### BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM**

BÁO MÔN TRÍ

CÁO ĐỒ ÁN TUỆ NHÂN TẠO



*Chủ Đề:*

# XÂY DỰNG HỆ THỐNG NHẬN DẠNG CHỮ CÁI VIẾT TAY THÔNG QUA DẠNG ẢNH

Ngành: **CÔNG NGHỆ THÔNG TIN**

Giảng viên hướng dẫn: TS. Nguyễn Khắc Chiến

Sinh viên thực hiện: Lê Văn Khang – 2011063668 – 20DTHB1

TP. Hồ Chí Minh, 2023

# LỜI CAM ĐOAN

Đề tài "NHẬN DIỆN CHỮ CÁI VIẾT TAY" được thực hiện minh bạch bởi các thành viên trong nhóm và có sự hỗ trợ của Giảng viên hướng dẫn là Ths.Nguyễn Khắc Chiến. Nội dung, bảng biểu, hình ảnh và số liệu đều do chúng em tìm hiểu, nghiên cứu thực hiện và từ một số nguồn tham khảo mà nhóm chúng em có trích nguồn rõ ràng. Chúng em xin đảm bảo rằng toàn bộ nội dung trong báo cáo là trung thực, không gian lận từ một nguồn nào khác. Chúng em xin chịu mọi trách nhiệm để đảm bảo tính minh bạch trong của bài báo cáo.

TP.HCM, ngày tháng năm 2023.

Sinh viên đồng thực hiện **Lê Văn Khang**

**MỤC LỤC**

**[LỜI CAM ĐOAN 1](#_Toc4026)**

**[MỤC LỤC 2](#_Toc14877)**

**[DANH MỤC HÌNH ẢNH 3](#_Toc17665)**

**[CHƯƠNG 1: TỔNG QUAN 4](#_Toc28577)**

[1.1. Giới thiệu đề tài 4](#_Toc4488)

[1.2. Nhiệm vụ đồ án 4](#_Toc31594)

[1.3. Công nghệ sử dụng - Ngôn ngữ lập trình: Python 6](#_Toc28721)

[1.4. Cấu trúc đồ án 6](#_Toc22278)

**[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT](#_Toc32184)** [7](#_Toc32184)

[2.1. Giới thiệu về Python 7](#_Toc14338)

[2.2. Giới thiệu về Pycharm 7](#_Toc32301)

[2.3. Giới thiệu về Deep Learning 8](#_Toc3073)

[2.4. Giới thiệu về CNN 10](#_Toc17207)

[2.5. Phân tích thuật toán (Xử lý ảnh) 11](#_Toc13937)

[2.6. Đánh giá thuật toán 14](#_Toc8549)

[2.7. Các gói package cần cài: 15](#_Toc9489)

**[CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM](#_Toc21791)** [16](#_Toc21791)

[3.1. Tổng quan về hệ thống 16](#_Toc14772)

[3.2. Kết quả huấn luyện và đánh giá mô hình 17](#_Toc31723)

[3.3. Code demo 21](#_Toc15056)

[3.4. Chương trình demo 24](#_Toc11565)

**[CHƯƠNG 4: KẾT LUẬN](#_Toc18917)** [25](#_Toc18917)

[1. Kết quả đạt được 25](#_Toc21893)

[2. Mặt hạn chế 25](#_Toc5566)

[3. Hướng phát triển 25](#_Toc21109)

**[Bảng phân công 26](#_Toc28698)**

**[TÀI LIỆU THAM KHẢO 27](#_Toc30720)**

# 

# DANH MỤC HÌNH ẢNH

*[Hình](#_Toc2176)* [1](#_Toc2176) *[. Mô Hình CNN................................................................................................](#_Toc2176)*[9](#_Toc2176)

*[Hình](#_Toc6979)* [2](#_Toc6979) *[. Mô hình RNN](#_Toc6979)* [10](#_Toc6979)

*[Hình](#_Toc16707)* [3](#_Toc16707) *[. Mô Hình GAN](#_Toc16707)* [10](#_Toc16707)

*[Hình](#_Toc1070)* [4](#_Toc1070) *[. Quy trình hoạt động của Tích chập (Convolution)](#_Toc1070)* [11](#_Toc1070)

*[Hình](#_Toc27931)* [5](#_Toc27931) *[. Quy trình hoạt động của Padding](#_Toc27931)* [12](#_Toc27931)

*[Hình](#_Toc12294)* [6](#_Toc12294) *[. Quy trình hoạt động của Pooling](#_Toc12294)* [12](#_Toc12294)

*[Hình](#_Toc13324)* [7](#_Toc13324) *[. Minh họa quy trình của Max Pooling và Average Pooling](#_Toc13324)* [12](#_Toc13324)

*[Hình](#_Toc26262)* [8](#_Toc26262) *[. Quy trình hoạt động của Fully Connected](#_Toc26262)* [13](#_Toc26262)

*[Hình](#_Toc30937)* [9](#_Toc30937) *[. Quy trình hoạt động của One – hot encoding](#_Toc30937)* [13](#_Toc30937)

*[Hình](#_Toc25169)* [10](#_Toc25169) *[. Sơ đồ tổng quan của hệ thống](#_Toc25169)* [16](#_Toc25169)

*[Hình](#_Toc6181)* [11](#_Toc6181) *[. Một vài ảnh chữ cái trong thư viện Alphabets](#_Toc6181)* [17](#_Toc6181)

*[Hình](#_Toc10387)* [12](#_Toc10387) *[. Sau khi huấn luyện](#_Toc10387)* [18](#_Toc10387)

*[Hình](#_Toc8920)* [13](#_Toc8920) *[. Đồ thị loss, accuracy của training set và validation set theo epoch](#_Toc8920)* [18](#_Toc8920)

*[Hình](#_Toc14122)* [14](#_Toc14122) *[. Biểu đồ hiển thị số lượng các chữ cái trong dữ liệu test](#_Toc14122)* [19](#_Toc14122)

*[Hình](#_Toc6084)* [15](#_Toc6084) *[. Tỷ lệ chính xác các chữ cái](#_Toc6084)* [20](#_Toc6084)

*[Hình](#_Toc11029)* [16](#_Toc11029) *[. Khai báo thư viện và đọc dữ liệu](#_Toc11029)* [21](#_Toc11029)

*[Hình](#_Toc24713)* [17](#_Toc24713) *[. Mô hình CNN](#_Toc24713)* [21](#_Toc24713)

*[Hình](#_Toc17547)* [18](#_Toc17547) *[. Huấn luyện và đánh giá chữ cái](#_Toc17547)* [22](#_Toc17547)

*[Hình](#_Toc5557)* [19](#_Toc5557) *[. Lưu tệp và đánh giá](#_Toc5557)* [22](#_Toc5557)

*[Hình](#_Toc27712)* [20](#_Toc27712) *[. Vẽ biểu đồ](#_Toc27712)* [22](#_Toc27712)

*[Hình](#_Toc20016)* [21](#_Toc20016) *[. Load ảnh, xử lý và tìm nét](#_Toc20016)* [23](#_Toc20016)

*[Hình](#_Toc7651)* [22](#_Toc7651) *[. Chức năng dự đoán ký tự](#_Toc7651)* [23](#_Toc7651)

*[Hình](#_Toc32216)* [23](#_Toc32216) *[. Duyệt contour, resize ảnh, vẽ hình và đặt ký tự](#_Toc32216)* [23](#_Toc32216)

*[Hình](#_Toc23531)* [24](#_Toc23531) *[. Hiện thị ảnh với ký tự dự đoán](#_Toc23531)* [23](#_Toc23531)

*[Hình](#_Toc10088)* [25](#_Toc10088) *[. Kết quả nhận dạng](#_Toc10088)* [24](#_Toc10088)

# CHƯƠNG 1: TỔNG QUAN

## Giới thiệu đề tài

Hiện nay, với sự phát triển mạnh mẽ của công nghệ, khối lượng dữ liệu hình ảnh trên toàn thế giới ngày càng khổng lồ. Việc phân tích, xử lý để khai thác sử dụng thông tin trong ảnh là một nhu cầu thiết yếu. Tuy nhiên, với số lượng dữ liệu khổng lồ, việc phân tích, xử lý thủ công sẽ mất rất nhiều thời gian và nguồn nhân lực.Thực tiễn đã chứng minh, việc ứng dụng trí tuệ nhân tạo (AI), cụ thể là Deep Learning, vào các nhiệm vụ phân tích, xử lý hình ảnh đã giúp tiết kiệm được nhiều thời gian và công sức.

Nhận dạng chữ cái viết tay (Handwriting Recognition - HWR) là một lĩnh vực nghiên cứu quan trọng trong thị giác máy tính. Bài toán nhận dạng chữ cái viết tay là bài toán nhận dạng các chữ cái từ các hình ảnh chữ cái viết tay. Bài toán này có độ khó cao do chữ cái viết tay có thể có nhiều biến thể khác nhau, phụ thuộc vào phong cách viết của người viết.

Đối với bài toán mạng nơron nhiều lớp sẽ được huấn luyện dựa trên các pixel, đơn vị nhỏ nhất của hình ảnh. Vì vậy mạng nơron nhân tạo là công cụ vô cùng thích hợp cho việc xử lý, phân tích hình ảnh và mang lại kết quả rất khả quan .

Dựa trên những lý do trên, nhóm nghiên cứu của chúng em lựa chọn đề tài “Nhận diện chữ cái viết tay”. Trong phạm vi của đề tài, nhóm sẽ tập trung nghiên cứu, tìm hiểu về Deep Learning, cụ thể là thuật toán CNN nhằm áp dụng vào xây dựng mô hình nhận diện chữ số viết tay bằng mạng nơron nhân tạo cho tỷ lệ nhận dạng đúng trên 90%..

## Nhiệm vụ đồ án

* + 1. **Tính cấp thiết đề tài**

Trong những năm gần đây, với sự phát triển của công nghệ, 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 số. Nhận dạng chữ số viết tay là bài toán được quan tâm rất nhiều vì nó là một trong các yêu cầu trong nhiều ứng dụng thực tế.

Để giải quyết bài toán nhận dạng chữ số viết tay, nhóm nghiên cứu đã lựa chọn sử dụng thuật toán CNN (Convolutional Neural Network) trong lĩnh vực Deep Learning.

Vì lý do đó, đề tài **“Nhận diện chữ số viết tay”** trong bộ môn “Trí tuệ nhân tạo” này của em sẽ tập trung tìm hiểu và ứng dụng thuật toán trong lĩnh vực Deep Learning phục vụ cho việc giải quyết vấn đề phát hiện những chữ cái trong văn bản một cách chính xác.

### Ý nghĩa khoa học và thực tiễn của đề tài

Nhận dạng chữ số viết tay là bài toán được quan tâm rất nhiều vì nó là một trong các yêu cầu trong thực tế được ứng dụng vào đời sống như :

* **Giáo dục:**

- **Chấm bài tự luận:** sử dụng để chấm bài tự luận của học sinh, giúp giáo viên tiết kiệm thời gian và công sức.

- **Đánh giá bài tập viết:** sử dụng để đánh giá bài tập viết của học sinh, giúp giáo viên nắm bắt được tình hình học tập của học sinh.

* **Y tế:**

- **Đọc kết quả xét nghiệm:** sử dụng để đọc kết quả xét nghiệm của bệnh nhân, giúp bác sĩ đưa ra chẩn đoán chính xác.

- **Đơn thuốc:** sử dụng để đọc đơn thuốc của bác sĩ, giúp dược sĩ chuẩn bị thuốc chính xác.

* **Tài chính:**

- **Phân tích tài liệu:** sử dụng để phân tích các tài liệu tài chính, chẳng hạn như hợp đồng, hóa đơn,...

- **Ký tên điện tử:** sử dụng để ký tên điện tử, giúp người dùng ký tên mà không cần sử dụng bút và giấy.

* **Công nghiệp:**

- **Nhập liệu:** sử dụng để nhập liệu từ các tài liệu viết tay.

- **Kiểm soát chất lượng:** sử dụng để kiểm tra chất lượng sản phẩm.

### Mục tiêu, đối tượng và phạm vi nghiên cứu của đề tài

Mục tiêu của đề tài là xây dựng một mô hình nhận dạng chữ số viết tay bằng thuật toán CNN có độ chính xác cao, có thể nhận dạng chính xác các chữ số viết tay trong nhiều điều kiện khác nhau.

Đối tượng nghiên cứu của đề tài là các chữ cái từ A đến Z.

Phạm vi nghiên cứu của đề tài bao gồm:

* Nghiên cứu các ký thuật nhận dạng chữ cái viết tay
* Xây dựng mô hình nhận dạng chữ cái viết tay bằng thuật toán CNN
* Đánh giá hiệu qua mô hình

## Công nghệ sử dụng - Ngôn ngữ lập trình: Python

### - Công cụ hỗ trợ:Pycharm 2023.2.1

### 

## Cấu trúc đồ án

Cấu trúc báo cáo đồ án nhóm em, nghiên cứu về “Nhận diện chữ cái viết tay”. Nội dung nghiên cứu gồm 4 chương như sau:

**Chương 1 – Tổng quan:** Giới thiệu khái quát và mục tiêu của đề tài, mục tiêu, đối tượng và pham vi đề tài. Giới thiệu về các kiến thức nền tảng cũng như công nghệ và phần mềm được sử dụng trong đề tài.

**Chương 2 – Cơ sở lý thuyết:** Trình bày các lý thuyết, khái niệm và định nghĩa có liên quan đến đề tài.

**Chương 3 – Kết quả thực nghiệm:** Trình bày tổng quan về hệ thống, chạy demo, đưa ra kết quả và đánh giá.

**Chương 4 – Kết luận:** Tổng kết quá trình thực nghiệm và rút ra hướng phát triển sau này của đồ án.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## Giới thiệu về Python

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

**Những lợi ích của Python bao gồm:**

* Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
* Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
* Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
* Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
* Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
* Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
* Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

## Giới thiệu về Pycharm

PyCharm được phát triển bởi Jet Brains và đây là môi trường phát triển tích hợp đa nền tảng (IDE) được thiết kế đặc biệt cho Python. Đây là IDE được sử dụng rộng rãi nhất và có sẵn ở cả phiên bản trả phí và nguồn mở miễn phí.

PyCharm là một IDE Python hoàn hảo với một các tính năng phong phú như tự đồng hoàn thiện code, điều hướng project nhanh, test và debug nhanh, hỗ trợ phát triển từ xa, khả năng truy cập cơ sở dữ liệu, v.v.

**Tính năng, đặc điểm:**

* Điều hướng mã thông minh
* Đánh dấu lỗi và trình gỡ lỗi (Debug) mạnh mẽ
* Hỗ trợ các framwork phát triển web python, Angular JS, Javascript

## Giới thiệu về Deep Learning

### Khái niệm về Deep Learning

Deep Learning (học sâu) là một nhánh của học máy dựa trên lý thuyết của mạng nơron nhân tạo. Deep Learning sử dụng các mô hình mạng nơron nhân tạo có nhiều lớp, mỗi lớp thực hiện một nhiệm vụ cụ thể trong quá trình học hỏi.

Mạng nơron nhân tạo là một mô hình toán học mô phỏng hoạt động của bộ não con người. Mạng nơron nhân tạo có nhiều lớp, mỗi lớp bao gồm các nơron. Các nơron trong cùng một lớp được kết nối với nhau thông qua các trọng số.

Trong Deep Learning, các mô hình mạng nơron nhân tạo được huấn luyện trên một tập dữ liệu lớn. Tập dữ liệu này bao gồm các mẫu đầu vào và các mẫu đầu ra mong muốn. Quá trình huấn luyện mô hình mạng nơron nhân tạo là quá trình tìm cách tối ưu hóa các trọng số của mô hình sao cho mô hình có thể dự đoán chính xác các mẫu đầu ra mong muốn từ các mẫu đầu vào.

Deep Learning đã đạt được những thành tựu đáng kể trong nhiều lĩnh vực, bao gồm:

* **Xử lý hình ảnh:** Deep Learning được sử dụng để nhận dạng khuôn mặt, nhận dạng chữ viết tay, phân loại hình ảnh,…
* **Xử lý ngôn ngữ tự nhiên:** Deep Learning được sử dụng để dịch ngôn ngữ, tổng hợp giọng nói, trả lời câu hỏi,...
* **Trí tuệ nhân tạo:** Deep Learning được sử dụng để phát triển các hệ thống trí tuệ nhân tạo như robot, xe tự lái,…

**Ưu điểm**

Deep Learning là một bước ngoặt to lớn trong lĩnh vực [trí tuệ nhân tạo](https://vietnix.vn/ai-la-gi/" \t "https://vietnix.vn/deep-learning-la-gi/_blank), cho phép khác nhà khoa học dữ liệu xây dựng nhiều mô hình có độ chính xác rất cao trong lĩnh vực nhận dạng ảnh, xử lý ngôn ngữ tự nhiên, xử lý giọng nói,… Một số ưu điểm vượt trội của Deep Learning gồm có:

* Kiến trúc mạng nơ-ron linh hoạt, có thể dễ dàng thay đổi để phù hợp với nhiều vấn đề khác nhau.
* Có khả năng giải quyết nhiều bài toán phức tạp với độ chính xác rất cao.
* Tính tự động hoá cao, có khả năng tự điều chỉnh và tự tối ưu.
* Có khả năng thực hiện tính toán song song, hiệu năng tốt, xử lý được lượng dữ liệu lớn.

**Nhược điểm**

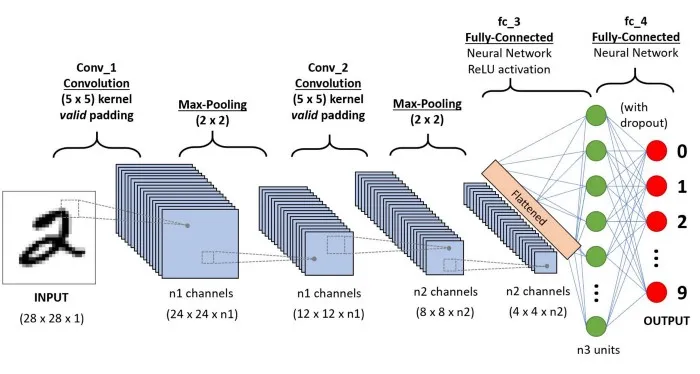
Bên cạnh những ưu điểm, mặt khác, hiện nay Deep Learning vẫn còn nhiều khó khăn và hạn chế, chẳng hạn như:

* Cần có khối lượng dữ liệu rất lớn để tận dụng tối đa khả năng của Deep Learning.
* Chi phí tính toán cao vì phải xử lý nhiều mô hình phức tạp.
* Chưa có nền tảng lý thuyết mạnh mẽ để lựa chọn các công cụ tối ưu cho Deep Learning.

### Các kỹ thuật và thuật toán Deep Learning

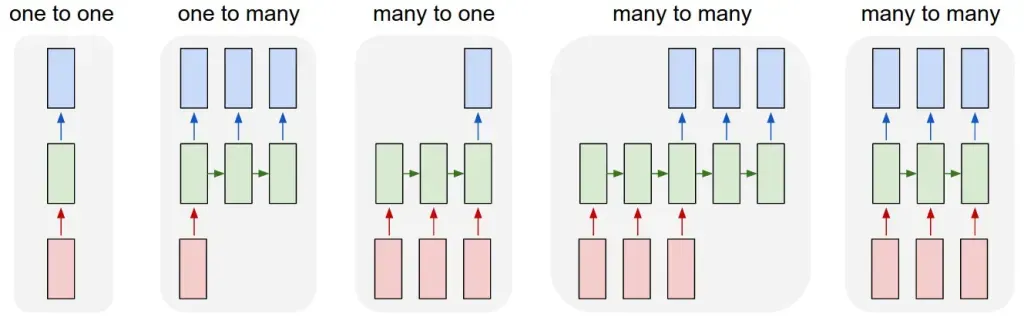
Có nhiều kỹ thuật và thuật toán Deep Learning khác nhau, từ đơn giản đến vô cùng phức tạp, có thể ứng dụng vào hầu hết các bài toán trong lĩnh vực trí tuệ nhân tạo hiện nay. Một số kỹ thuật, thuật toán deep learning phổ biến nhất hiện nay:

* **Mạng nơron cổ điển :** là mạng kết nối đầy đủ, thường được xác định bằng các perceptron đa lớp. chủ yếu được sử dụng cho các bài toán phân lớp nhị phân.
* **Mạng nơron tích chập (Convolutional Neural Network - CNN):** là một kiến trúc [Neural Network](https://vietnix.vn/neural-network/" \t "https://vietnix.vn/deep-learning-la-gi/_blank) nhân tạo nâng cao, được xây dựng để giải quyết các bài toán phức tạp, đặc biệt là liên quan đến xử lý hình ảnh.



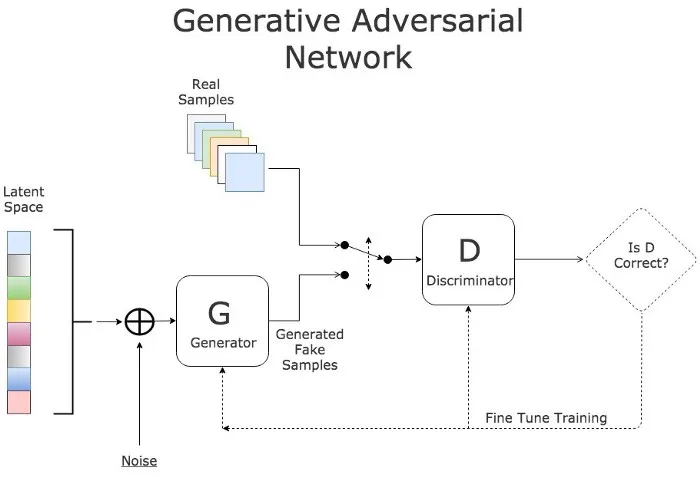
*Hình 1**. Mô Hình CNN*

* **Mạng nơron hồi quy (Recurrent Neural Network - RNN):** là một loại mạng nơron nhân tạo có khả năng học hỏi các mối quan hệ giữa các dữ liệu theo thời gian. RNN được sử dụng phổ biến trong các bài toán xử lý ngôn ngữ tự nhiên, chẳng hạn như dịch ngôn ngữ, tổng hợp giọng nói, trả lời câu hỏi,...



*Hình 2**. Mô hình RNN*

* **Mạng sinh đối nghịch (Generative Adversarial Networks - GAN):** là lớp mô hình có mục tiêu tạo ra dữ liệu giả giống với thật, tên của mạng được dựa trên kiến trúc gồm hai mạng có mục tiêu đối nghịch nhau: Generator và Discriminator. Trong đó Generator học cách sinh dữ liệu giả để lừa mô hình Discriminator, còn Discriminator lại học cách phân biệt giữa dữ liệu giả và dữ liệu thật



*Hình 3**. Mô Hình GAN*

## Giới thiệu về CNN

****CNN**** được viết tắt của ****Convolutional Neural Network**** hay còn được gọi là****CNNS**** mang nơ-ron tích chập, là một trong những [mô hình Deep Learning](https://vietnix.vn/deep-learning-la-gi/" \t "https://vietnix.vn/cnn-la-gi/_blank) cực kỳ tiên tiến, bởi chúng cho phép bạn xây dựng những hệ thống có độ chính xác cao và thông minh. Nhờ khả năng đó, CNN có rất nhiều ứng dụng, đặc biệt là những bài toán cần nhận dạng vật thể (object) trong ảnh.

Trong thuật toán CNN, mục đích của CNN là tập hợp các thuật toán để có mô hình dữ liệu bằng cách sử dụng nhiều lớp xử lý cấu trúc phức tạp. Hiểu đơn giản, CNN là

một lớp của mạng nơron sâu, được áp dụng phổ biến nhất để phân loại dữ liệu như hình ảnh, video, lời nói trực quan.

CNN vô cùng quan trọng để tạo nên những hệ thống nhận diện thông minh với độ chính xác cao trong thời đại công nghệ ngày nay. Lý do cụ thể vì sao CNN đặc biệt phát huy hiệu quả trong việc nhận dạng (detection), chúng ta sẽ tìm hiểu kỹ hơn ngay dưới đây.

## Phân tích thuật toán (Xử lý ảnh)

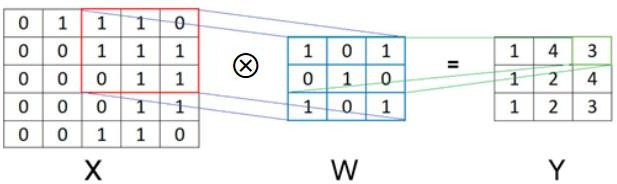
### Tích chập (Convolution):

Kernel là một ma trận vuông kích thước k\*k trong đó k là số lẻ: 1, 3, 5, 7, 9,… Ví dụ kernel kích thước 3\*3.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 0 | 1 |
| 𝑊 | = [0 | 1 | 0] |
|  | 1 | 0 | 1 |

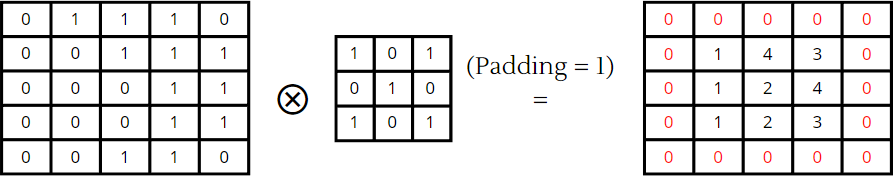
Kí hiệu phép tính tích chập (⊗), kí hiệu Y=X⊗W.

Với từng phần tử Xij trong ma trận X lấy ra một ma trận có kích thước bằng kích thước của kernel W có phần tử Xij gọi là ma trận A. Sau đó tính tổng các phần tử của ma trận A và ma trận W, rồi viết vào ma trận kết quả Y.



*Hình 4**. Quy trình hoạt động của Tích chập (Convolution)*

### Padding:

Mỗi lần thực hiện phép tính tích chập xong thì kích thước của ma trận Y đều nhỏ hơn X. Tuy nhiên nếu muốn ma trận Y thu được có kích thước bằng ma trận X thì bổ sung giá trị 0 ở viền ngoài ma trận X. Padding = k có nghĩa là thêm k vector 0 vào bốn phía của ma trận.

*Hình 5**. Quy trình hoạt động của Padding*

### Stride:

Nếu thực hiện tuần tự các phần tử trong ma trận X , thu được ma trận Y cùng kích thước ma trận X , ta gọi là stride = 1. Tuy nhiên nếu stride = k ( k > 1 ) thì ta chỉ thực hiện phép tính convolution trên các phần tử 𝑋1+𝑖∗𝑘,1+𝑗∗𝑘.

Công thức tổng quát cho phép tính tích chập của ma trận X kích thước m\*n với kernel kích thước k \* k , stride = s , padding = p ra ma trận Y kích thước:

(

### Pooling:

m − k + 2p

𝑠

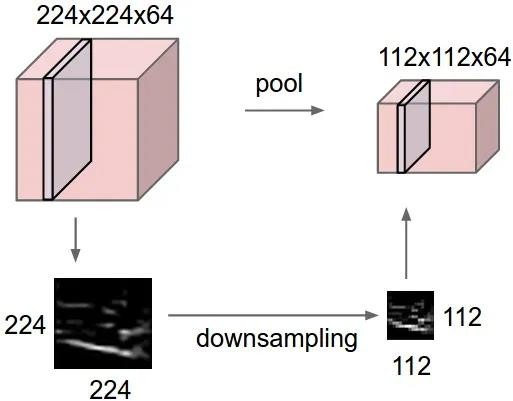
+ 1) × (

n − k + 2p

𝑠

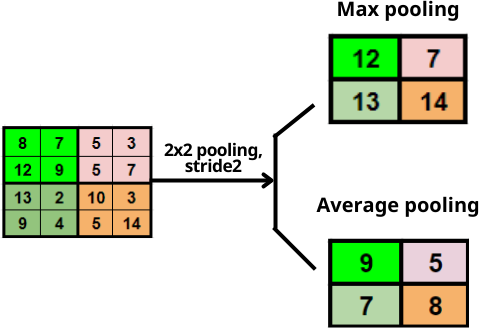
+ 1)

Pooling layer là lớp ở giữa các lớp convolution layer có tác dụng giảm kích thước dữ liệu . Thông thường , khi pooling thì dùng kernel kích thước (2x2 ).



*Hình 6**. Quy trình hoạt động của Pooling*

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

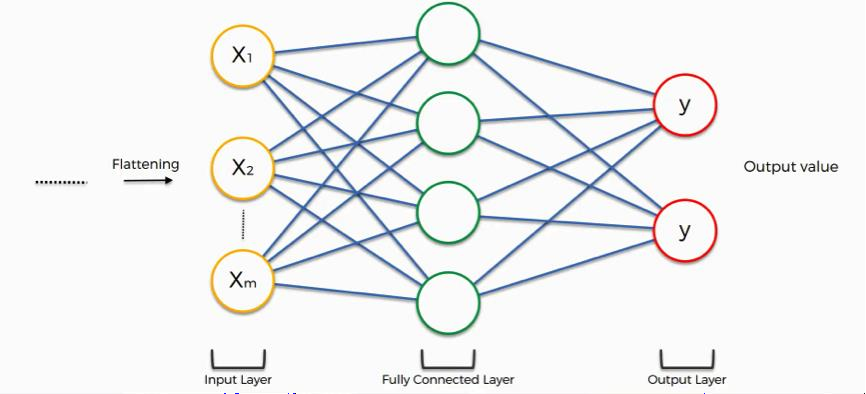


*Hình 7**. Minh họa quy trình của Max Pooling và Average Pooling*

### Fully connected:

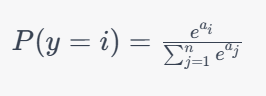
Sau khi ảnh được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của ảnh ( ví dụ mắt , mũi , khung mặt, ...) thì tensor của output của layer cuối cùng, kích thước H \* W \* D , sẽ được chuyển về 1 vector kích thước ( H \* W \* D ) .

Sau đó ta dùng các fully connected layer để kết hợp các đặc điểm của ảnh để ra được output của model .



*Hình 8**. Quy trình hoạt động của Fully Connected*

### Hàm softmax:



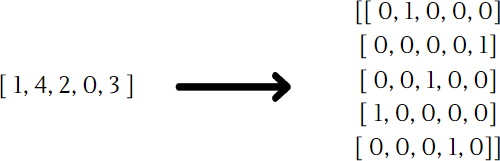
Trong đó :

- P(y=i) là xác suất để đầu ra là i.

- ai là giá trị đầu vào ứng với lớp i

- ∑j=1neaj là tổng của tất cả các giá trị exponential của đầu vào.

### One - hot encoding:

Chuyển đổi label của ảnh từ giá trị sang vector cùng kích thước với output của model.

*Hình 9**. Quy trình hoạt động của One – hot encoding*

## Đánh giá thuật toán

CNN (Convolutional Neural Network) là một thuật toán khá dễ hiểu và dễ cài đặt

được tạo ra để đem đến cho các lập trình viên một mô hình Deep Learning cực kỳ tiến bộ. Tuy có thể bản chất và cách ứng dụng nó không hề dễ dàng, nhưng kết quả nó mang lại là rất đáng phải lưu tâm. Thuật toán CNN cũng có một số ưu nhược điểm riêng:

### Ưu điểm

Một số ưu điểm vượt trội của CNN gồm có:

* Ưu điểm chính của CNN so với các phiên bản tiền nhiệm là nó tự động phát hiện các tính năng quan trọng mà không cần bất kỳ sự giám sát nào của con người, độ chính xác cao trong các vấn đề nhận dạng hình ảnh. Ví dụ, với nhiều hình ảnh về chó và mèo, nó sẽ tự học các đặc điểm riêng biệt cho từng lớp.
* CNN cũng hiệu quả về mặt tính toán. Nó sử dụng các phép toán tích chập và gộp và thực hiện chia sẻ tham số. Điều này cho phép các mô hình CNN chạy trên mọi thiết bị.
* CNN có thể học các đặc trưng cục bộ từ dữ liệu đầu vào, chẳng hạn như các cạnh, đường cong và các đối tượng nhỏ. Điều này giúp CNN có thể xử lý dữ liệu có cấu trúc một cách hiệu quả.
* CNN sử dụng các bộ lọc để học các đặc trưng cục bộ. Điều này giúp giảm số lượng tham số cần thiết để huấn luyện mô hình, từ đó giúp mô hình có thể học nhanh hơn và hiệu quả hơn.
* CNN có thể chống biến dạng trong dữ liệu đầu vào. Điều này giúp mô hình có thể nhận dạng các đối tượng ngay cả khi đối tượng đó bị dịch chuyển, xoay hoặc thay đổi kích thước.

### Nhược điểm

Bên cạnh những ưu điểm, mặt khác, hiện nay CNN vẫn còn nhiều khó khăn và hạn chế, chẳng hạn như:

* Khả năng giải thích: sử dụng các hàm phí tuyến để học các đặc trưng từ dữ liễu đầu vào. Điều này khiến viếc giải thích các mô hình đưa ra quyết định trở nên khó khắn.
* Khả năng mở rộng: CNN có thể trở nên phức tạp và khó huấn luyện khi số lượng tham số tăng.
* CNN không mã hóa vị trí và hướng của đối tượng.
* Thiếu khả năng bất biến trong không gian đối với dữ liệu đầu vào.
* Rất nhiều dữ liệu đào tạo được yêu cầu.
* Khó hơn với những đoạn văn dài khi lượng từ vựng lớn.

## Các gói package cần cài:

Keras là một giao diện lập trình ứng dụng (API) mạng thần kinh cho Python được tích hợp chặt chẽ với TensorFlow do Google phát triển , được sử dụng để xây dựng các mô hình học máy. Công cụ này có thể sử dụng chung với các thư viện Deep Learning nổi tiếng như TensorFlow, CNTK, Theano. Các mô hình của Keras cung cấp một cách đơn giản, thân thiện với người dùng để xác định mạng nơ-ron và nó còn có thể chạy trên cả CPU và GPU.

TensorFlow là một thư viện mã nguồn mở hàng đầu được thiết kế để phát triển và triển khai các ứng dụng học máy (machine learning) tiên tiến nhất. TensorFlow là một hệ thống phong phú để quản lý mọi khía cạnh của một hệ thống máy học tuy nhiên, lớp học này tập trung vào việc sử dụng một API TensorFlow cụ thể để phát triển và huấn luyện các mô hình máy học

Matplotlib là một thư viện vẽ đồ thị cho ngôn ngữ lập trình Python. Nó cung cấp một API hướng đối tượng để tạo các biểu đồ, đồ thị, và các hình ảnh khác. Thường dùng vẽ đồ thị trong một hình, vẽ một số đường trong vùng vẽ đồ thị, trang trí đồ thị bằng nhãn, v.v.

OpenCV trong học sâu là một khía cạnh cực kỳ quan trọng của nhiều thuật toán Học máy. OpenCV là một thư viện mã nguồn mở (gói) dành cho các ứng dụng thị giác máy tính, học máy và xử lý hình ảnh chỉ chạy trên CPU

Scikit-learning (Sklearn) là một thư viện hữu ích và mạnh mẽ nhất để học máy bằng Python. Sklearn cung cấp một loạt các công cụ hiệu quả để học máy và mô hình thống kê bao gồm phân loại, hồi quy, phân cụm và giảm kích thước thông qua giao diện nhất quán trong Python. Sklearn phần lớn được viết bằng ngôn ngữ Python, được xây dựng dựa trên NumPy, SciPy và Matplotlib.

# CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

## Tổng quan về hệ thống

**Input:** 1 tấm ảnh có các chữ cái.

### Các bước :

### Đọc và tiền xử lý hình ảnh đầu vào

* Đầu tiên, bạn đọc hình ảnh đầu vào bằng OpenCV. Sau đó, bạn tạo một bản sao của hình ảnh để thực hiện các thay đổi trên nó.
* Áp dụng một bộ lọc Gaussian để làm mờ hình ảnh và làm cho việc xác định đường viền dễ dàng hơn. Sau đó chuyển đổi hình ảnh sang ảnh grayscale để giảm chiều sâu màu.
* Áp dụng ngưỡng nhị phân để tạo hình ảnh nhị phân (đen trắng) với các đối tượng màu trắng và nền đen.

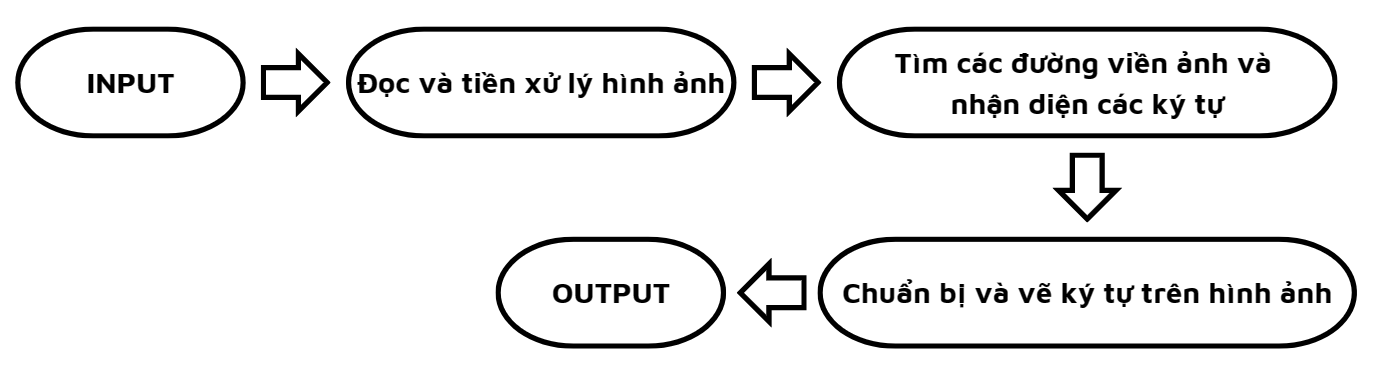
### Tìm các đường viền của các ký tự

* Sử dụng hình ảnh nhị phân, bạn tìm các đường viền của các ký tự bằng cách sử dụng **cv2.findContours(**).
* hàm này nhận một hình ảnh của một ký tự, sử dụng mô hình để dự đoán ký tự đó, và trả về ký tự dự đoán.

### Vẽ ký tự trên hình ảnh đầu ra

* Đối với mỗi đối tượng được tìm thấy, bạn cắt ra ký tự từ hình ảnh ban đầu dựa trên vị trí và kích thước của đối tượng.
* Ký tự này sau đó được thay đổi kích thước để phù hợp với kích thước đầu vào mong muốn của mô hình (28x28 pixel).
* Tiếp theo, bạn sử dụng mô hình đã được huấn luyện để dự đoán ký tự trong hình ảnh nhỏ này.
* Kết quả dự đoán được vẽ lên hình ảnh đầu ra dưới dạng văn bản, và một hình chữ nhật màu xanh lá cây được vẽ xung quanh ký tự để chỉ ra ký tự đã nhận diện.

**Output:** Hiển thị các số nhận diện được trên từng chữ số.

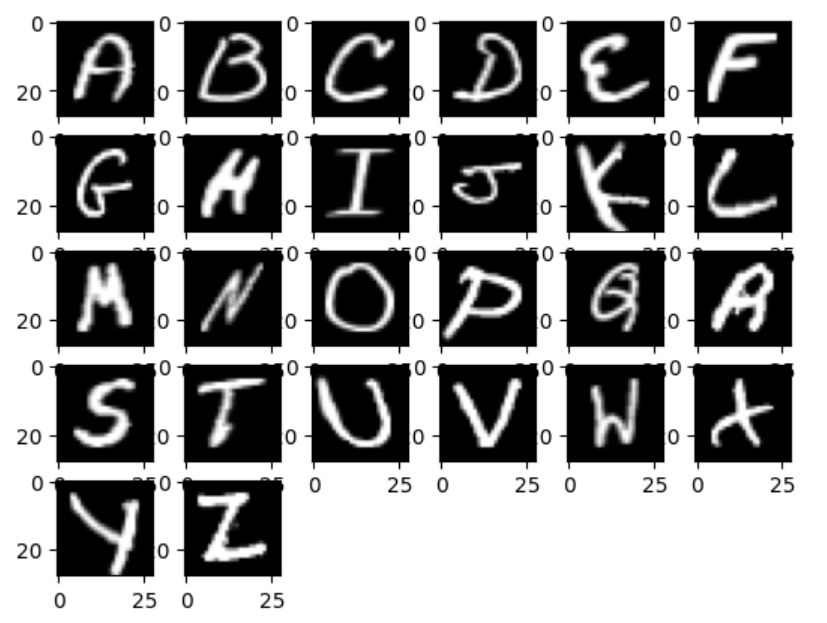


*Hình 10**. Sơ đồ tổng quan của hệ thống*

## Kết quả huấn luyện và đánh giá mô hình

### Bộ dữ liệu

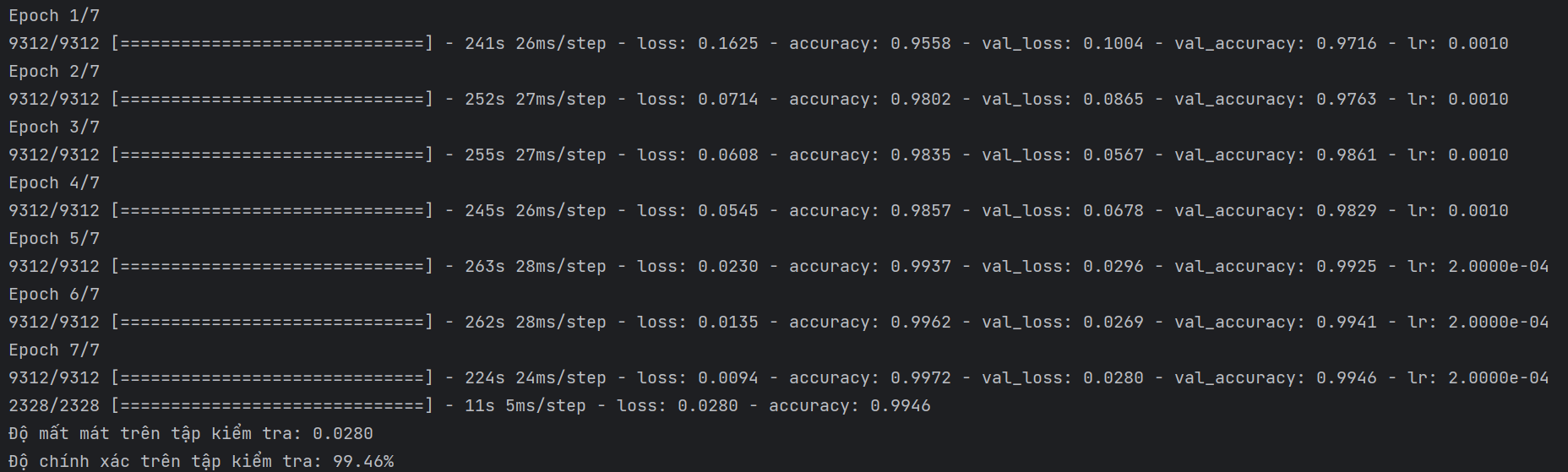
Bộ dữ liệu được sử dụng trong demo là bộ dữ liệu thư viện “ALPHABETS” có sẵn chứa bộ dữ liệu chứa 26 thư mục (A-Z) chứa các hình ảnh viết tay có kích thước 2828 pixel, mỗi bảng chữ cái trong hình ảnh được gắn ở giữa vào ô 2020 pixel.



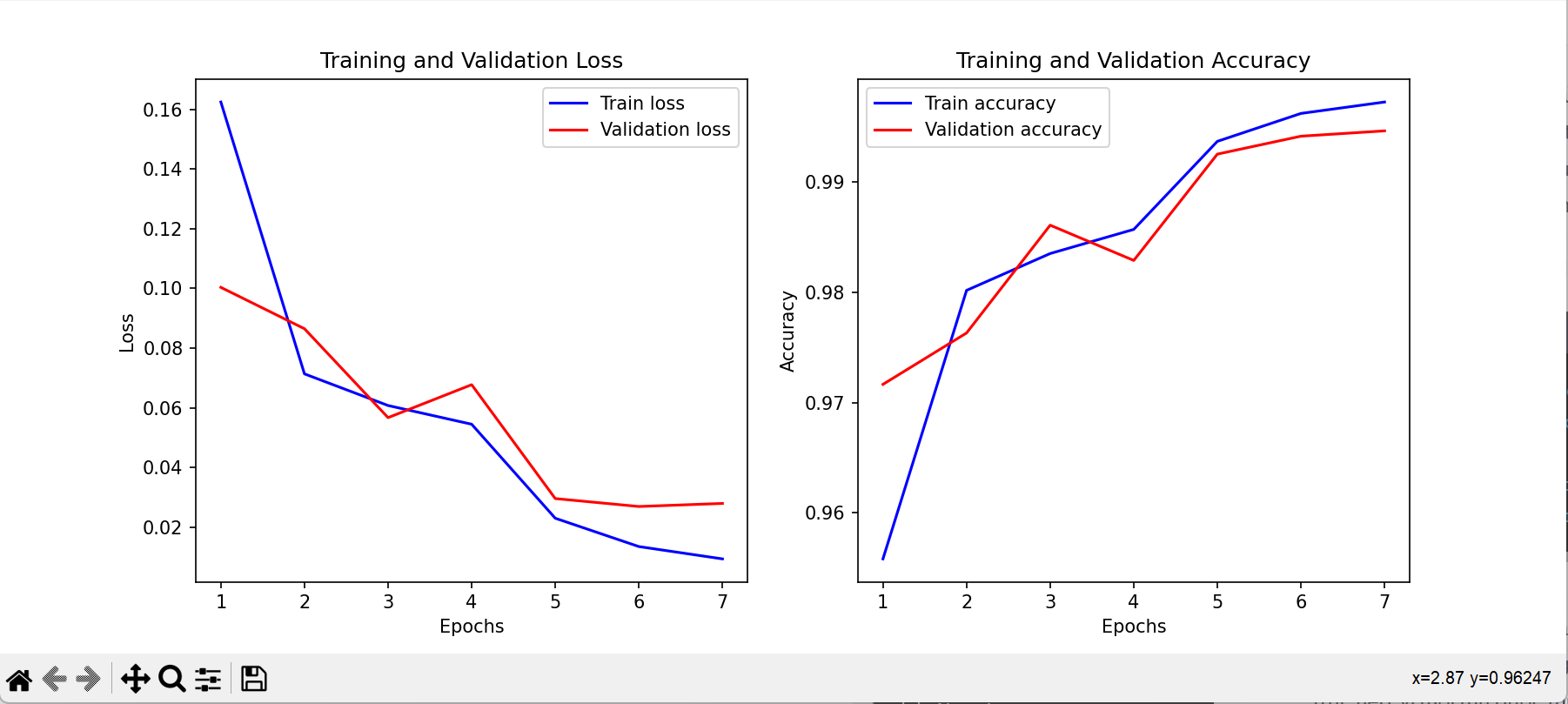
*Hình 11**. Một vài ảnh chữ cái trong thư viện Alphabets*

### Kết quả huấn luyện

Kết quả quá trình huấn luyện của mô hình sau 7 epochs cho ta thấy cho ta thấy tỷ lệ chính xác của huấn luyện của tập train là khá cao 99,72% và tỷ lệ thất bại của tập train là 0,9%. Tập validation kiểm định mô hình có tỷ lệ chính xác thực là 99,46% và tỷ lệ thất bại thực là 2,8%.



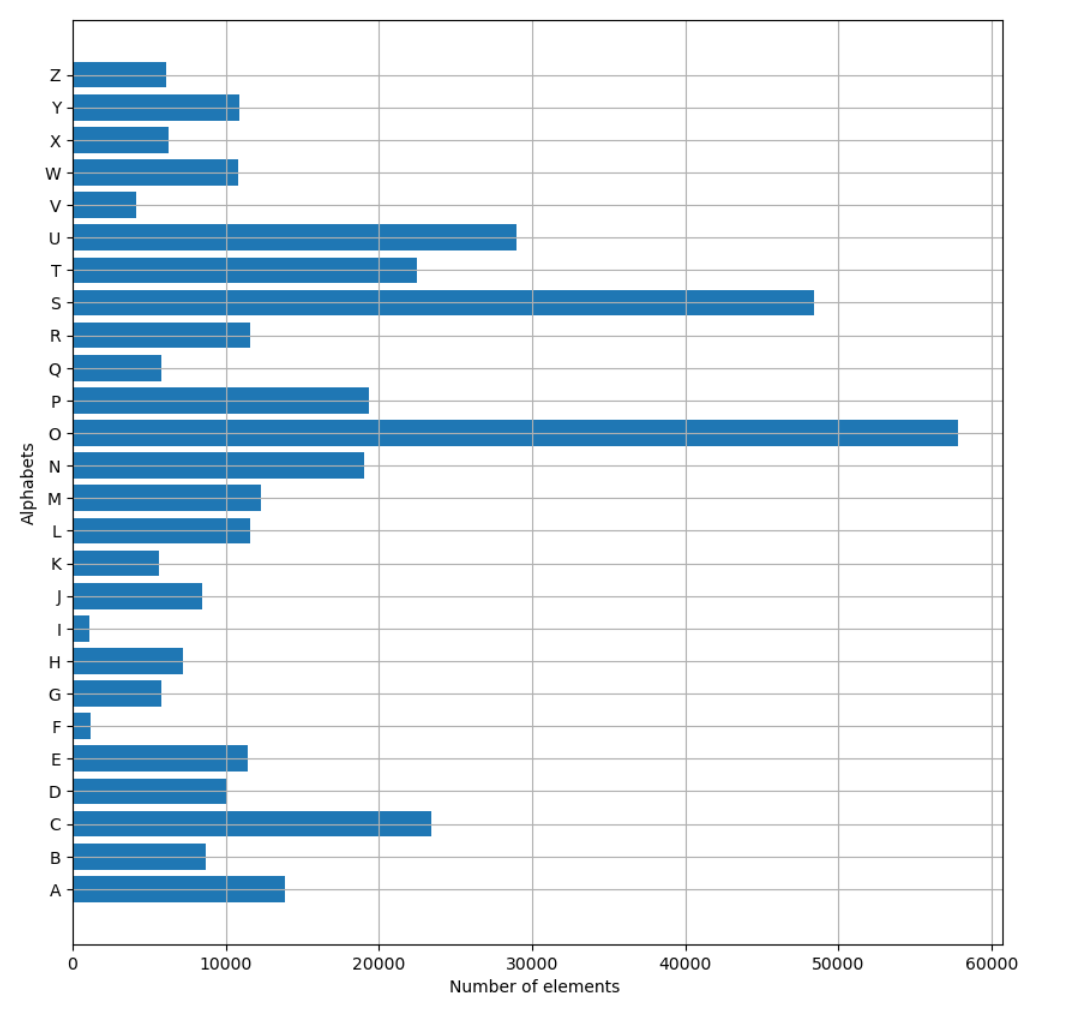
*Hình 12**. Sau khi huấn luyện*

**

*Hình 13**. Đồ thị loss, accuracy của training set và validation set theo epoch*

### Đánh giá mô hình

Theo kết quả nghiên cứu thì không thể có trường hợp nhận diện đúng 100% các chữ cái viết tay. Nguyên nhân có cả khách quan lẫn chủ quan. Khách quan là do mỗi người có một cách viết nét dày, mảnh khác nhau, một cách viết chữ khác nhau đối với mỗi chữ cái... Chủ quan là mô hình học không thể có độ chính xác 100%.



*Hình 14**. Biểu đồ hiển thị số lượng các chữ cái trong dữ liệu test*

Sau khi chạy mô hình trong tập dữ liệu test tỷ lệ chính xác là 99,46% và tỷ lệ thất bại là 2,80%.

Độ chính xác các chữ cái từ A đên Z:

A : 99.79% B : 99.06% C : 99.53% D : 97.44%

E : 99,87% F : 97.83% G : 98,91% H : 99,32%

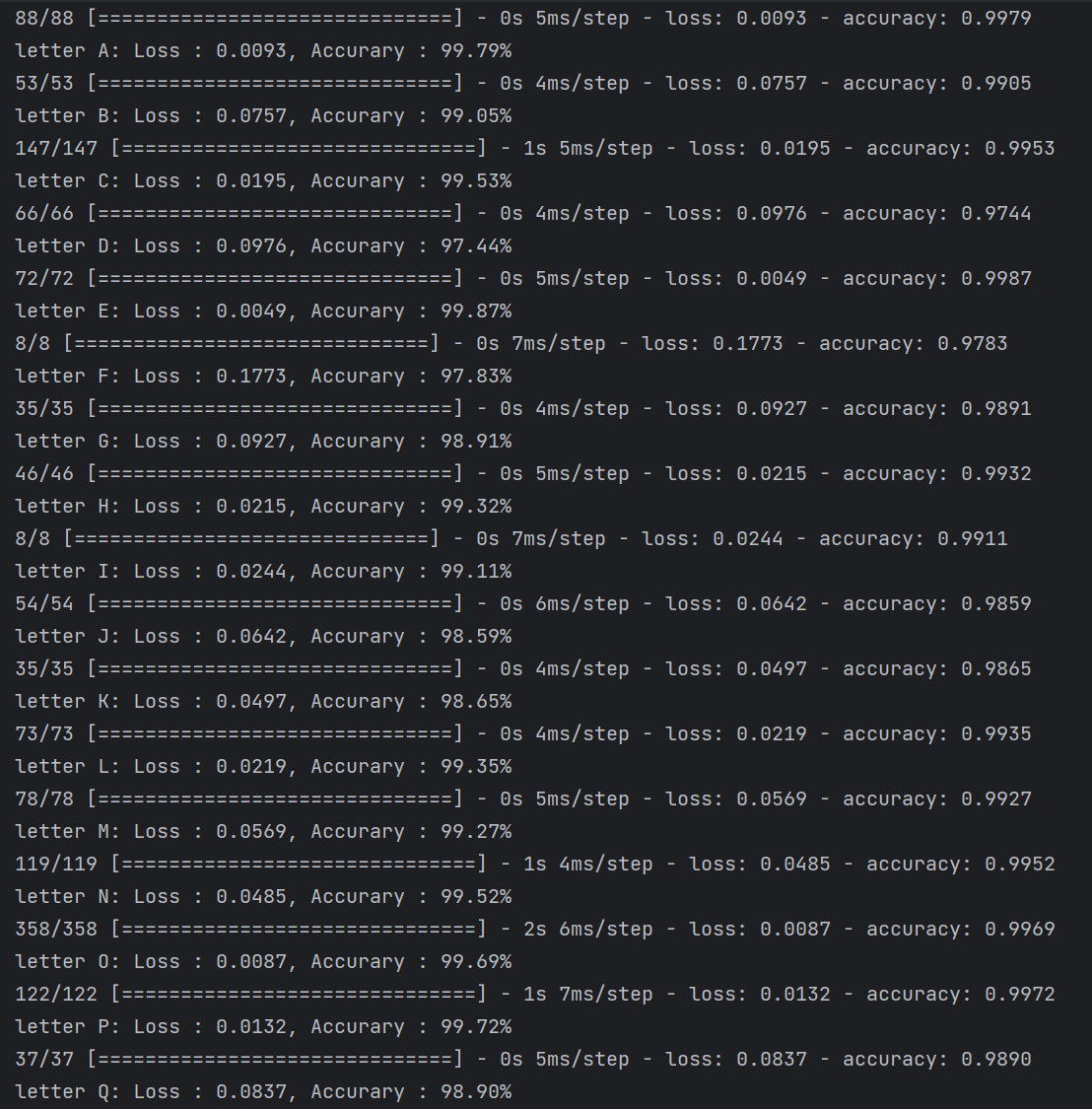
I : 99,11% J : 98,59% K : 98,65% L : 99,35%

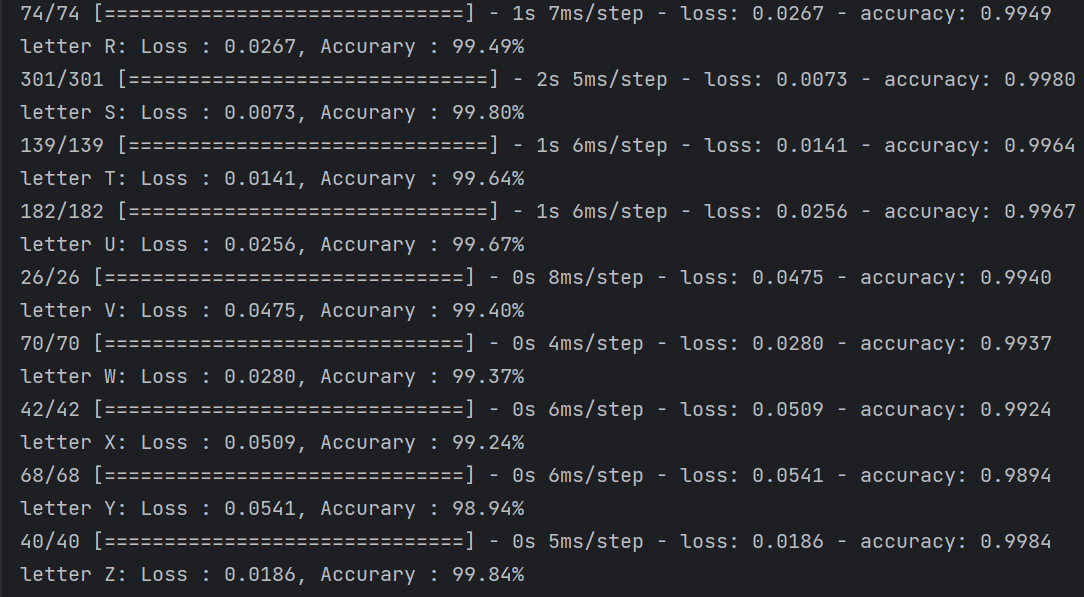
M: 99,27% N : 99,52% O : 99,69% P : 99,72%

Q : 99,90% R : 99,49% S : 99,80% T : 99,64%

U : 99,67% V: 99,40% W : 99,37% X : 99,24%

Y : 98,24% Z : 99,84%





*Hình 15**. Tỷ lệ chính xác các chữ cái*

## Code demo

### 3.3.1. Train dữ liệu

### - Khai báo thư viện và đọc dữ liệu từ Alphabets

### 

*Hình 16**. Khai báo thư viện và đọc dữ liệu*

### - Mô hình CNN

### 

*Hình 17**. Mô hình CNN*

### 

### - Huấn luyện mô hình và đánh giá từng chữ cái

### 

*Hình 18**. Huấn luyện và đánh giá chữ cái*

### - Lưu lại tệp train và đánh giá mô hình

### 

*Hình 19**. Lưu tệp và đánh giá*

### *-* Vẽ biểu đồ

### 

*Hình 20**. Vẽ biểu đồ*

### **3.3.2. Predict (Dự đoán)**

### **-** Tải ảnh lên, chuyển ảnh về màu xám và tìm đường nét ký tự

### 

*Hình 21**. Load ảnh, xử lý và tìm nét*

### - Chức năng dự đoán ký tự

### 

*Hình 22**. Chức năng dự đoán ký tự*

### - Duyệt contour, resize ảnh, vẽ hình và đặt ký tự dự đoán

### 

*Hình 23**. Duyệt contour, resize ảnh, vẽ hình và đặt ký tự*

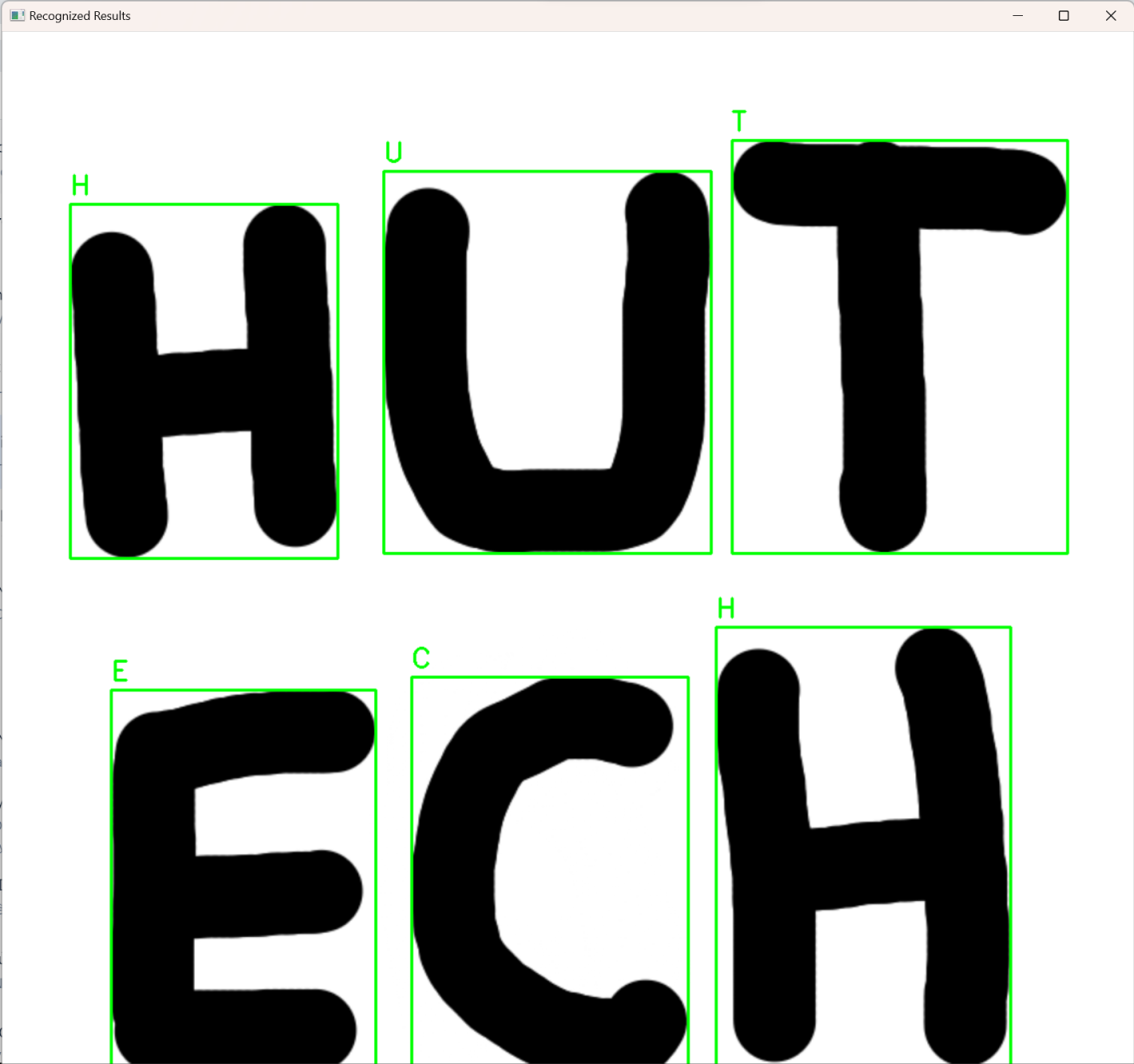
### - Hiển thị ảnh với ký tự dự đoán

### 

*Hình 24**. Hiện thị ảnh với ký tự dự đoán*

## Chương trình demo

* **Giao diện:**



*Hình 25**. Kết quả nhận dạng*

* **Chức năng:** Hiển thị kết quả nhận diện lên trên đầu các chữ số khi đưa hình ảnh trực tiếp bằng thư viện OpenCV**.**
* **Môi trường hoạt động của chương trình demo:** Được demo bằng ngôn ngữ Python trên phần mềm Pycharm phiên bản 2023.2.1

**CHƯƠNG 4: KẾT LUẬN**

**1. Kết quả đạt được**

### Qua việc tìm hiểu và nghiên cứu tìm hiểu về thuật toán CNN cũng như làm đồ án “nhận diện chữ cái viết tay” đã giúp nhóm chúng em có được những kiến thức cơ bản về thuật toán CNN để có thể xây dựng một số ứng dụng minh họa khác, còn có thẻ áp dụng vào một số bài toán cơ bản, giải quyết một số vấn đề cơ bản trong cuộc sống.

### Về đề tài “Nhận diện chữ cái viết tay” đã áp dụng thuật toán CNN và đạt được những kết quả ban đầu được đề ra. Tuy nhiên trong quá trình triển khai và tìm hiểu, nhóm chung em nhận thấy đề tài còn nhiều thiếu sót và nhưng hạn chế cần được khắc phục.

**2. Mặt hạn chế**

* Ứng dụng chưa thể nhận diện chính xác các chữ cái cũng như hình ảnh trong môi trường khác nhau.
* Vì kiến thức còn hạn chế nên ứng dụng chưa có giao diện.
* Chưa thể nhận diện chữ số.

**3. Hướng phát triển**

# Để phát triền đề tài tốt nhất, thông qua tìm hiểu từ các trang thông tin và cá nhân em có một số hướng phát triển như sau :

# Tìm hiểu các đề xuất cải tiến của thuật toán CNN cũng hư kiến trúc mô hình Alexnet, VGG-16, GoogleNet để khắc phục các hạn chế hiện có của CNN

# Nghiên cứu và tìm hiểu về nhận diện thêm chứ số để đưa vào ứng dụng có thể nhận diện được chữ có số và chữ cái.

# Tăng độ dự đoán của ứng dụng để có thể nhận biết một cách chính xác nhất

# Thiết kế giao điện thân thiện có chức năng gắn hình để mọi người có thể dễ tiếp cận và sự dụng trong tương lai.

# Bảng phân công

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ Tên | MSSV | Công Việc | Tiến độ |
| 1 | Lê Văn Khang | 2011063668 | Viết báo cáo,tìm code, viết code, chạy thuật toán, tìm hiểu thuật toán, tham khảo tài liệu. | 100 |

# TÀI LIỆU THAM KHẢO

1. <https://vietnix.vn/deep-learning-la-gi/>
2. <https://glints.com/vn/blog/deep-learning-la-gi/>
3. <https://www.youtube.com/watch?v=mVO_JWailTA&t=92s&pp=ygUibmjhuq1uIGRp4buHbiBjaOG7ryB2aeG6v3QgdGF5IGNubg%3D%3D>
4. <https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>
5. <https://www.youtube.com/watch?v=iU5Mjvu006M&pp=ygUiaGFuZHdyaXRpbmcgcmVjb2duaXRpb24gcHl0aG9uIGNubg%3D%3D>
6. <https://www.youtube.com/watch?v=p2wdxQzSks8&t=16s&pp=ygUibmjhuq1uIGRp4buHbiBjaOG7ryB2aeG6v3QgdGF5IGNubg%3D%3D>