TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA ĐIỆN ĐIỆN TỬ

BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP

**NHẬN DIỆN CHỮ VIẾT TAY**

**SỬ DỤNG MẠNG NƠ-RON NHÂN TẠO**

**NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ - TRUYỀN THÔNG**

Sinh viên: **NGUYỄN THANH TUYẾT HÂN**

MSSV: 15141152

**CAO VĂN CẢNH**

MSSV: 15141104

TP. HỒ CHÍ MINH – 06/2019

**MỤC LỤC**

[Chương 1 Giới thiệu 1](#_Toc407486277)

[1.1 GIỚI THIỆU 1](#_Toc407486278)

[1.2 MỤC TIÊU ĐỀ TÀI 1](#_Toc407486279)

[1.3 GIỚI HẠN ĐỀ TÀI 2](#_Toc407486279)

[1.4 PHƯƠNG PHÁP NGHIÊN CỨU 2](#_Toc407486279)

[1.5 ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU 2](#_Toc407486279)

[Chương 2 CƠ SỞ LÝ THUYẾT 4](#_Toc407486280)

[2.1 TÌNH HÌNH NGHIÊN CỨU 4](#_Toc407486281)

[2.2 XỬ LÝ ẢNH 4](#_Toc407486289)

[2.2.1 Khái niệm cơ bản về ảnh số 5](#_Toc407486290)

[2.2.2 Các giai đoạn của quá trình xử lý ảnh 7](#_Toc407486291)

[2.3 DEEP LEARNING](#_Toc407486297) 8

[2.3.1 Giới thiệu](#_Toc407486290) 8

[2.3.2 Mạng nơ-ron nhân tạo 8](#_Toc407486290)

[Chương 3 THIẾT KẾ HỆ THỐNG 12](#_Toc407486298)

[3.1 YÊU CẦU HỆ THỐNG 12](#_Toc407486299)

[3.2 SƠ ĐỒ KHỐI HỆ THỐNG 12](#_Toc407486300)

[3.2.1 Ngõ vào 12](#_Toc407486290)

[3.2.2 Tiền xử lý](#_Toc407486290) 13

[3.2.3 Mạng nơ-ron nhân tạo](#_Toc407486290) 14

[3.2.4 Kết quả 1](#_Toc407486290)8

[3.3 LƯU ĐỒ HỆ THỐNG 19](#_Toc407486301)

[3.3.1 Lưu đồ huấn luyện 19](#_Toc407486290)

[3.2.3 Lưu đồ chương trình 20](#_Toc407486290)

[Chương 4 kết quả 21](#_Toc407486298)

[4.1 GIAO DIỆN CHƯƠNG TRÌNH 21](#_Toc407486299)

[4.2 HOẠT ĐỘNG CỦA HỆ THỐNG 21](#_Toc407486300)

[4.3 KẾT QUẢ HOẠT ĐỘNG 22](#_Toc407486300)

[Chương 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 30](#_Toc407486298)

[5.1 KẾT LUẬN 30](#_Toc407486299)

[5.2 HƯỚNG PHÁT TRIỂN 30](#_Toc407486300)

[TÀI LIỆU THAM KHẢO 29](#_Toc407486302)

# Giới thiệu

## Giới thiệu

Nhận diện là bài toán xuất hiện cách đây khá lâu và vẫn luôn thu hút nhiều sự quan tâm, nghiên cứu. Đặc biệt là trong vài thập niên gần đây, do sự thúc đẩy của quá trình tin học hóa trong mọi lĩnh vực, bài toán nhận dạng không còn dừng lại ở mức độ nghiên cứu nữa mà nó trở thành một lĩnh vực để áp dụng vào thực tế. 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 chữ. Trong số này, nhận dạng chữ là bài toán được quan tâm rất nhiều và cũng đã đạt được nhiều thành tựu rực rỡ. Các ứng dụng có ý nghĩa thực tế lớn có thể kể đến như: nhận dạng chữ in dùng trong quá trình sao lưu sách báo trong thư viện, nhận dạng chữ viết tay dùng trong việc phân loại thư ở bưu điện, thanh toán tiền trong nhà bang và lập thư viện sách cho người mù (ứng dụng này có ý nghĩa: scan sách bình thường, sau đó cho máy tính nhận dạng và trả về dạng tài liệu mà người mù có thể đọc được).

Xuất phát từ yêu cầu thực tế, đang rất cần có những nghiên cứu về vấn đề này. Chính vì vậy, chúng tôi đã chọn đề tài nhận dạng ký tự viết tay làm đồ án tốt nghiệp với mong muốn phần nào áp dụng vào bài toán thực tế.

## mục Tiêu đề tài

* Xây dựng một mạng nơ-ron nhân tạo có khả năng nhận diện được chữ viết tay với độ chính xác tốt.
* Thu nhận ảnh từ camera và đưa vào hệ thống để xử lý. Sau đó, tách ảnh văn bản thành từng ảnh kí tự theo thứ tự từ trái sang phải. Sau khi đưa vào mạng nơ-ron, hệ thống sẽ tiến hành nhận dạng chữ đã viết và xuất ra kết quả trên cửa sổ hệ thống.
* Thiết kế một giao diện hoàn chỉnh, thuận tiện cho việc sử dụng.

## Giới hạn đề tài

* Hệ thống không nhận dạng được số đếm cũng như không phát hiện được khoảng trắng giữa các từ.
* Hệ thống phải được chạy trên máy tính có cài sẵn ngôn ngữ Python, Pycharm và một số thư viện cần thiết.
* Số lượng ký tự được nhận dạng cùng một lúc không thể trên 22, các ký tự không được viết dính liền nhau.
* Khoảng cách giữa camera và ký tự cần nhận dạng từ 15cm đến 60cm, ký tự cần nhận dạng có chiều cao từ 4cm đến 15cm, chiều rộng chiếm 2/3 chiều dài kí tự, độ dày tối của ký tự là từ khoảng 0.3cm đến 0.5cm, với độ dày 0.5cm kí tự cần nhận dạng có độ chính xác cao hơn..
* Ảnh sau quá trình tiền xử lý ảnh phải có phông nền trắng và chữ màu đen hoặc xanh. Ảnh không được chứa bất kì thành phần khác ngoài hai thành phần trên.
* Camera phải có chất lượng tốt, ánh sáng trong phòng vừa đủ và đều.

## phương pháp nghiên cứu

* Tìm hiểu về Deep Learning (Học Sâu) và Artificial Neural Network (Mạng nơ-ron nhân tạo).
* Tìm hiểu về ngôn ngữ Python, thư viện OpenCV, Tensorflow và Keras.
* Tìm hiểu về tiền xử lý ảnh và các phương pháp nhận dạng.
* Xây dựng chương trình trên máy tính cá nhân.

## đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

* Các phương pháp xử lí ảnh đầu vào.
* Các phương pháp nhận diện chữ viết tay.
* Ngôn ngữ lập trình Python, thư viện OpenCV, Tensorflow, Keras.

Phạm vi nghiên cứu:

* Việc nhận dạng chữ viết tay phải thỏa mãn các điều kiện sau:
  + Chữ phải được viết trên cùng một hàng, không viết quá dính liền nhau.
  + Chất lượng camera tốt.
  + Ánh sáng tốt.

# cơ sở lý thuyết

## Tình hình nghiên cứu

Trong cuộc sống này nay, công nghệ được ứng dụng trong nhiều lĩnh vực từ nông nghiệp cho đến công nghiệp, nhờ sự phát triển mạnh mẽ của ngành công nghệ thông tin, ngành công nghiệp máy tính cũng như sự xuất hiện của Big Data, một số trí tuệ nhân tạo (AI) đã được ra đời. AI là trí tuệ do con người phát triển nhằm mục tiêu giúp máy tính có thể nhận biết và có một số hành vi thông minh như con người. Các ứng dụng của AI hiện nay được nhiều công ty công nghệ áp dụng vào sản phẩm của họ nhằm nâng cao đời sống của con người hiện nay. Google đã xây dựng được một hệ thống có khả năng nhận diện chữ viết cực kì chính xác ở khá nhiều ngôn ngữ khác nhau từ tiếng anh, trung, nhật...

## Xử lý ảnh

Đầu vào của quá trình xử lý ảnh là các ảnh gốc ban đầu, thu được qua máy ảnh, thiết bị ghi hình hay thiết bị thu nhận hình ảnh khác. Ảnh ban đầu thường có chất lượng thấp (do ảnh hưởng của nhiễu, bị lệch so với ảnh gốc một góc bất kỳ, bị đứt nét). Quá trình tiền xử lý ảnh để nâng cao chất lượng ảnh đầu vào trước khi đưa vào nhận dạng là cực kì cần thiết. Quá trình này bao gồm công đoạn khôi phục ảnh và tăng cường ảnh.

Khôi phục ảnh nhằm mục đích loại bỏ hay làm giảm tối thiểu các ảnh hưởng của môi trường bên ngoài lên ảnh thu nhận được.

Tăng cường ảnh là một công đoạn quan trọng, tạo tiền đề cho xử lý ảnh. Tăng cường ảnh không phải làm tăng lượng thông tin trong ảnh mà là làm nỗi bật những đặc trưng của ảnh giúp cho công việc xử lý phía sau được hiệu quả hơn.

### Khái niệm cơ bản về ảnh số

Ảnh số là tập hợp hữu hạn các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật. Số điểm ảnh xác định độ phân giải của ảnh. Ảnh có độ phân giải càng cao thì càng thể hiện rõ nét các đặt điểm của tấm hình càng làm cho tấm ảnh trở nên thực và sắc nét hơn. [1]

Điểm ảnh: là một phần tử của ảnh số tại toạ độ (x, y) với độ xám hoặc màu nhất định. Kích thước và khoảng cách giữa các điểm ảnh đó được chọn thích hợp sao cho mắt người cảm nhận sự liên tục về không gian và mức xám (hoặc màu) của ảnh số gần như ảnh thật. Mỗi phần tử trong ma trận được gọi là một phần tử ảnh. [1]

Mức xám của ảnh: Là kết quả của sự biến đổi tương ứng 1 giá trị độ sáng của 1 điểm ảnh với 1 giá trị nguyên dương. Mức xám dùng 1 byte biểu diễn: 28 =256 mức, tức là từ 0 đến 255. [1]

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. [2]

### Các giai đoạn xử lý ảnh

Quá trình xử lý ảnh được thể hiện như hình 2.5

**Thu nhận ảnh**

**Tiền**

**xử lý**

**Phân**

**đoạn**

**Biểu diễn ảnh**

**Nhận dạng và nội suy**

**Hình 2.1:** Quá trình xử lý ảnh

### *2.2.2.1 Thu nhận ảnh*

Ảnh thu được từ nhiều nguồn khác nhau như máy ảnh, máy quay phim, máy quét, các thiết bị ghi hình khác.

### *2.2.2.2 Tiền xử lý*

Sử dụng các kỹ thuật xử lý ảnh để nâng cao chất lượng ảnh. Mục đích là để điều chỉnh độ sáng nhằm khắc phục hậu quả của việc chiếu sáng không đều, giảm nhỏ thành phần nhiễu của ảnh tức là các đối tượng xuất hiện ngoài ý muốn, hiệu chỉnh giá trị độ sáng giữa nền và đối tượng, chuẩn hoá độ lớn, màu, dạng của ảnh. Ảnh thu thập được thường có nhiễu và cần loại bỏ nhiễu hay ảnh thu được không sắc nét, bị mờ cần làm rõ các chi tiết trước khi đưa vào xử lý.

* Chuyển xám ảnh

Đơn vị tế bào của ảnh số là pixel. Đối với ảnh màu từng pixel sẽ mang thông tin của ba màu cơ bản tạo ra bản màu khả kiến là Đỏ (R), Xanh lá (G) và Xanh biển (B). Trong mỗi pixel của ảnh màu, ba màu cơ bản R, G và B được bố trí sát nhau và có cường độ sáng khác nhau. Đối với ảnh xám, thông thường mỗi pixel mang thông tin của 256 mức xám (tương ứng với tám bit) như vậy ảnh xám hoàn toàn có thể tái hiện đầy đủ cấu trúc của một ảnh màu tương ứng thông qua tám mặt phẳng bit theo độ xám.

Trong hầu hết quá trình xử lý ảnh, cấu trúc của ảnh được quan tâm nhiều và ảnh hưởng của yếu tố màu sắc thì được bỏ qua. Do đó bước chuyển từ ảnh màu thành ảnh xám là một công đoạn phổ biến trong các quá trình xử lý ảnh vì nó làm tăng tốc độ xử lý là giảm mức độ phức tạp của các thuật toán trên ảnh.

Công thức chuyển các thông số giá trị màu của một pixel thành mức xám tương ứng như sau:

G = α.CR + β.CG + δ.CB (2.1)

Trong đó các giá trị CR, CG và CB lần lượt là các mức độ màu Đỏ, Xanh và Xanh biển của pixel màu.

* Phương pháp lọc nhiễu Gaussian Blur

Bộ lọc Gaussian là bộ lọc thông cao (chỉ giữ lại thành phần tần số thấp), một

cách trực quan hơn đây được xem là phương pháp làm mờ mịn giống như hiệu ứng hình ảnh được đặt dưới một lớp màn trong suốt bị mờ. Gaussian blur là một bộ lọc làm mờ ảnh, sử dụng lý thuyết hàm Gaussian để tính toán việc chuyển đổi mỗi pixel của hình, phương trình hàm Gaussian dùng trong không hai chiều:

(2.2)

Trong đó x và y là tọa độ theo hai trục đứng và ngang còn σ là giá trị quyết định độ lệch giữa các điểm trên bề mặt Gaussian. [3]

* Nhị phân hóa

Quá trình này chuyển từ ảnh màu, ảnh đa cấp xám sang ảnh nhị phân (ảnh hai cấp xám, ảnh đen trắng). Thuật toán phân ngưỡng cài đặt ở đây sử dụng hàm phân ngưỡng:

(2.3)

Trong đó, g(x,y) là giá trị điểm ảnh ở vị trí (x,y) của ảnh nguồn, g’(x,y) là giá trị điểm ảnh tương ứng ở vị trí (x,y) của ảnh đích. T là giá trị ngưỡng.

Giá trị cụ thể của ngưỡng phụ thuộc vào từng ảnh, vùng ảnh đầu vào đang xét và không thể lấy cố định. [4]

* Phương pháp giãn ảnh

Là quá trình chập của những bức ảnh nhằm loại bỏ điểm đen bị vây bởi các điểm trắng. Trong kỹ thuật này một ma trận (N+1)x(N+1) được quét trên toàn ảnh và thực hiện so sánh một pixel của ảnh với (N+1)x2-1 điểm lân cận.

(2.4)

### Trong đó A là ma trận điểm ảnh của ảnh nhị phân, B là phần tử cấu trúc.

### *2.2.2.3 Phân đoạn ảnh*

Là quá trình phân chia nội dung các đối tượng cần khảo sát ra khỏi ảnh, phân chia các đối tượng tiếp giáp nhau, phân tách các đối tượng riêng biệt thành các đối tượng con.

*2.2.2.4 Biểu diễn ảnh*

Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lận cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được.

*2.2.2.5 Nhận dạng và nội suy ảnh*

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản là nhận dạng theo tham số và nhận dạng theo cấu trúc. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng.

## deep learning

### Giới thiệu

Học sâu là một tên gọi dễ hiểu hơn của mạng nơ-ron nhân tạo. Chúng được sử dụng để giúp robot học tập tốt hơn. Kỹ thuật này lấy ý tưởng từ bộ não sinh học con người. Một hệ thần kinh của con người bao gồm nhiều nơ-ron, trong đó, một nơ-ron sẽ nhận thông tin từ một số nơ-ron xử lý tín hiệu rồi truyền cho nơ-ron khác liên kết với nó.

### Mạng nơ-ron nhân tạo

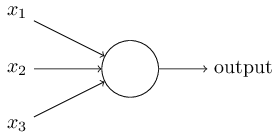
### *Perceptron cơ bản*

Một mạng nơ-ron nhân tạo được cấu thành bởi các nơ-ron đơn lẻ được gọi là các perceptron. Nơ-ron nhân tạo được lấy cảm hứng từ nơ-ron sinh học như hình mô tả bên dưới:



**Hình 2.2:** Nơ-ron sinh học [8]

Như hình trên, một nơ-ron có thể nhận nhiều đầu vào và cho ra một kết quả đầu duy nhất. Mô hình của perceptron cũng tương tự:



**Hình 2.12:** Mô hình nơ-ron nhân tạo [8]

Một perceptron sẽ nhận một hoặc nhiều đầu vào (kí hiệu là x) dưới dạng nhị phân và cho ra một kết quả (kí hiệu là o) duy nhất cũng dưới dạng nhị phân. Tầm ảnh hưởng của mỗi đầu vào khác nhau được quyết định bởi tham số trọng lượng (weight kí hiệu là w) của nó. Kết quả đầu ra được quyết định dựa vào một ngưỡng b (bias) nào đó:

o = (2.5)

### *Nơ-ron với Hàm kích hoạt (Activation Function)*

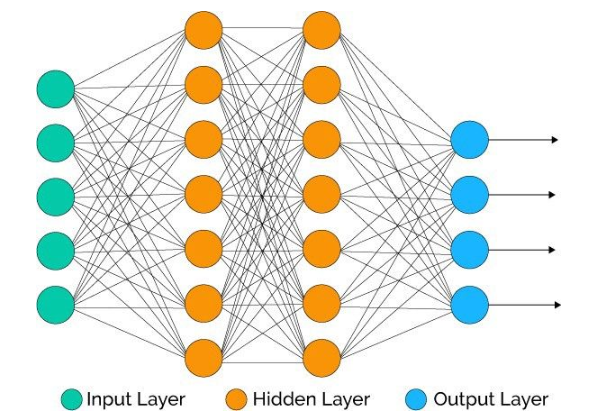
Với đầu vào và ra dạng nhị phân, hệ thống sẽ không có độ linh hoạt nên để tăng độ linh động thì giá trị của đầu vào và đầu ra sẽ ở trong khoảng [0,1]. Đề làm được điều này đầu ra sẽ được quyết định bởi một hàm kích hoạt f(z). Hàm này có thể là hàm sigmoid, tanh, hoặc ReLU,…

### *Kiến trúc của một mạng nơ-ron nhân tạo*

Mạng nơ-ron nhân tạo là sự kết hợp của các nơ-ron đơn lẻ. Một mạng nơ-ron nhân tạo sẽ có 3 kiểu tầng:

* Tầng vào (input layer): thể hiện đầu vào của mạng.
* Tầng ra (output layer): thể hiện đầu ra của mạng.
* Tầng ẩn (hidden layer): nằm giữa tầng vào và tầng ra thực hiện chức năng suy luận logic của mạng.

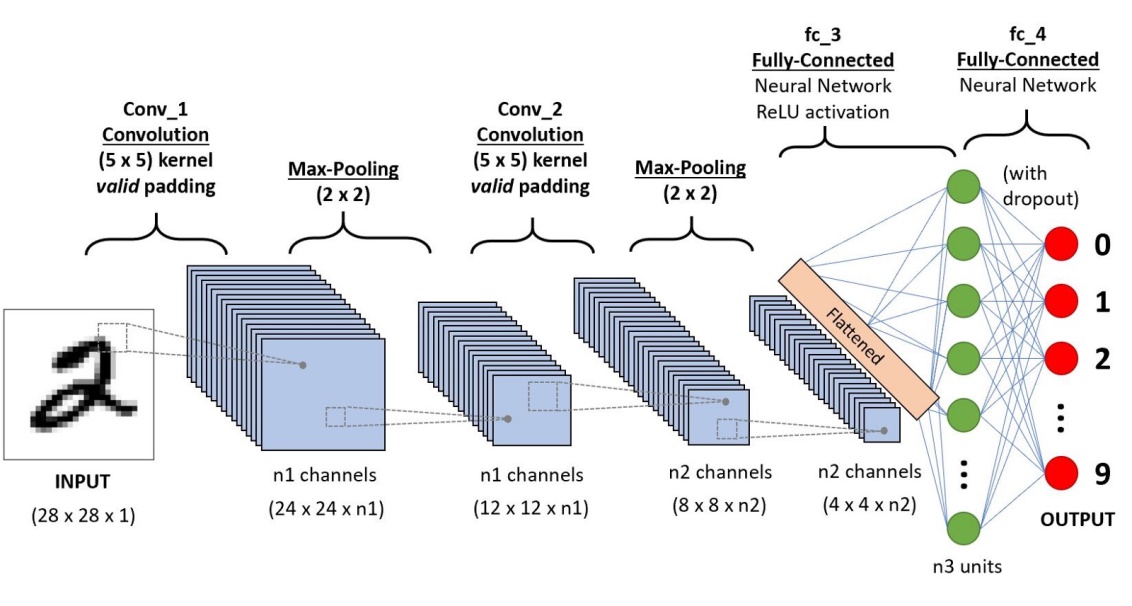
Mỗi mạng nơ-ron chỉ có tầng vào và 1 tầng ra nhưng có thể có nhiều tầng ẩn. Số lượng nơ-ron mỗi tầng khác nhau tùy thuộc vào công việc mà mạng đó thực hiện. Mỗi nơ-ron trong một lớp thường được liên kết với từng nơ-ron ở lớp tiếp theo. Mạng này còn được gọi là mạng kết nối đầy đủ (full-connected network).[10]



**Hình 2.3:** Mạng nơ-ron kết nối đầy đủ

Để một mạng nơ-ron nhận biết được quan hệ giữa đầu vào và đầu ra cần một quá trình huấn luyện. Trong quá trình huấn luyện một số lượng lớn dữ liệu sẽ được đưa vào tầng vào, qua một quá trình tính toán mạng sẽ đưa ra một kết quả đầu ra (gọi là prediction). Qua việc tính toán sự khác biệt (loss) giữa kết quả đầu ra và kết quả mong muốn (gọi là expectation) có sẵn, sau đó điều chỉnh tham số trọng lượng cũng như ngưỡng giữa các lớp nơ-ron, sự khác biệt của hai kết quả trên sẽ ngày càng nhỏ và hệ thống sẽ ngày càng chính xác. Quá trình điều chỉnh ngược lại tham số trọng lượng giữa các nơ-ron gọi là quá trình backpropagation.

Với một bài toán nhận diện hình ảnh, nếu chỉ sử dụng mạng nơ-ron truyền thống như trên thì hệ thống sẽ chỉ phân biệt được những bức ảnh đơn giản với độ chính xác có thể chấp nhận được. Khi kích thước của bức ảnh tăng hoặc người nghiên cứu muốn tăng độ chính xác của hệ thống, mạng nơ-ron tích chập cho ra kết quả tốt hơn nhiều so với mạng nơ-ron được kết nối đầy đủ.



**Hình 2.4:** Mạng nơ-ron tích chập

Mạng nơ-ron tích chập nhận diện về không gian cục bộ tốt hơn mạng nơ-ron truyền thống vì thông thường không phải nơ-ron nào của lớp trước cũng được kết nối với nơ-ron của tầng tiếp theo. Trong mạng nơ-ron tích giá trị của mỗi nơ-ron của tầng tiếp theo là kết quả tích chập của tầng trước đó, nhờ đó kết nối cục bộ giữa các pixel được sinh ra. Ngoài ra nhờ vào lớp Tổng hợp (Pooling), thông tin có trong các lớp tích chập sẽ được chắt lọc và tổng hợp. [11]

# thiết kế hệ thống

## yêu cầu hệ thống

* Sử dụng webcam được kết nối bằng cáp với máy tính để chụp hình những kí tự cần được nhận diện.
* Khối tiền xử lý phải tách đầy đủ từng kí tự có trong hình theo thứ tự từ trái sang phải, sau đó từng ảnh phải được đưa về kích thước chuẩn 28x28 pixel để phù hợp với đầu vào của mạng nơ-ron.
* Mạng nơ-ron phải có độ chính xác trên 85%.

## Sơ đồ khối hệ thống

CAMERA

MẠNG NƠ-RON NHÂN TẠO

KẾT QUẢ

PHÂN ĐOẠN ẢNH

HÌNH ẢNH CÓ SẴN

CHUẨN HÓA ẢNH

KHỐI XỬ LÝ ẢNH ĐẦU VÀO

**Hình 3.1:** Sơ đồ khối hệ thống

### Ngõ vào

Gồm hình ảnh từ camera hoặc hình ảnh có sẵn trong máy tính.

Camera mà đề tài sử dụng được kết nối với máy tính để chụp ảnh chứa các ký tự. Trên thị trường hiện nay có hai loại webcam phổ biến đó là Webcam có dây và Webcam không dây, một số dòng webcam có chất lượng tốt được nhiều ngươi ưa chuộng như là Logitech C270HD, Logitech B525 HD, Microsoft Lifecam Cinema, Microsoft Lifecam Studio, Genius Facecam 1000x, …

Để tăng tiến độ cho quá trình xử lý ảnh thì ảnh đầu vào cũng cần một số điều kiện dựa trên tính chất của chữ cái cần xác định. Chúng ta cần dựa vào tính chất màu của kí tự (màu đen), nền chứa kí tự là màu trắng. Việc ảnh chứa một số loại thành phần màu khác, chúng ta có thể loại bỏ bằng cách xử lý ảnh xám và chuyển ảnh về ảnh trắng đen.

Nhóm đã tiến hành sử dụng một chân máy ảnh để cố định webcam nhằm giảm thiểu độ rung khi lấy ảnh đầu vào.

Ảnh đầu vào là ảnh mang kí tự đen nền trắng, việc giảm thiểu nhiễu là điều rất cần trong xử lý ảnh, nên nhóm cũng sử dụng một chân máy ảnh khác để cố định ảnh đầu vào.

Khoảng cách giữa camera và ký tự cần nhận dạng từ 15cm đến 60cm, ký tự cần nhận dạng có chiều cao từ 4cm đến 15cm, chiều rộng chiếm 2/3 chiều dài kí tự, độ dày tối của ký tự là từ khoảng 0.3cm đến 0.5cm, với độ dày 0.5cm kí tự cần nhận dạng có độ chính xác cao hơn.

### Tiền xử lý

Phần tiền xử lý bao gồm 3 khối: Khối xử lý ảnh đầu vào, khối phân đoạn ảnh và khối chuẩn hóa ảnh.

Phần này có chức năng xử lý đối với ảnh đầu vào thô, ảnh đầu vào có kích thước khá lớn sẽ ảnh hưởng đến quá trình huấn luyện và kiểm tra nên nhóm quyết định chuẩn hóa kích thước để phù hợp cho quá trình nhận dạng. Quá trình xử lý ảnh diễn ra như sau:

Ảnh sau khi đọc vào hệ thống tiến hành chuyển ảnh màu BGR sang trường màu xám bằng hàm cv2.cvtColor().

Điều này sẽ dễ dàng hơn cho việc lọc nhiễu Gaussian bằng hàm thông dụng cv2.GaussianBlur().

Ảnh sau khi chuyển xám và lọc nhiễu thì cần xác được ngưỡng mức xám để chuyển đổi sang giá trị nhị phân. Vì kí tự được nhận diện là màu đen nên sẽ có mức xám thấp. Để chuyển sang ảnh nhị phân, hàm cv2.threshold được dùng để lấy ngưỡng ảnh, nếu giá trị điểm ảnh cao hơn giá trị ngưỡng thì nó được gán giá trị 0 (màu trắng), ngược lại giá trị điểm ảnh nhỏ hơn giá trị ngưỡng thì nó được gán giá trị 1 (màu đen). Ở hàm cv2.threshold, hình ảnh sau khi được làm mờ sẽ được áp dụng threshold = 130, các pixel có cường độ lớn hơn 130 đều được set bằng 255, ngược lại thì được set bằng 0. Ngoài ra, ảnh có thể được inverse thresholding với hàm cv2.THRESH\_BINARY\_INV (hình Threshold Binary Inverse).

Tiếp đến hệ thống sẽ sử dụng hàm cv2.dilate để thêm pixel vào biên của kí tự, với bộ lọc là ma trận 5x1 duyệt qua tất cả các pixel của ảnh nhị phân để tính toán.

Bước tiếp theo hệ thống sử dụng hàm cv2.findContour để tìm tất cả các đường biên của kí tự, sau đó sử dụng hàm sorted để sắp xếp các đường biên từ trái sang phải rồi khôi phục lại các chi tiết của kí tự bằng cách dùng hàm tạo hình chữ nhật cv2.rectangle theo đường bao (contour) đã được tìm thấy trước đó.

Sau khi tạo hình chữ nhật theo đường bao hệ thống tiến hành tách kí tự khỏi ảnh văn bản, ở đây sẽ dùng hàm RoI để tách từng kí tự riêng biệt. Kí tự sau khi tách khỏi ảnh văn bản sẽ được nhị phân hóa lần 2 nhằm đảo ngược chữ trắng nền đen sang chữ đen nền trắng với ngưỡng được chọn là 127 thông qua hàm cv2.THRESH\_BINARY\_INV.

Để thuận lợi cho mạng nơ-ron nhận diện chính xác kí tự, nhóm tiến hành chuẩn hóa kích thước từng kí tự về ảnh có chiều rộng và chiều cao là 28pixel, việc chuẩn hóa thông qua hàm cv2.resize(), ảnh sau chuẩn hóa sẽ được lưu đến một thư mục chọn trước để thuận tiện cho mạng nơ-ron trích xuất hình ảnh.

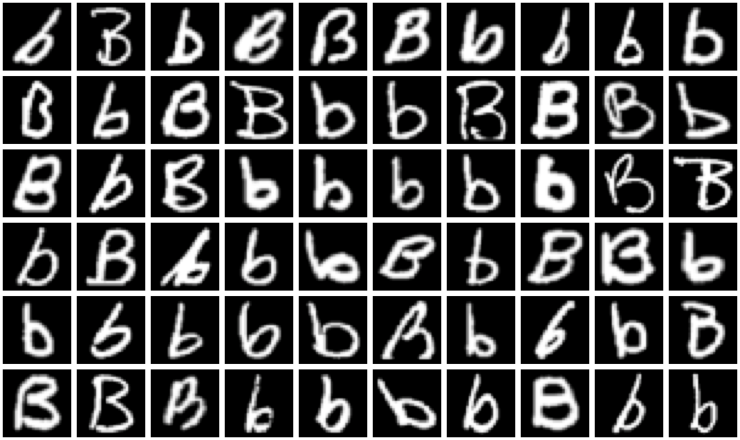
### Mạng nơ-ron nhân tạo

Trong lĩnh vực học máy có các phương pháp học sau:

* Học có giám sát (supervised learning).
* Học không có giám sát (unsupervied learning).
* Học bán giám sát (semi-supervised learning).
* Học tăng cường (reinforcement learning).

Học có giám sát dùng thường dùng cho những bài toán phân lớp. Học không có giám sát thường áp dụng cho những bài toán phân cụm. Học bán giám sát là sự pha trộn của hai phương pháp trên, thường được áp dụng khi có dữ liệu lớn nhưng không đồng nhất, lúc được dán nhãn lúc không. Học tăng cường dùng để áp dụng cho những bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất, chủ yếu được áp dụng vào Lý Thuyết Trò Chơi.[13]

Vì mục đích của đề tài là nhận diện kí tự viết tay nên học có giám sát là phù hợp nhất. Với học có giám sát thì một tập dữ liệu chữ viết tay có phân loại và dán nhãn sẵn là không thể thiếu. EMNIST là một cơ sở dữ liệu lớn bao gồm các chữ viết tay và thường được sử dụng để đào tạo các hệ thống xử lý hình ảnh. Bộ cơ sở dữ liệu này có nguồn gốc từ NIST Special Database19 và được chuyển những bức ảnh có kích cỡ là 28x28 pixel. Có sáu bộ dữ liệu khác nhau trong tập dữ liệu này.



**Hình 3.2:** Tập dữ liệu EMNIST

Vì đề tài là nhận dạng chữ viết tay nên tập dữ liệu EMNIST Letters là phù hợp nhất.

*Thiết kế mạng nơ-ron truyền thống:*

Bước đầu tiên là xác định số lượng nơ-ron mỗi tầng cũng như là số lượng tầng ẩn cần thiết. Theo tập dữ liệu EMNIST, mỗi hình đều có kích cỡ là 28x28 pixel do đó số lượng nơ-ron của tầng vào bắt buộc là 28 x 28 = 784 nơ-ron. Tập dữ liệu EMNIST Letters được gắn nhãn từ 1 đến 26 tượng trưng cho kí tự từ A đến Z trong bảng chữ cái. Tuy nhiên trong lập trình, mảng luôn được đánh số từ 0 nên số lượng nơ-ron đầu ra phải là 27 (mảng được đánh số từ 0 đến 26, nơ-ron số 0 là nơ-ron dư).

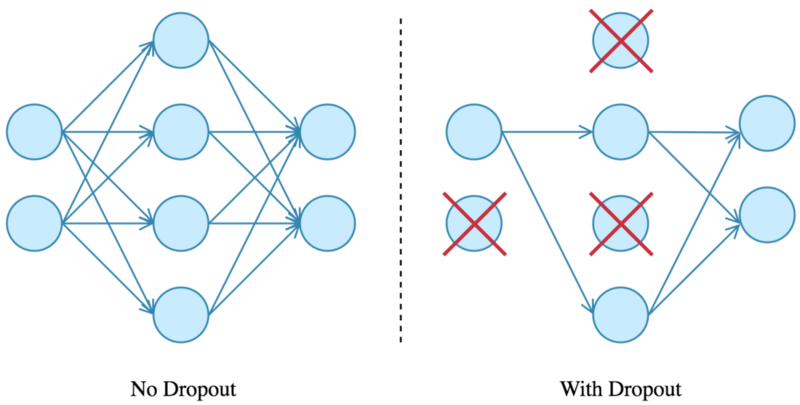
Số lượng Tầng ẩn thường được dựa vào độ phức tạp của bài toán cần nhận dạng. Đối với bài toán nhận dạng chữ viết thì từ 2 đến 3 Tầng ẩn là phù hợp nhất vì khi số lượng tầng ẩn tăng cao thì số lượng nơ-ron cũng tăng cao. Khi số lượng nơ-ron quá cao, hệ thống sẽ có khả năng ghi nhớ nhiều hơn là học hỏi, việc này gây ra vấn đề over-fitting (trọng số quá khớp). Việc này có thể nhận biết khi có sự khác biệt cực lớp giữa độ chính xác trong quá trình huấn luyện và độ chính xác trong quá trình kiểm tra.[14]

Số lượng nơ-ron mỗi tầng, cũng phải được chọn một cách cẩn thận bởi vì việc này có thể dễ dàng gây ra vấn đề trọng số quá khớp và một số vấn đề khác. Đầu tiên, khi số lượng nơ-ron quá nhiều, dòi hỏi dữ liệu để huấn luyện cũng phải lớn để huấn luyện hết tất cả nơ-ron trong hệ thống. Ngoài ra, việc này cũng làm thời gian cần thiết để huấn luyện tăng lên đáng kể. Tuy nhiên khi sử dụng quá ít nơ-ron thì hệ thống thường sẽ không đủ khả năng để phát hiện đầy đủ những điểm quan trọng của một dữ liệu phức tạp, kết quả là độ chính xác của hệ thống sẽ không cao. [15]

Dưới đây là một số quy tắc để xác định số lượng nơ-ron cần thiết trong Tầng ẩn:

* Số lượng nơ-ron thường nằm giữa kích thước của lớp đầu vào và kích thước lớp đầu ra.
* Số lượng nơ-ron thường bằng 2/3 kích thước lớp đầu vào cộng kích thước lớp đầu ra.
* Số lượng tế bào thần kinh phải nhỏ hơn hai lần kích thước đầu vào.

Từ đây có thể xác định số lượng nơ-ron trong tầng ẩn nằm giữa 50 và 700. Tuy nhiên, để hệ thống tránh được vấn đề trọng số quá khớp, kỹ thuật Dropout thường được áp dụng trong quá trình huấn luyện. Kỹ thuật này sẽ bỏ qua một số nơ-ron một cách ngẫu nhiên trong một chu kì huấn luyện (còn gọi là epochs). Qua quá trình thử nghiệm, số chu kì huần luyện đối với bài toán này là từ 10 đến 20 chu kì vì sau đó hệ thống không có xu hướng tốt hơn mà vấn đề trọng số quá khớp bắt đầu xảy ra.[15]



**Hình 3.3:** Kỹ thuật Dropout

Dưới đây là một số kết quả của quá trình huấn luyện mà nhóm đã thực hiện:

**Bảng 3.1:** Kết quả một số mạng nơ-ron truyền thống

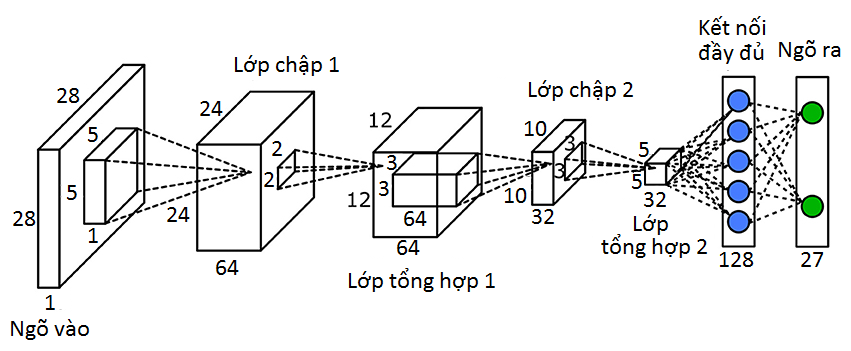
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Số Tầng Ẩn | Số lượng nơ-ron mỗi tầng | Tỉ lệ Dropout | Epochs | Độ chính xác |
| 3 | 600 | 0.4 | 50 | 87,6% |
| 3 | 200 | 0.25 | 20 | 84.8% |
| 3 | 300 | 0.2 | 15 | 86,3% |
| 3 | 230 | 0.2 | 15 | 86,3% |
| 3 | 300 | 0.35 | 10 | 84,4% |
| 3 | 300 | 0.5 | 10 | 82,1% |
| 2 | 512 | 0.2 | 10 | 88,1% |
| 2 | 300 | 0.5 | 10 | 82,8% |

*Thiết kế mạng nơ-ron tích chập:*

Đầu tiên, cần phải xác định kích cỡ của bộ lọc. Vì hình ảnh đầu vào có kích thước khá nhỏ, bộ lọc có kích thước 3x3 hoặc 5x5 sẽ hợp lí hơn vì bộ lọc kích thước lớn hơn (9x9 hoặc 11x11) thường được sử dụng cho những ảnh có số lượng pixel khá lớn. Trong đồ án này, nhóm quyết định kích cỡ bộ lọc sẽ lần lượt là 5x5 ở lớp nơ-ron tích chập thứ nhất và 3x3 ở lớp nơ-ron tích chập thứ hai.

Số lượng bộ lọc trong lớp tích chập thường được sử dụng nhất là từ 32 đến 1024 bộ lọc. Tuy nhiên, vì hình ảnh đầu vào không quá lớn nên nhóm sẽ sử dụng 32 hoặc 64 bộ lọc ở hai lớp tích chập.[16] [17] [18]

Mô hình mạng nơ-ron đề tài sử dụng được thể hiện qua hình và bảng sau đây:



**Hình 3.4:** Cấu trúc mạng nơ-ron

**Bảng 3.2:** Cấu trúc mạng nơ-ron tích chập

|  |  |  |
| --- | --- | --- |
| Lớp | Thông số | Kích thước ngõ ra |
| Ngõ vào | Không | 28x28x1 |
| Tích chập 1 | 64 bộ lọc có kích cỡ [5,5] | 24x24x64 |
| Tổng hợp 1 | Kích cỡ [2x2] | 12x12x64 |
| Tích chập 2 | 32 bộ lọc có kích cỡ [3,3] | 10x10x32 |
| Tổng hợp 2 | Kích cỡ [2x2] | 5x5x32 |
| Dropout | 20% Dropout |  |
| Flatten | Không | 800x1 |
| Kết nối đầy đủ 1 | 128 nơ-ron, 20% dropout | 128x1 |
| Ngõ ra | Hàm kích hoạt là softmax | 27x1 |

Tổng số lượng tham số là 126139 nơ-ron, trong đó 1662 nơ-ron thuộc lớp Tích chập 1, 18464 nơ-ron thuộc lớp Tích chập 2, 102528 nơ-ron thuộc lớp Kết nối đầy đủ và 3483 nơ-ron thược lớp Ngõ ra.

### Kết quả

Kết quả của quá trình huấn luyện sẽ là một mạng nơ-ron có thông số được điều chỉnh một cách tự động bởi máy tính trong quá trình huấn luyện. Mạng nơ-ron này sẽ được lưu trong máy tính với đuôi “.model”.

Kết quả của quá trình nhận diện chữ viết sẽ là một trong 26 chữ cái từ “a” đến “z”. Chữ cái này sẽ được hiển thị trên Textbox của giao diện phần mềm. Kết quả sẽ được sắp xếp đúng thứ tự từ trái qua phải như trên ảnh đầu vào.

## LƯU đồ hệ thống

### Lưu đồ huấn luyện

Đ

S

Kết thúc

Lưu hệ thống đã huấn luyện

Epochs = 10

So sánh giá trị dự đoán và giá trị đã được dán nhãn

Điều chỉnh trọng số của hệ thống

Dự đoán ngõ ra theo ảnh ngõ vào

Bắt Đầu

Tải ảnh huấn luyện từ kho dữ liệu

Khởi tạo thông số hệ thống

**Hình 3.5:** Lưu đồ quá trình huấn luyện

### Lưu đồ chương trình

Bắt đầu

Đọc ảnh

ngõ vào

Xử lý ảnh và chuẩn hóa kích thước

Đưa ảnh vào mạng nơ-ron để nhận dạng

Hiển thị kết quả

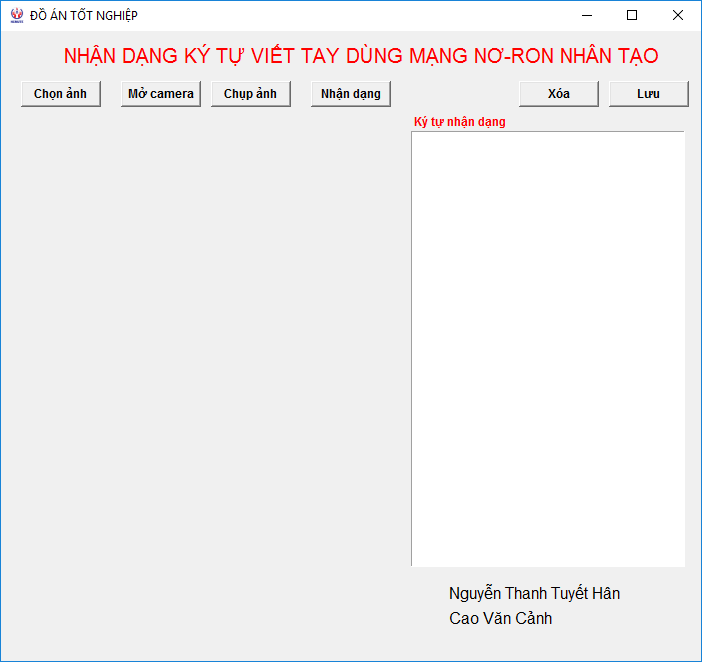
Kết thúc

**Hình 3.6:** Lưu đồ quá trình nhận diện

# kết quả

## GIao diện chương trình

Chương trình nhận dạng ký tự viết tay sử dụng mạng nơ-ron nhân tạo có chức năng là nhận dạng qua hình ảnh tĩnh được người dùng chọn từ thư mục có sẵn hoặc từ wedcam.



**Hình 4.1:** Giao diện chương trình

## hoạt động của hệ thống

Người sử dụng nhấn vào “Chọn ảnh” để nhập ảnh đầu vào. Chương trình sẽ chuyển đến một đường dẫn với thư mục ngẫu nhiên chứa ảnh được chọn từ người dùng.

Sau đó ảnh mà người dùng đã chọn sẽ hiển thị trên Label với vị trí được thiết kế sẵn.

Nếu người dùng muốn chọn ảnh trực tiếp từ camera thì có thể nhấn nút “Mở camera”. Khung camera sẽ xuất hiện lên cửa sổ.

Sau đó người dùng có thể nhấn chụp ảnh để lưu lại ảnh chứa kí tự, ảnh này sau đó sẽ được hiển thị vào Label với vị trí đã được thiết lặp sẵn.

Dù với ảnh đầu vào được chọn từ webcam hay từ file có sẵn, người dùng khi nhấn “Nhận dạng” thì kí tự có trong ảnh sẽ được nhận dạng và hiển thị trên Textbox với vị trí được thiết lặp sẵn.

Người dùng có thể nhấn “Lưu” để lưu lại kết quả nhận dạng với định dạng được hỗ trợ như “.txt” hay “.doc”.

Mặt khác, người dùng có thể xóa đi kết quả vừa nhận dạng khi không muốn lưu kết quả có thể nhấn “Xóa”. Hệ thống sẽ xóa tất cả ký tự có trong textbox khi người dùng nhấn nút này. Ngoài ra, người dùng có thể xóa từ ký tự riêng lẻ bằng bàn phím có sẵn.

## Kết quả hoạt động

Sau đây là kết quả của quá trình huấn luyện, hai bảng 4.1 và 4.2 dưới đây thể hiện ma trận tương quan của hệ thống cũng như số lượng hình nhận dạng chính xác trên 100 mẫu của mỗi ký tự.

**Bảng 4.1:** Kết quả nhận dạng chữ cái từ A đến M

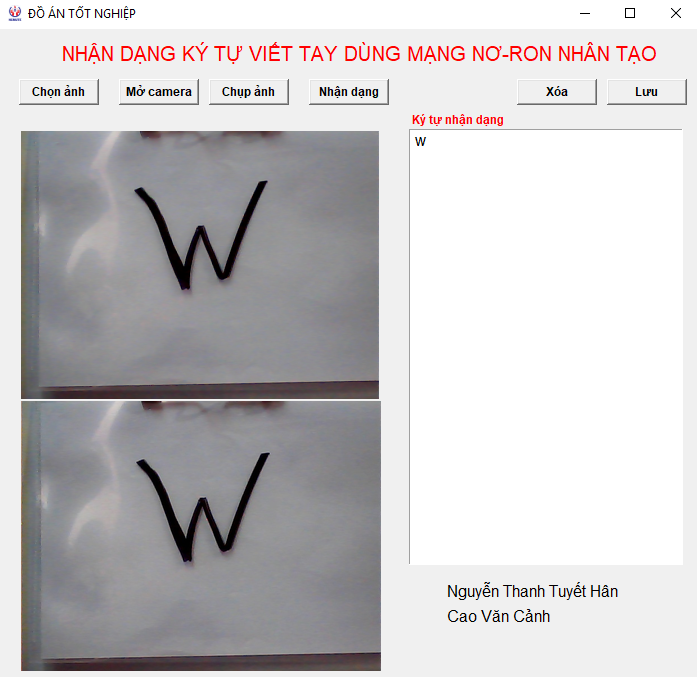
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kết  Quả | Chữ Cái | | | | | | | | | | | | |
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| A | **0.92** | 0.01 | 0 | 0 | 0 | 0 | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | **0.91** | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | **0.95** | 0 | 0.03 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0.01 | 0.02 | 0 | **0.92** | 0 | 0 | 0 | 0 | 0.01 | 0.01 | 0 | 0 | 0 |
| E | 0 | 0.01 | 0.03 | 0 | **0.95** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | **0.92** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0.01 | 0.01 | 0 | 0 | 0 | 0.01 | **0.79** | 0 | 0.01 | 0.01 | 0 | 0 | 0 |
| H | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | **0.95** | 0 | 0 | 0 | 0 | 0.01 |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0.69** | 0.03 | 0 | 0.37 | 0 |
| J | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.02 | **0.93** | 0 | 0 | 0 |
| K | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | **0.96** | 0 | 0 |
| L | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0.26 | 0 | 0 | **0.62** | 0 |
| M | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0.96** |
| N | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0.02 |
| O | 0 | 0 | 0 | 0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 0 | 0 | 0 | 0.02 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Q | 0.04 | 0 | 0 | 0 | 0 | 0 | 0.15 | 0 | 0 | 0 | 0 | 0 | 0 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0.01 | 0 | 0 | 0 |
| T | 0 | 0 | 0 | 0 | 0 | 0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| U | 0.01 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| V | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 |
| X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0.02 | 0 | 0 |
| Y | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Z | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Tổng | 0.99 | 0.99 | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 | 1 | 0.99 | 0.99 | 0.98 | 0.99 | 1 |
| Số mẫu | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Số hình đúng | 97 | 93 | 97 | 93 | 97 | 95 | 82 | 97 | 85 | 95 | 98 | 65 | 97 |
| T.gian  (ms) | 6.1 | 5.3 | 4.5 | 4.3 | 4.3 | 4.3 | 4.4 | 4.4 | 4.4 | 4.3 | 4.4 | 4.2 | 4.2 |

**Bảng 4.2:** Kết quả nhận dạng chữ cái từ N đến Z

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kết  Quả | Chữ Cái | | | | | | | | | | | | |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A | 0 | 0 | 0 | 0.05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0.05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 0.01 | 0 | 0.15 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| H | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| J | 0 | 0 | 0 | 0 | 0 | 0.01 | 0.01 | 0 | 0.01 | 0 | 0 | 0.01 | 0 |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0.02 | 0 | 0 |
| L | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 | 0 |
| M | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| N | **0.94** | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0.02 | 0 | 0 | 0 |
| O | 0 | **0.89** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 0 | 0 | **0.98** | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Q | 0 | 0.01 | 0.01 | **0.75** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0 |
| R | 0 | 0 | 0 | 0 | **0.94** | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | **0.97** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T | 0 | 0 | 0 | 0 | 0 | 0 | **0.97** | 0 | 0 | 0 | 0 | 0 | 0 |
| U | 0.01 | 0.02 | 0 | 0 | 0 | 0 | 0 | **0.93** | 0.01 | 0.01 | 0 | 0 | 0 |
| V | 0.01 | 0.01 | 0 | 0 | 0.01 | 0 | 0 | 0.05 | **0.94** | 0 | 0.01 | 0.03 | 0 |
| W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0.95** | 0 | 0 | 0 |
| X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0.93** | 0 | 0 |
| Y | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.02 | 0 | 0.02 | **0.94** | 0 |
| Z | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0.99** |
| Tổng | 0.99 | 0.99 | 0.99 | 0.99 | 0.96 | 0.99 | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |
| Số mẫu | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Số hình đúng | 95 | 90 | 99 | 82 | 99 | 99 | 100 | 97 | 98 | 95 | 94 | 97 | 100 |
| T.gian  (ms) | 4.2 | 4.2 | 4.3 | 4.4 | 4.4 | 5.6 | 4.5 | 4.5 | 4.4 | 4.3 | 4.3 | 4.2 | 4.3 |

Từ hai bảng trên, có thể thấy hệ thống vẫn còn nhầm lẫm giữa hai chữ cái ‘I’ và ‘L’ vì trong 100 mẫu đầu vào có hình ảnh chữ ‘L’ thì chỉ nhận dạng đúng 65 hình còn 35 hình còn lại cho ra kết quả là chữ ‘I’.

Sau đây là kết quả của một số hình ảnh thực tế, mà hệ thống nhận dạng đúng.

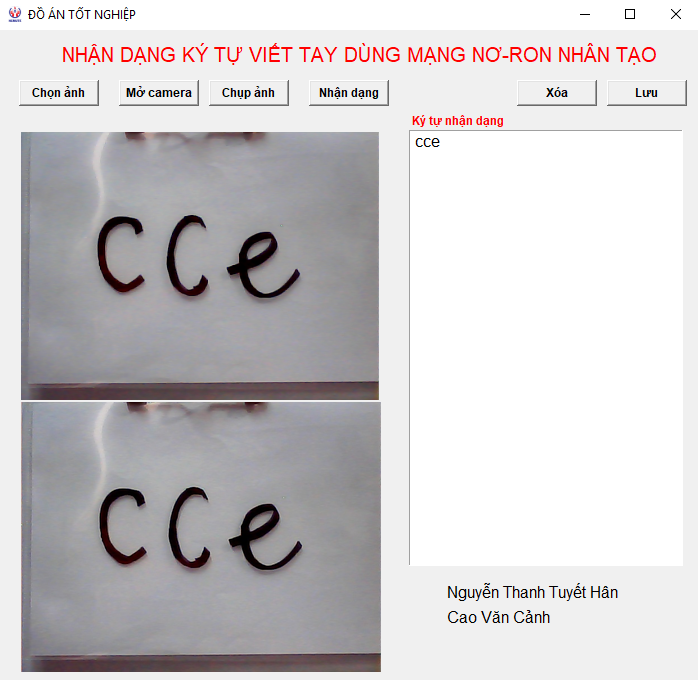
**Hình 4.9:** Một số hình nhận diện đúng

Nhận xét:Kết quả cho thấy hệ thống có khả năng nhận dạng tốt với các kí tự được viết đầy đủ các nét, độ dày kí tự không quá mỏng, nên việc phân loại nền trắng chữ đen trong quá trình xử lý ảnh diễn ra nhanh hơn.

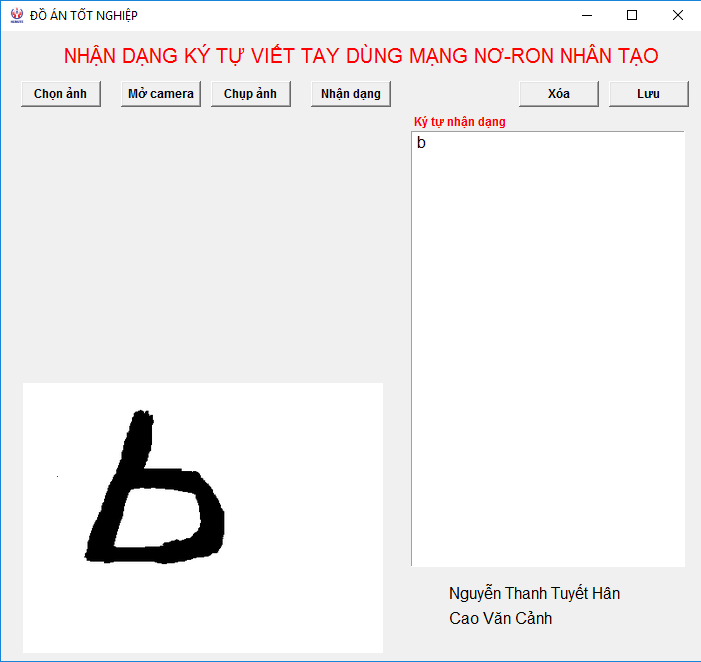
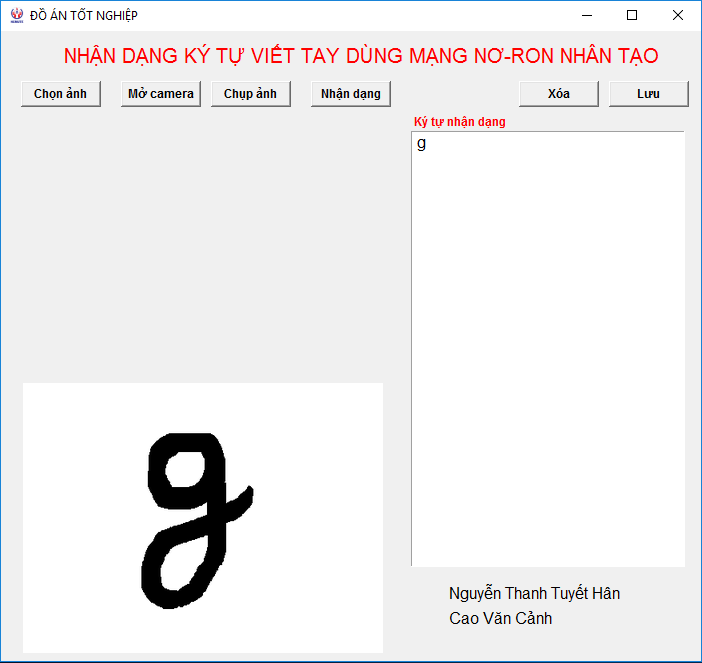
**Hình 4.10:** Một số hình nhận diện đúng

Nhận xét:Hệ thống sẽ cho kết quả nhận diện nhanh hơn đối với các trường hợp kí tự có nét viết đầy đủ, độ dày vừa phải và các kí tự không dính liền nhau, độ sáng ảnh văn bản không quá tối. Ngược lại, hệ thống sẽ nhận diện sai hoặc một vài trường hợp hệ thống không nhận diện được.

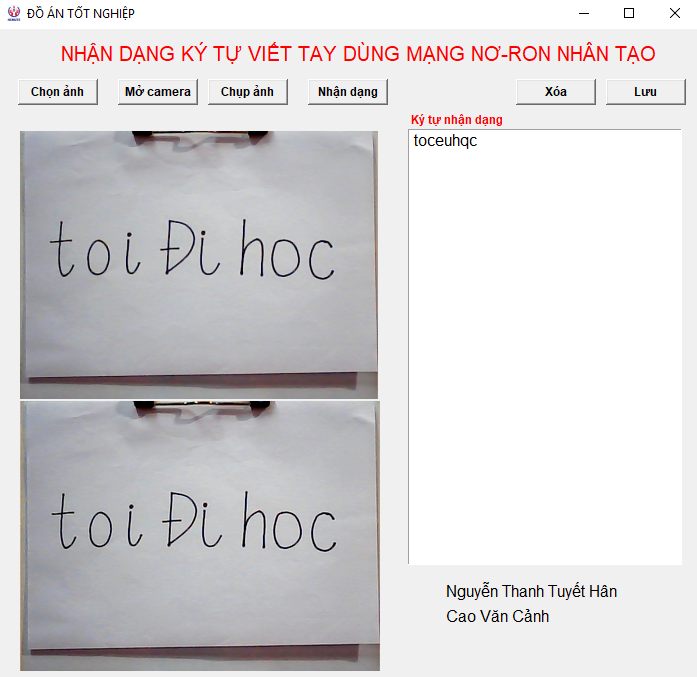
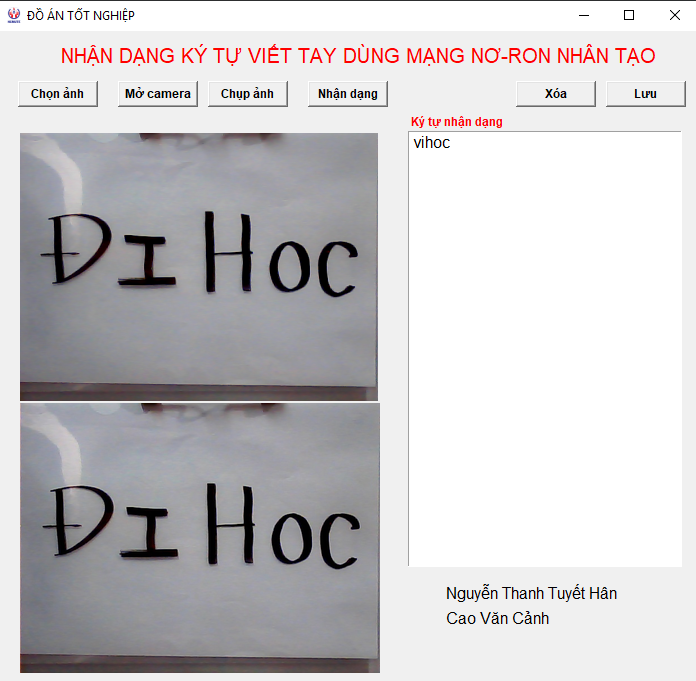
**Hình 4.11:** Một số hình nhận diện đúng

Nhận xét:Đối với trường hợp này, ảnh văn bản có chứa khung/viền không mong muốn thì việc này cũng không ảnh hưởng đến quá trình nhận diện, bởi vì quá trình tiền xử lý ảnh đã giải quyết vấn đề này. Các ký tự đáp ứng đầy đủ về khoảng cách, về độ dày cũng như nét viết cho thấy quá trình nhận diện sẽ diễn ra nhanh chóng.



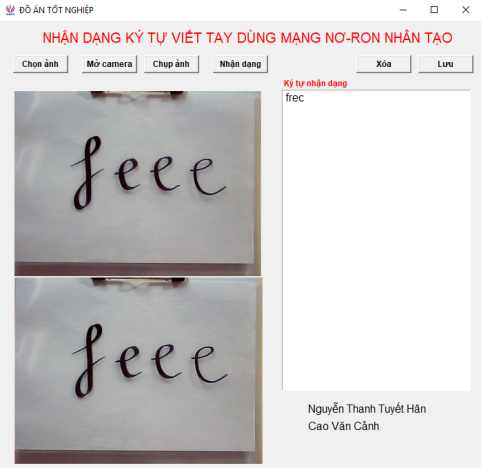
**Hình 4.12:** Một số hình nhận diện đúng

Nhận xét:Trong trường hợp này, ảnh văn bản đã đáp ứng đầy đủ về độ dày cũng như về định dạng nền trắng chữ đen. Tuy nhiên qua quá trình kiểm tra vấn đề nét viết được yêu cầu cao về trường hợp này nhằm đảm bảo độ chính xác đạt hiệu quả tốt nhất.

**Hình 4.13:** Một số hình nhận diện sai

Nhận xét:Dựa vào kết quả cho thấy, hệ thống có khả năng nhận dạng kém đối với ký tự có độ dày quá mỏng, trường hợp nét chữ viết cũng ảnh hưởng lớn đến nhận diện vì xác suất ký tự giống nhau xảy ra rất cao.



**Hình 4.14:** Một số hình nhận diện sai

Nhận xét:Trong trường hợp này ký tự f được viết với độ dày đầy đủ, nét viết cũng đảm bảo nên khả năng đúng rất cao, với ký tự e được chia thành ba trường hợp như minh họa trên cho thấy hệ thống phụ thuộc rất nhiều vào nét viết, độ cao các kí tự và độ dày cần thiết.

Như vậy qua quá trình kiểm tra và thực nghiệm cho thấy hệ thống có khả năng nhận dạng kém đối với những ký tự có độ dày dưới 0.2cm. Thời gian đáp ứng tùy thuộc vào số lượng ký tự trong hình, thời gian nhận dạng một ký tự sẽ từ 0.01 giây đến 0.015 giây. Ngoài ra, hệ thống khó nhận dạng được những ký tự dính liền nhau hoặc những ký tự quá nhỏ (có chiều cao nhỏ hơn 4cm và độ rộng nhỏ hơn 3cm).

# kết luận và hướng phát triển

## kết luận

Đề tài đã xây dựng một mạng nơ-ron nhân tạo có khả năng nhận diện được chữ viết tay với độ chính xác tốt.

Đề tài đã thiết kế một giao diện chương trình hoàn chỉnh, thuận tiện cho việc sử dụng. Chương trình đã xây dựng cho kết quả nhận dạng với tỷ lệ cao đối với các ảnh đầu vào chụp đủ cường độ sáng, vị trí, góc chụp thích hợp. Giao diện giao tiếp với người dùng được tối ưu hóa, rất đơn giản.

Bên cạnh đó, trong quá trình nhận diện ảnh văn bản cũng còn một số hạn chế như là: ký tự trong ảnh văn bản có độ dày nằm trong khoảng 0.2cm-0.5cm, khi hai ký tự viết liền nhau chương trình sẽ không nhận diện được. Ngoài ra, hệ thống không thể nhận diện được khoảng trắng giữa các từ cũng như số đếm. Bên cạnh đó, ký tự được nhận diện trên một ảnh văn bản còn giới hạn về số lượng.

## hướng phát triển

Do thời gian có hạn, đề tài được thiết kế còn mang tính nghiên cứu, nên để triển khai trong đề tài trong thực tế cần phải phát triển thêm các nội dung sau:

* Cải thiện tập mẫu để hệ thống đạt được độ chính xác cao hơn, nhận dạng đúng trong nhiều trường hợp và ở nhiều điều kiện khác nhau.
* Kết hợp thêm các phần cứng hỗ trợ, hệ thống chiếu sáng để tăng cường khả năng nhận dạng trong điều kiện thiếu sáng ,…
* Phát triển hệ thống không phụ thuộc nhiều vào chất lượng camera.
* Nhận diện sự khác biệt giữa chữ in hoa và chữ viết thường.
* Nhận diện được khoảng trắng giữa các ký tự và nhận dạng được số đếm.
* Có khả năng nhận diện hình ảnh có nhiều dòng ký tự.

TÀI LIỆU THAM KHẢO

1. Ngô Thị Nhung, “*Xử Lý Ảnh Trong Matlab*”, Báo cáo thực tập, Trường ĐHCN Hà Nội.
2. PGS.TS Nguyễn Quang Hoan, “*Xử Lý Ảnh*”, NXB Hà Nội, Học Viện Công Nghệ Bưu Chính Viễn Thông 2006.
3. PGS.TS Nguyễn Thanh Hải, “*Giáo trình Xử lí ảnh*”, NXB Đại học Quốc gia TP.HCM, Trường Đại học Sư Phạm Kỹ Thuật TPHCM 09/2015.
4. Bùi Ngọc Huy, “*Nghiên Cứu Kỹ Thuật Xử Lý Nhiễu, Hiệu Chỉnh Ảnh Nhị Phân và Ứng Dụng Cho Phiếu Thi Trắc Nghiệm*”, Luận Văn Thạc Sĩ, Trường Đại Học Công Nghệ Thông Tin Và Truyền Thông Thái Nguyên, 2013.
5. Michael Copeland, “*What’s the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning*”, July 29 2016.
6. Vũ Hữu Tiệp, “*Machine Learning cơ bản*”, Bài 35: Lược sử Deep Learning, 06/2018.
7. Vũ Hữu Tiệp, “*Machine Learning cơ bản*”, Bài 35: Perceptron Learning Algorithm, 01/2017.
8. Đỗ Minh Hải, “*Mạng nơ-ron nhân tạo - Neural Networks*”, < <https://dominhhai.github.io/vi/2018/04/nn-intro/>>
9. Kurt Hornik (1991), “*Approximation Capabilities of Multilayer Feedforward Networks*”, Neural Networks, 4(2), 251-257.
10. Neural Smithing (1999), “Supervised Learning in Feedforward Artificial Neural Networks”, Page 31.
11. Alex Krizhevsky, Ilya S., Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, Advances in neural information processing systems, 2012, pp. 1097 - 1105Christian Gerber, Mokdong Chung, “*Number Plate Detection with a MultiConvolutional Neural Network Approach with Optical Character Recognition for Mobile Devices*”, J Inf Process Syst, Vol.12, No.1, pp.100~108, March 2016.
12. Sagar Sharma, “*Activation Functions in Neural Networks*”, About Towards Data Science, September 6 2017.
13. Vũ Hữu Tiệp, “*Machine Learning cơ bản*”, Bài 2: Phân nhóm các thuật toán Machine Learning, 12/2016.
14. Nitish S., Geoffrey H., Alex Krizhevsky, Ilya Sutskever, Ruslan S., “*Dropout:A Simple Way to Prevent Neural Networks from Overfitting*”, Journal of Machine Learning Research 15, 2014, pp. 1929 – 1958.
15. Hinton, G.E. (2006),”A Fast Learning Algorithm for Deep Belief Nets” (PDF). Neural Computation.18(7):1527-1554.
16. Fei-Fei Li, Justin Johnson, Serena Yeung, “*Lecture 9: CNN Architecture*”, Stanford University, May 2, 2017.
17. Arden Dertat Applied Deep Learning - Part 4: Convolutional Neural Networks”,< <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>>.
18. Shashank Ramesh, " *A guide to an efficient way to build neural network architectures- Part II: Hyper-parameter selection and tuning for Convolutional Neural Networks using Hyperas on Fashion-MNIST*”,<<https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-ii-hyper-parameter-42efca01e5d7>>.