**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙚🕮🙘**

**LẬP TRÌNH PYTHON**

**TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG**

**NHẬN DIỆN CHỮ VIẾT TAY**

**Giáo viên bộ môn:** Nguyễn Hải Triều

**Sinh viên thực hiện:**

1. Cao Nguyễn Hải Duy – 61132601
2. Huỳnh Đình Minh Trí – 61132212

Khánh Hòa, tháng 06 năm 2022

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙚🕮🙘**

**LẬP TRÌNH PYTHON**

**TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG**

**NHẬN DIỆN CHỮ VIẾT TAY**

**Giáo viên bộ môn:** Nguyễn Hải Triều

**Sinh viên thực hiện:**

1. Cao Nguyễn Hải Duy – 61132601
2. Huỳnh Đình Minh Trí – 61132212

Khánh Hòa, tháng 06 năm 2022

**Mục Lục**

[CHƯƠNG 1. GIỚI THIỆU DỰ ÁN 5](#_Toc106076235)

[1.1. Thông tin dự án 5](#_Toc106076236)

[1.2. Công cụ thực hiện dự án 5](#_Toc106076237)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 6](#_Toc106076238)

[2.1. Ngôn ngữ lập trình 6](#_Toc106076239)

[2.2. Các khái niệm cơ bản 6](#_Toc106076240)

[2.2.1. Deep Learning 6](#_Toc106076241)

[2.2.2. Neural Network 6](#_Toc106076242)

[2.2.3. Convolutional Neural Network (CNN) 7](#_Toc106076243)

[2.3. Nền tảng lập trình 7](#_Toc106076244)

[2.3.1. Visual Studio Code 7](#_Toc106076245)

[2.3.2. Google Colab 8](#_Toc106076246)

[CHƯƠNG 3. GIẢI QUYẾT BÀI TOÁN 9](#_Toc106076247)

[3.1. Hướng giải quyết cơ bản 9](#_Toc106076248)

[3.2. Xây dựng và huấn luyện mô hình trên Google Colab 9](#_Toc106076249)

[3.2.1. Thu thập và chuẩn bị dữ liệu 9](#_Toc106076250)

[3.2.2. Phân tích số liệu các tập dữ liệu 9](#_Toc106076251)

[3.2.3. Trực quan hóa dữ liệu 10](#_Toc106076252)

[3.2.4. Xử lý dữ liệu 10](#_Toc106076253)

[3.2.5. Xây dựng mô hình 10](#_Toc106076254)

[3.3. Thiết kế chương trình thử nghiệm mô hình 12](#_Toc106076255)

[3.3.1. Thiết kế giao diện và chức năng cơ bản 12](#_Toc106076256)

[3.3.2. Xử lý dữ liệu đầu vào 13](#_Toc106076257)

[CHƯƠNG 4. KẾT LUẬN 14](#_Toc106076258)

[4.1. Kết quả thu được 14](#_Toc106076259)

[4.2. Kết luận 15](#_Toc106076260)

[4.3. Các tài liệu tham khảo 15](#_Toc106076261)

1. GIỚI THIỆU DỰ ÁN
   1. Thông tin dự án

* Tên đề tài: Xây dựng hệ thống nhận diện chữ viết tay.
* Môn học: Lập trình Python.
* Sinh viên thực hiện:
  + Cao Nguyễn Hải Duy – 61132601 – Lớp 61.CNTT-CLC
  + Huỳnh Đình Minh Trí – 61132212 – Lớp 61.CNTT-CLC
* Mục tiêu dự án: Xây dựng được một hệ thống nhận diện được ký tự alphabet viết tay và dự đoán chính xác ký tự đấy là gì.
  1. Công cụ thực hiện dự án
* Ngôn ngữ lập trình: Python.
* Môi trường phát triển: Visual Studio Code, Google Colab.
* Mô hình áp dụng: Convolutional Neural Network (CNN).

1. CƠ SỞ LÝ THUYẾT
   1. Ngôn ngữ lập trình

* Python: hiện đang là ngôn ngữ lập trình phổ biến nhất. Python dễ học, dễ sử dụng, là ngôn ngữ mã nguồn và có nhiều thư viện hỗ trợ cho nhiều mảng khác nhau như phát triển website, phát triển game, khoa học dữ liệu, học máy, …
* Các thư viện Python được sử dụng:
  + Numpy, Pandas: Xử lý dữ liệu từ dataset.
  + Matplotlib: Trực quan hóa dữ liệu, mô phỏng dữ liệu dưới dạng ảnh.
  + Sci-kit Learn: Chia tập train – test và xáo trộn dữ liệu.
  + Keras: cung cấp API xây dựng mô hình và sử dụng backend Tensorflow.
  + Tensorflow: Xử lý dữ liệu đầu ra của dataset cho tương thích với dữ liệu đầu ra của model.
  + Tkinter: Thiết kế giao diện để thử nghiệm mô hình đã huấn luyện.
  + OpenCV (cv2): Xử lý hình ảnh.
  1. Các khái niệm cơ bản
     1. Deep Learning

Deep Learning là một tập con của Machine Learning, về cơ bản là một mạng neuron có ba lớp trở lên. Những mạng lưới thần kinh này cố gắng mô phỏng hành vi của não người, mặc dù không phù hợp với khả năng của nó cho phép nó “học” từ một lượng lớn dữ liệu. Mặc dù mạng neuron với một lớp duy nhất vẫn có thể đưa ra các dự đoán gần đúng, nhưng các lớp ẩn bổ sung có thể giúp tối ưu hóa và tinh chỉnh để có độ chính xác.

* + 1. Neural Network

Neural Network là mạng Neuron nhân tạo, sử dụng các mô hình toán học phức tạp để xử lý thông tin. Chúng dựa trên mô hình hoạt động của các tế bào thần kinh và khớp thần kinh trong não của con người. Tương tự như bộ não con người, mạng neuron nhân tạo kết nối các nút đơn giản, còn được gọi là tế bào thần kinh. Và một tập hợp các nút như vậy tạo thành một mạng lưới các nút, do đó có tên là mạng neuron nhân tạo.

Tương tự như bộ não con người, trong mạng nơ-ron nhân tạo, một loạt các thuật toán được sử dụng để xác định và nhận ra các mối quan hệ trong các tập dữ liệu. Mạng neuron nhân tạo được sử dụng trên nhiều công nghệ và ứng dụng khác nhau như trò chơi điện tử, thị giác máy tính, nhận dạng giọng nói, lọc mạng xã hội, dịch tự động và chẩn đoán y tế. Đáng ngạc nhiên là mạng nơ-ron nhân tạo được sử dụng cho các hoạt động truyền thống và sáng tạo, như hội họa và nghệ thuật.

* + 1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNNs – Mạng neuron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh.

Một Neural Network được xem là một mạng CNN khi nó có ít nhất một lớp Convolutional (lớp tích chập).

* 1. Nền tảng lập trình
     1. Visual Studio Code

Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

Visual Studio Code hiện nay đang rất được ưa chuộng bởi nó có những ưu điểm sau đây:

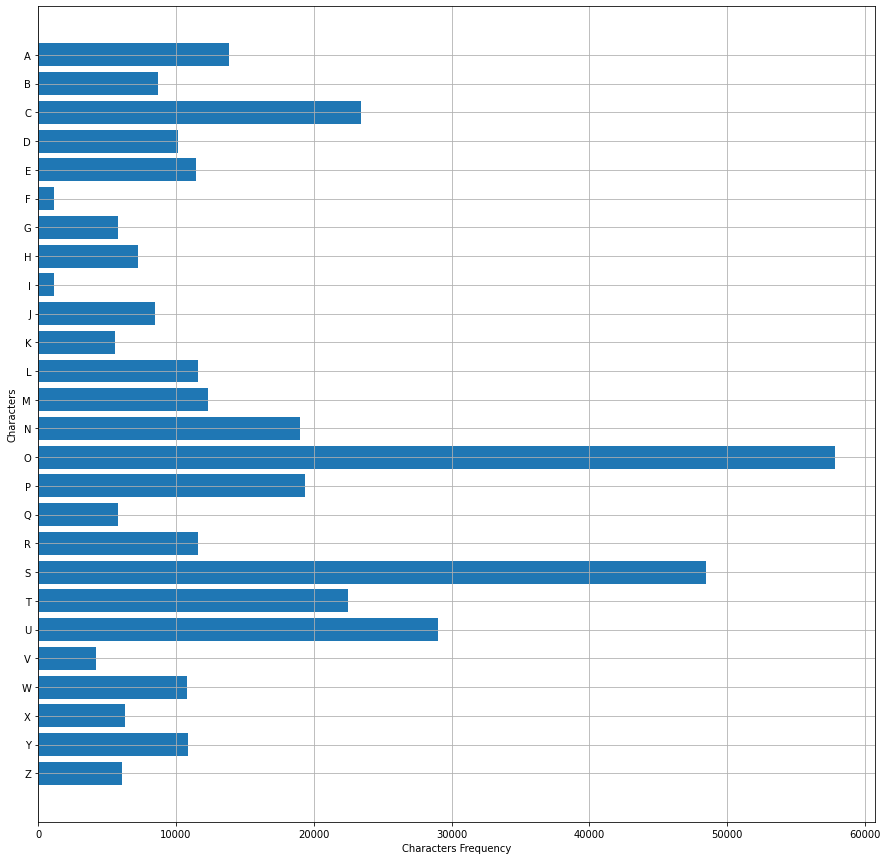
* Hỗ trợ đa nền tảng: Windows, Linux, Mac.
* Hỗ trợ đa ngôn ngữ: C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript, JSON, …
* Ít dung lượng.
* Tính năng mạnh mẽ.
* Intellisense chuyên nghiệp.
* Giao diện thân thiện.
  + 1. Google Colab

Google Colaboratory (Colab) là nền tảng lập trình Jupyter notebook trực tuyến, cho phép người dùng lập trình Python trên trình duyệt, và chạy trên một máy chủ của Google.

Colab không yêu cầu người dùng phải cài đặt gì trên máy tính cá nhân, và việc chạy trên hosted runtime của Google giúp giảm thiểu lượng tài nguyên sử dụng trên máy tính cá nhân của người dùng, rất phù hợp cho việc học tập và nghiên cứu các lĩnh vực Data Science hay Machine Learning, AI, …

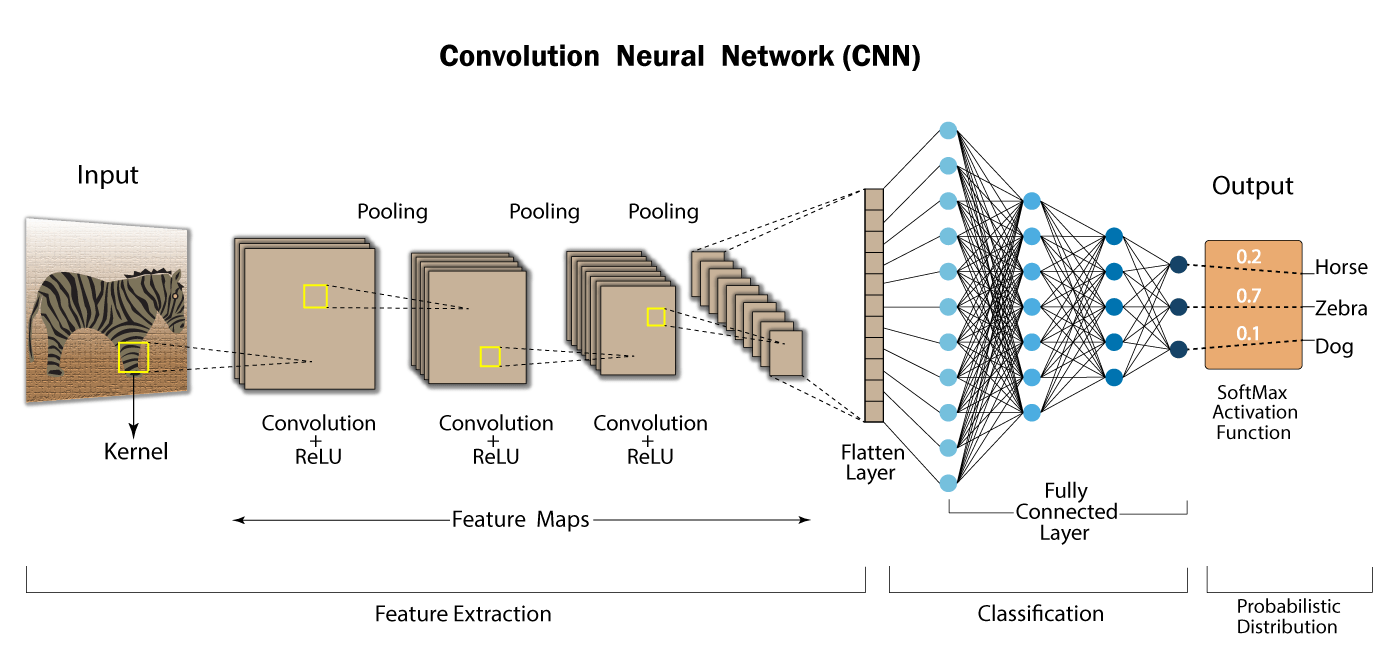
1. GIẢI QUYẾT BÀI TOÁN
   1. Hướng giải quyết cơ bản

* Thu thập và xử lý dữ liệu.
* Xây dựng mô hình CNN.
* Thử nghiệm và đánh giá mô hình bằng dữ liệu tập train và test.
* Xây dựng chương trình cho phép người dùng vẽ ký tự lên một canvas. Thử nghiệm và đánh giá mô hình bằng dữ liệu bên ngoài (ảnh ký tự vẽ trên canvas).
  1. Xây dựng và huấn luyện mô hình trên Google Colab
     1. Thu thập và chuẩn bị dữ liệu
* Nguồn dataset huấn luyện mô hình: <https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format>.
  + 1. Phân tích số liệu các tập dữ liệu
* Thống kê tổng quan về tập dữ liệu, bao gồm số lượng giá trị mất mát của từng thuộc tính.
* Xác định được thuộc tính đầu ra của tập dữ liệu.
  + 1. Trực quan hóa dữ liệu



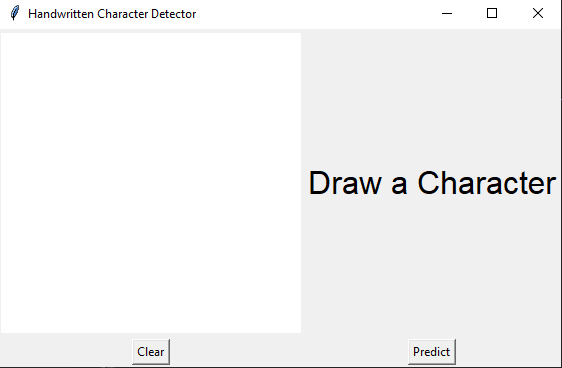
Hình 1 – Tần suất của các ký tự trong tập dữ liệu

* + 1. Xử lý dữ liệu
* Xử lý dữ liệu mất mát (nếu có).
* Chuẩn hóa dữ liệu đầu vào và đầu ra cho phù hợp với đầu vào và đầu ra của mô hình.
* Lưu dữ liệu đã qua xử lý thành file riêng (không bắt buộc).
  + 1. Xây dựng mô hình
* Chuẩn bị các tập dữ liệu đã được xử lý trước đó.



Hình 2 – Sơ đồ mạng CNN

* Thiết lập mô hình:
  + Feature Extraction: Sử dụng các lớp Convolution (tích chập), sau mỗi lớp Convolution là một lớp Pooling (tổng hợp).
  + Classification:
    - Sử dụng một hoặc nhiều lớp Dense để mô phỏng mạng lưới neuron.
    - Lớp Dense cuối cùng sẽ chứa số unit ngang với số class output của tập dữ liệu (đối với tập dữ liệu đã cho thì có 26 class tương ứng với 26 chữ cái trong bảng alphabet).
    - Lớp Dense cuối cùng sử dụng hàm activation SoftMax, trả về một mảng chứa các giá trị % khả năng input là một trong các class output (ví dụ: 0.46% là chữ A, 17.3% là chữ B, …, 6.29% là chữ Z).
* Xây dựng mô hình:
  + Truyền tập dữ liệu train vào mô hình.
  + Huấn luyện mô hình.
  + Lưu mô hình thành file.
  + Thực hiện dự đoán kết quả mô hình trên tập dữ liệu test.
  + Đánh giá độ chính xác của mô hình trên tập train và test.
  1. Thiết kế chương trình thử nghiệm mô hình
     1. Thiết kế giao diện và chức năng cơ bản
* Ta sử dụng Tkinter để xây dựng giao diện cơ bản gồm các thành phần sau:
  + Canvas trống nền trắng để vẽ ký tự lên.
  + Khu vực hiển thị kết quả dự đoán.
  + Nút bấm để dự đoán ký tự đã vẽ.
  + Nút bấm để xóa sạch canvas.
* Chức năng:
  + Chương trình sẽ load model từ file khi khởi động.
  + Người dùng sử dụng chuột để vẽ trên canvas (giữ chuột trái).
  + Sau khi vẽ xong và bấm nút dự đoán, chương trình sẽ chụp màn hình khu vực canvas lại và xử lý ảnh để giống với dữ liệu đầu vào đã được đưa vào huấn luyện mô hình.
  + Ảnh sau khi đã qua xử lý sẽ trở thành dữ liệu đầu vào để model dự đoán.
  + Output của model là một mảng chứa các giá trị % khả năng ảnh đầu vào là từng ký tự.
  + Ta chọn ra giá trị lớn nhất từ mảng cùng với ký tự gắn với index của giá trị đó trong mảng (0: A, 1: B, …, 25: Z) rồi hiển thị chúng trên giao diện.



Hình 3 – Giao diện chương trình

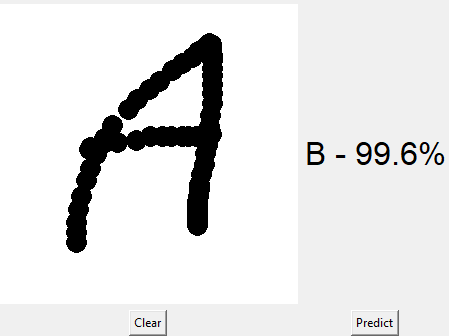
* + 1. Xử lý dữ liệu đầu vào
* Vì ảnh chụp canvas có kích thước và độ phân giải khác với ảnh được dùng để đưa vào huấn luyện model nên ta cần xử lý ảnh cho giống với ảnh huấn luyện model nhất có thể. Ở đây ta sẽ sử dụng thư viện OpenCV để xử lý.
* Ảnh huấn luyện model là ảnh grayscale có kích thước 28x28 nên trước tiên ta cần phải chuyển đổi ảnh chụp canvas thành ảnh grayscale.
* Sử dụng bộ lọc Gaussian Blur để làm mịn nét rồi sử dụng bộ lọc Threshold để loại bỏ các pixel nhiễu trong ảnh.
* Sau khi xử lý ảnh xong, ta sẽ resize ảnh thành kích thước 28x28 rồi sử dụng numpy để reshape thành mảng có kích thước (1, 28, 28, 1) (1 ảnh, 28x28, 1 kênh màu).

1. KẾT LUẬN
   1. Kết quả thu được

* Nhìn chung, mô hình có kết quả đạt được khá tốt trên 2 tập train và test và cả dữ liệu bên ngoài.

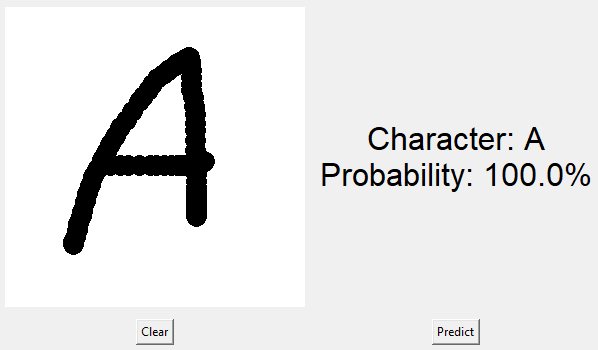
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Dense Layers** | **Dense Units** | **Training Accuracy (%)** | **Validation Accuracy (%)** | **Note** |
| v0 | 1 | 64 | 98.16 | 97.69 |  |
| v1 | 1 | 128 | 98.25 | 97.73 |  |
| v2 | 2 | 64, 64 | 97.5 | 97.11 |  |
| v2.1 | 2 | 64, 128 | 97.52 | 97.09 |  |
| v2.2 | 2 | 128, 128 | 97.94 | 97.48 |  |
| v3 | 3 | 64, 64, 64 | 97.8 | 97.47 |  |
| v3.1 | 3 | 64, 64, 128 | 97.93 | 97.49 |  |
| v3.2 | 3 | 64, 128, 128 | 97.92 | 97,47 |  |
| v3.3 | 3 | 128, 128, 128 | 97.39 | 96.91 |  |
| Final | 3 | 128, 128, 128 | 98.06 | 97.78 | ● Last convolution layer uses "valid" padding instead of the default "same". ● Filters for each Convolution layer are 32, 64 and 128. |

* Đối với dữ liệu bên ngoài, việc xử lý ảnh chụp canvas là điều quan trọng nhất quyết định độ chính xác của kết quả dự đoán:
  + Nếu chỉ resize và reshape ảnh thì kết quả dự đoán gần như không bao giờ chính xác.



Hình 4 – Kết quả dự đoán trước khi xử lý ảnh

* + Nếu xử lý ảnh trước khi truyền cho mô dự đoán thì kết quả dự đoán sẽ cải thiện rõ rệt.



Hình 5 – Kết quả dự đoán sau khi xử lý ảnh

* Tuy nhiên, đối với dữ liệu bên ngoài thì kết quả nhận dạng của một số ký tự không được chính xác (đặc biệt là các ký tự F, I, U dễ bị dự đoán nhầm thành các ký tự khác).
  1. Kết luận

Cần phải có sự đồng nhất giữa dữ liệu dùng để train model và dữ liệu đưa vào để model dự đoán để có thể tối ưu hóa khả năng và kết quả dự đoán của mô hình.

* 1. Các tài liệu tham khảo
* Slide bài giảng trên eLearning.
* Tài liệu chính thức của các thư viện Python đã dùng.
* Hướng dẫn thiết lập mô hình CNN cơ bản: <https://www.tensorflow.org/tutorials/images/cnn>.
* Tổng quan về CNN, hướng dẫn chọn tham số cho CNN: <https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/>.