

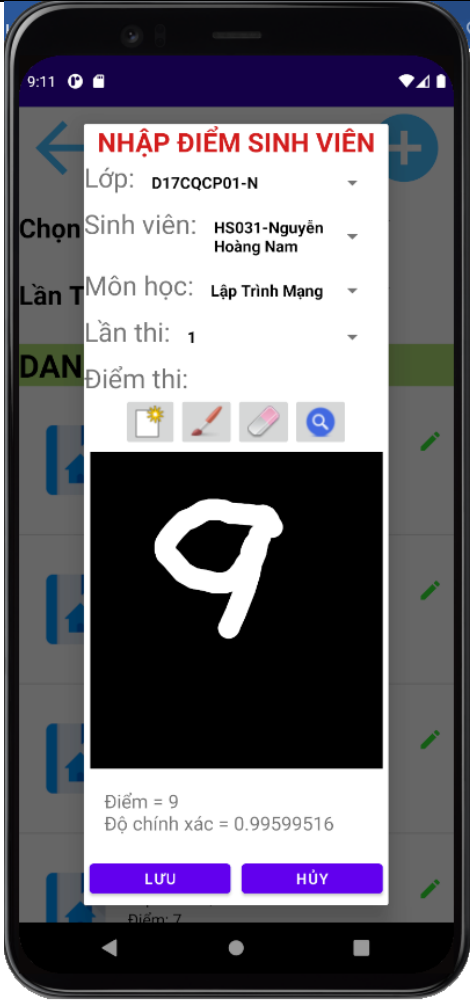
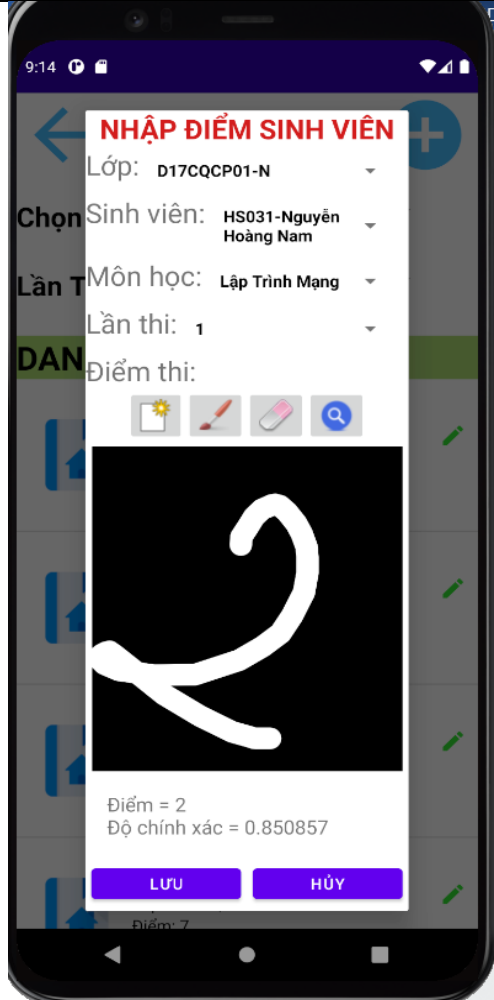


Bấm vào icon xóa: hiện ra hộp thoại để xác nhận có chắc chắn xóa sinh viên hay không? Bấm có thì kiểm tra thông tin và xóa nếu bản ghi đó không có liên kết dữ liệu với bảng nào trong database.



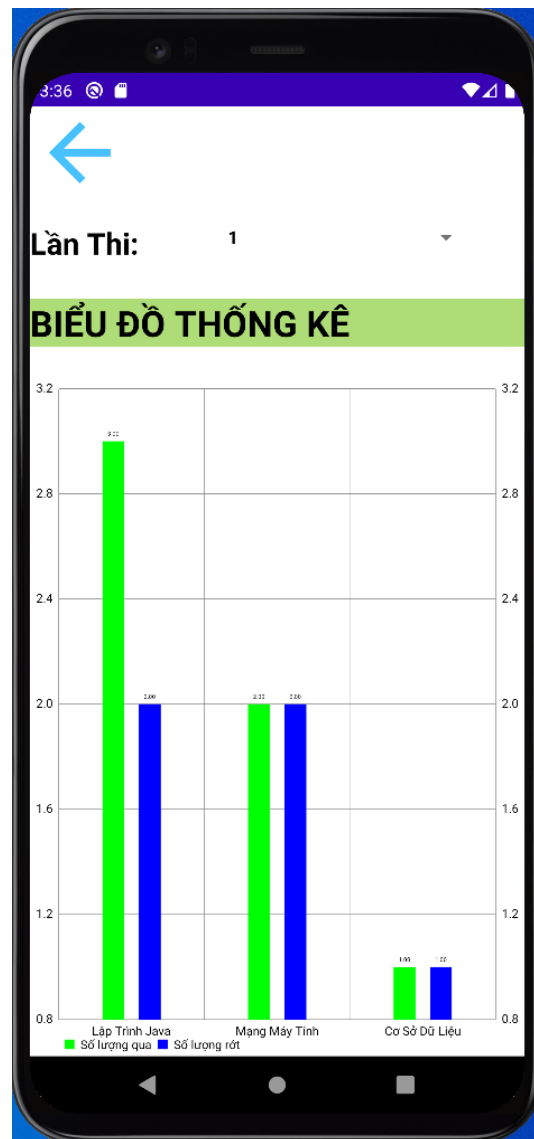
➤ Giao diện nhập điểm

Chọn lớp, chọn sinh viên, chọn môn, chọn lần thi. Sau đó vẽ hình ảnh điểm thi vào và bấm icon tìm kiếm sẽ nhận dạng được điểm sau khi vẽ. Bấm lưu khi đã chắc chắn mọi thuộc tính đều đã nhập. Báo lỗi khi đã tồn tại bản ghi.

<p>Số 9 là mô hình đoán đúng nhiều nhất</p> <ul style="list-style-type: none">- Số dự đoán 9- Độ chính xác gần 100%	<p>Số 2 thì độ chính xác thấp hơn nhưng vẫn đúng</p> <ul style="list-style-type: none">- Số dự đoán 2- Độ chính xác: 85,0857%
	

➤ **Giao diện thống kê**

Bấm vào icon thống kê, in ra số lượng sinh viên qua và rớt của từng môn. Chọn lần thi thì load lại dữ liệu tương ứng với lần thi hiện tại.



KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận

Qua luận văn này, em đã tìm hiểu và nắm khái quát được những kiến thức của mạng neural nhân tạo. Sử dụng mô hình CNN trong công việc xây dựng nhận dạng ký tự số viết tay. Luận văn đã được hoàn thành về mặt cơ bản nội dung đã đề ra. Đó là sự kết hợp giữa xử lý ảnh và lý thuyết nhận dạng nhằm giải quyết một phần bài toán nhận dạng ký tự số viết tay để ứng dụng vào phần quản lý điểm thi.

Đồng thời chương trình đã thực hiện được việc quản lý sinh viên, quản lý lớp học. Chương trình đang bước đầu thử nghiệm và đạt được kết quả thực nghiệm dựa trên 10.000 tập mẫu với độ chính xác trên 98, 85%.

Hạn chế

Tuy đã hoàn thành về nội dung và chương trình nhận dạng nhưng vẫn còn một số hạn chế cần được khắc phục như: Chương trình đang ở mức thử nghiệm, chưa áp dụng vào thực tế, nhận dạng ký tự còn sai nhiều.

Hướng phát triển

Nâng cao hiệu quả chương trình, nhận diện nhiều ký tự cùng lúc.

Phát triển chương trình thành module phần cứng. Có khả năng tương thích với các thiết bị quan sát như camera.

Nghiên cứu theo hướng một ứng dụng cụ thể như: scan văn bản, nhận dạng CMND và các phần mềm sử dụng trong ngân hàng, sử dụng cho các nhà trường cho việc nhập điểm tự động...

PHỤ LỤC

[1] **LeCun:** (8/7/1960) là một nhà khoa học máy tính người Pháp làm việc chủ yếu trong các lĩnh vực máy học, thị giác máy tính, robot di động và khoa học thần kinh tính toán. Ông là Giáo sư Bạc của Viện Khoa học Toán học Courant tại Đại học New York, và Phó Chủ tịch, Trưởng nhóm Khoa học AI tại Facebook.

<https://scholar.google.com/citations?user=WLN3QrAAAAAJ&hl=en>

[2] <https://www.microsoft.com/en-us/research/wp-content/uploads/2003/08/icdar03.pdf>

[3] Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged, Aradiv'ertan'uk tere 1., H-6720 Szeged, Hungary

[4] <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>

[5] https://en.wikipedia.org/wiki/Support_vector_machine

TÀI LIỆU THAM KHẢO

1. <https://engmrk.com/lenet-5-a-classic-cnn-architecture/>
2. <https://dlapplications.github.io/2018-07-06-CNN/>
3. https://d2l.ai/vn.com/chapter_convolutional-neural-networks/index_vn.html
4. <https://machinelearningcoban.com/2016/12/26/introduce/>
5. <https://developer.android.com/codelabs/digit-classifier-tflite?hl=Es#5>