TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN

![A logo for a university

Description automatically generated]()

**BÁO CÁO GIỮA KỲ**

***Xây dựng ứng dụng nhận diện chữ viết tay***

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | Đỗ Quốc Bình | **Đỗ Thế Toàn** | **Nguyễn Đại Nghĩa** |
| MSSV | 21012040 | **21012524** | **21012886** |
| Email | 21012040@st.phenikaa-uni.edu.vn | **21012524**@st.phenikaa-uni.edu.vn | **21012886**@st.phenikaa-uni.edu.vn |

**28/05/2024**

## Mô Tả Thuật Toán

* **Trainning tập dữ liệu MNIST**

**Model:** đã được sử dụng thuật toán **Convolutional Neural Network (CNN)** để huấn luyện mô hình nhận dạng chữ số viết tay MNIST.

**Chi tiết các thành phần và thuật toán:**

1. **CNN:**
   * Là một loại mạng nơ-ron nhân tạo chuyên dùng cho xử lý ảnh, được thiết kế để tự động và thích ứng trích xuất các đặc trưng quan trọng từ dữ liệu hình ảnh.
   * Hoạt động dựa trên các lớp **Conv2D** (lớp tích chập) và **MaxPooling2D** (lớp tổng hợp).
     + Lớp tích chập: áp dụng các bộ lọc (kernel) lên hình ảnh để trích xuất các đặc trưng (đường nét, góc cạnh...).
     + Lớp tổng hợp: giảm kích thước của hình ảnh sau khi tích chập, giúp giảm số lượng tham số và tăng tính khái quát của mô hình.
2. **Flatten:**
   * Chuyển đổi dữ liệu đầu ra từ các lớp tích chập và tổng hợp thành dạng vector để đưa vào các lớp Dense (lớp kết nối đầy đủ).
3. **Dense:**
   * Là các lớp kết nối đầy đủ, thực hiện các phép biến đổi tuyến tính và phi tuyến để đưa ra dự đoán cuối cùng.
4. **Dropout:**
   * Kỹ thuật regularization giúp ngăn chặn overfitting (hiện tượng mô hình học thuộc dữ liệu huấn luyện), bằng cách ngẫu nhiên loại bỏ một số nơ-ron trong quá trình huấn luyện.
5. **Adam:**
   * Thuật toán tối ưu hóa được sử dụng để điều chỉnh các tham số của mô hình trong quá trình huấn luyện, giúp mô hình hội tụ nhanh chóng và đạt độ chính xác cao.
6. **Categorical Crossentropy:**
   * Hàm mất mát được sử dụng để đánh giá sự khác biệt giữa dự đoán của mô hình và nhãn thực tế, từ đó hướng dẫn quá trình tối ưu hóa các tham số.
7. **Softmax:**
   * Hàm kích hoạt được sử dụng ở lớp đầu ra, giúp chuyển đổi các giá trị đầu ra thành các xác suất, cho biết khả năng hình ảnh đầu vào thuộc về mỗi lớp (từ 0 đến 9).

**Tổng kết:** Đã sử dụng CNN để trích xuất đặc trưng từ hình ảnh, sau đó sử dụng các lớp Dense và Dropout để phân loại các chữ số viết tay. Quá trình huấn luyện sử dụng thuật toán Adam và hàm mất mát Categorical Crossentropy để tối ưu hóa mô hình.

* **Tiền xử lý ảnh**

Đã được sử dụng kết hợp nhiều kỹ thuật xử lý ảnh để tiền xử lý ảnh chứa chữ số viết tay, chuẩn bị cho quá trình nhận dạng sau này. Các thuật toán chính được sử dụng bao gồm:

1. **Chuyển đổi ảnh xám (Grayscaling):**
   * Thuật toán chuyển đổi ảnh màu RGB thành ảnh xám bằng cách tính trung bình giá trị của ba kênh màu đỏ (R), xanh lá cây (G) và xanh dương (B) tại mỗi pixel.
2. **Làm mịn ảnh bằng Gaussian Blur:**
   * Thuật toán làm mờ ảnh bằng cách sử dụng bộ lọc Gaussian, giúp giảm nhiễu và loại bỏ các chi tiết nhỏ không quan trọng.
3. **Tìm ngưỡng Otsu:**
   * Thuật toán tự động tìm ngưỡng tối ưu để phân tách ảnh xám thành ảnh nhị phân (đen trắng), dựa trên phân bố cường độ sáng của các pixel.
4. **Tìm Contours:**
   * Thuật toán tìm các đường bao quanh các vùng liên thông trong ảnh nhị phân.
5. **Lọc Contours:**
   * Sử dụng các tiêu chí như diện tích và tỷ lệ khung hình để loại bỏ các contours không phải là chữ số.
6. **Sắp xếp Contours:**
   * Sắp xếp các contours từ trái sang phải dựa trên tọa độ x của chúng.
7. **Tạo vùng quan tâm (ROI):**
   * Cắt ra vùng chứa chữ số từ ảnh gốc, có thể thêm vùng đệm xung quanh để đảm bảo không bị mất thông tin.
8. **Thay đổi kích thước ảnh:**
   * Thay đổi kích thước ROI về kích thước chuẩn 28x28 để phù hợp với đầu vào của mô hình nhận dạng chữ số.

**Tổng kết:** đã sử dụng các kỹ thuật xử lý ảnh cơ bản để tách các chữ số viết tay từ ảnh đầu vào, chuẩn hóa kích thước và định dạng của chúng, giúp cải thiện độ chính xác của quá trình nhận dạng sau này.

## Mô Tả Ứng Dụng

Dưới đây là quy trình hoạt động chi tiết của ứng dụng:

1. **Khởi tạo ứng dụng:**
   * app = Flask(\_\_name\_\_): Tạo một đối tượng Flask đại diện cho ứng dụng web.
   * model\_path: Đường dẫn đến tệp tin mô hình đã được huấn luyện trước đó.
   * load\_model(model\_path): Tải mô hình từ tệp tin đã lưu vào bộ nhớ.
   * UPLOAD\_FOLDER: Thư mục lưu trữ tạm thời các ảnh tải lên.
2. **Hàm preprocess\_image:**
   * Hàm này được định nghĩa trong tệp image\_processing.py (không có trong đoạn code cung cấp) và thực hiện tiền xử lý ảnh đầu vào.
   * Nó chuyển đổi ảnh thành ảnh xám, làm mịn, tìm ngưỡng, phát hiện các contours (đường bao) của các chữ số, lọc và sắp xếp chúng.
   * Sau đó, nó cắt ra từng chữ số thành các vùng quan tâm (ROI), thay đổi kích thước về 28x28 và trả về danh sách các ảnh chữ số đã được xử lý.
3. **Hàm predict\_digit:**
   * Hàm này nhận một ảnh chữ số đã xử lý, chuẩn hóa dữ liệu và đưa vào mô hình để dự đoán.
   * Kết quả dự đoán là nhãn lớp có xác suất cao nhất, tương ứng với chữ số từ 0 đến 9.
4. **Trang chủ (/)**:
   * @app.route('/'): Khi người dùng truy cập trang chủ của ứng dụng, hàm index sẽ được gọi.
   * render\_template('index.html', prediction=None): Hiển thị giao diện trang chủ index.html với giá trị prediction là None (chưa có dự đoán).
5. **Xử lý dự đoán (/predict)**:
   * @app.route('/predict', methods=['POST']): Khi người dùng tải lên ảnh và nhấn nút gửi, yêu cầu POST sẽ được gửi đến đường dẫn /predict, và hàm predict sẽ được gọi.
   * Hàm kiểm tra xem có tệp tin ảnh được tải lên không.
   * Nếu có, lưu ảnh vào thư mục UPLOAD\_FOLDER.
   * Gọi hàm preprocess\_image để xử lý ảnh và tách các chữ số.
   * Duyệt qua từng chữ số, gọi hàm predict\_digit để dự đoán và lưu kết quả vào danh sách predictions.
   * Ghép các dự đoán thành một chuỗi và trả về kết quả dưới dạng JSON ({'prediction': result}).
6. **Chạy ứng dụng:**
   * if \_\_name\_\_ == '\_\_main\_\_':: Khởi chạy ứng dụng ở chế độ debug.

**Tóm tắt quy trình:**

1. Người dùng truy cập trang chủ.
2. Tải lên ảnh chứa chữ số viết tay.
3. Ảnh được xử lý và tách thành các chữ số riêng lẻ.
4. Mô hình dự đoán từng chữ số.
5. Kết quả dự đoán được hiển thị vào ô text box trên trang web.

## Kết quả đánh giá mô hình

- Accuracy: 99,8%

- Loss: 0.006%

## Hình Ảnh Minh Họa

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

Màn hình giao diện ban đầu

Ảnh có chứa văn bản, phần mềm, ảnh chụp màn hình, Phần mềm đa phương tiện

Mô tả được tạo tự động

Tải ảnh lên từ máy tính

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

Kết quả sau khi chạy

## Kế hoạch dự án

|  |  |
| --- | --- |
| **Nội dung công việc** | **Tên** |
| Xây dựng mô hình | Đỗ Quốc Bình |
| Huấn luyện mô hình | Đỗ Thế Toàn |
| Nghiên cứu đề tài |
| Thiết kế sản phẩm | Nguyễn Đại Nghĩa |
| Làm báo cáo |
| Toàn bộ dự án | 100% |

**Bảng đề xuất chấm điểm**

|  |  |
| --- | --- |
| Tên | Đề xuất điểm |
| Đỗ Quốc Bình | 10 |
| Đỗ Thế Toàn | 10 |
| Nguyễn Đại Nghĩa | 10 |