**TRƯỜNG ĐẠI HỌC HỌC VĂN LANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

A red and black shield with white logo

Description automatically generated

**BÁO CÁO MÔN HỌC**

**PHÂN TÍCH DỮ LIỆU VÀ HỌC SÂU**

NGÀNH: CÔNG NGHỆ THÔNG TIN

***Tên Đồ án:***

**NHẬN DIỆN CHỮ SỐ VIẾT TAY BẰNG MÔ HÌNH CNN**

**SVTH: Trần Minh Phúc - 2274802010694**

**SVTH: Châu Gia Kiệt - 2274802010449**

**SVTH: Nguyễn Hoàng Khanh - 2274802010393**

**SVTH: Võ Đình Ngọc Bình - 2274802010066**

**GVHD: Trần Kim Mỹ Vân**

**LHP:**

**GVHD: Phan Gia Phước**

**TP. Hồ Chí Minh – năm 2025**

MỤC LỤC

[I. GIỚI THIỆU 3](#_Toc194321679)

[1. Bối cảnh 3](#_Toc194321680)

[2. Lý do chọn đề tài 3](#_Toc194321681)

[3. Lý do chọn đề tài 3](#_Toc194321682)

[II. Tổng quan lý thuyết 3](#_Toc194321683)

[1. CNN là gì? 3](#_Toc194321684)

[2. Kiến trúc CNN cơ bản 3](#_Toc194321685)

[3. Dữ liệu hình ảnh 4](#_Toc194321686)

[III. Mô hình nhận diện 5](#_Toc194321687)

[1. Dữ liệu 5](#_Toc194321688)

[2. Tiền xử lý dữ liệu 5](#_Toc194321689)

[3. Kiến trúc mô hình CNN 5](#_Toc194321690)

[IV. Kết quả & đánh giá: 6](#_Toc194321691)

[1. Kết quả huấn luyện mô hình CNN tự xây dựng 6](#_Toc194321692)

[V. Kết luận 6](#_Toc194321693)

[TÀI LIỆU THAM KHẢO 7](#_Toc194321694)

# I. GIỚI THIỆU

## 1. Bối cảnh

Trong thời đại công nghệ 4.0, trí tuệ nhân tạo (AI) và học sâu (Deep Learning) ngày càng được ứng dụng rộng rãi, đặc biệt trong thị giác máy tính (Computer Vision). Bài toán nhận diện chữ viết tay là một ví dụ điển hình được ứng dụng trong giao diện người - máy, kiểm tra dữ liệu số hóa, và phân loại thông tin.

## 2. Lý do chọn đề tài

Bài toán nhận diện chữ số viết tay từ tập dữ liệu MNIST là một bài toán đơn giản nhưng hiệu quả để hiểu và thực hành quy trình huấn luyện một mô hình học sâu CNN.

## 3. Lý do chọn đề tài

* Tìm hiểu về Convolutional Neural Networks (CNN).
* Áp dụng CNN vào bài toán nhận diện chữ số viết tay với tập dữ liệu MNIST.
* Xây dựng và huấn luyện mô hình CNN đơn giản.
* Đánh giá hiệu quả mô hình qua độ chính xác và biểu đồ học.

# II. Tổng quan lý thuyết

## 1. CNN là gì?

Convolutional Neural Networks(CNN) là một loại mạng thần kinh nhân tạo chuyên biệt để xử lý dữ liệu hình ảnh. CNN có khả năng tự động trích xuất đặc trưng từ hình ảnh, giảm thiểu yêu cầu về tiền xử lý dữ liệu so với các phương pháp truyền thống. Cấu trúc của CNN bao gồm các lớp tích chập (Convolutional Layer), lớp phi tuyến (Activation Function), lớp pooling (Pooling Layer) và lớp kết nối đầy đủ (Fully Connected Layer).

## 2. Kiến trúc CNN cơ bản

Một mô hình CNN điển hình bao gồm các thành phần chính sau:

* **Lớp tích chập (Convolutional Layer):** Lớp này thực hiện phép toán tích chập trên ảnh đầu vào bằng các bộ lọc (kernel) để trích xuất các đặc trưng quan trọng.
* **Lớp phi tuyến (Activation Function - ReLU):** Áp dụng hàm kích hoạt như ReLU để đưa mô hình vào trạng thái phi tuyến và tăng cường khả năng học của mạng.
* **Lớp pooling (Pooling Layer):** Thực hiện phép toán giảm kích thước dữ liệu nhưng vẫn giữ nguyên thông tin quan trọng, thường sử dụng Max Pooling hoặc Average Pooling.
* **Lớp kết nối đầy đủ (Fully Connected Layer - FC):** Kết nối toàn bộ các neuron từ các lớp trước để dự đoán nhãn của hình ảnh đầu vào.
* **Lớp Softmax:** Biến đầu ra thành xác suất dự đoán cho từng lớp.

## 3. Dữ liệu hình ảnh

Tập dữ liệu MNIST bao gồm 60.000 ảnh huấn luyện và 10.000 ảnh kiểm tra. Mỗi ảnh là ảnh đen trắng kích thước 28x28 pixel, biểu diễn chữ số viết tay từ 0 đến 9

**Hình ảnh của nhiều loại số khác nhau:**

A number and a number

AI-generated content may be incorrect..

**Tiền xử lý dữ liệu:** Các bước tiền xử lý như chuẩn hóa hình ảnh, tăng cường dữ liệu (Data Augmentation), và chia tập dữ liệu thành tập huấn luyện, kiểm tra và đánh giá nhằm tăng độ chính xác của mô hình.

# III. Mô hình nhận diện

## 1. Dữ liệu

Sử dụng dữ liệu từ thư viện tf.keras.datasets.mnist, bao gồm ảnh và nhãn của chữ số viết tay 0–9. Dữ liệu được chia thành tập huấn luyện (x\_train, y\_train) và tập kiểm tra (x\_test, y\_test).

## 2. Tiền xử lý dữ liệu

* Ảnh được reshape từ (28, 28) thành (28, 28, 1) để tương thích với đầu vào mạng CNN.
* Dữ liệu ảnh được chuẩn hóa từ khoảng 0–255 về [0, 1].

## 3. Kiến trúc mô hình CNN

Mô hình nhận diện ảnh sử dụng một mạng nâng cao dựa trên CNN (Convolutional Neural Network). Các lớp chính trong kiến trúc bao gồm:

* Conv2D(28 kernel 3x3, ReLU)
* MaxPooling2D(pool size 2x2)
* Flatten()
* Dense(128, activation=ReLU)
* Dropout(0.2)
* Dense(10, activation=softmax)

Mô hình được biên dịch với:

* Loss: sparse\_categorical\_crossentropy
* Optimizer: Adam
* Metrics: Accuracy

Mô hình được huấn luyện với 20 epochs trên toàn bộ tập dữ liệu huấn luyện.

# IV. Kết quả & đánh giá:

## 1. Kết quả huấn luyện mô hình CNN tự xây dựng

* Mô hình có tốc độ hội tụ tốt.
* Biểu đồ loss giảm dần và accuracy tăng dần ổn định qua mỗi epoch.
* Sau 20 epochs, độ chính xác trên tập kiểm tra đạt mức cao (>98%).
* Thời gian huấn luyện nhanh, phù hợp với các mô hình đơn giản.

A black square with white number three

AI-generated content may be incorrect.

# V. Kết luận

Qua quá trình nghiên cứu và thực hiện, nhóm đã xây dựng thành công một mô hình CNN đơn giản nhưng hiệu quả cho bài toán nhận diện chữ số viết tay sử dụng tập dữ liệu MNIST. Mô hình đạt độ chính xác cao, huấn luyện nhanh và ổn định, cho thấy tính ứng dụng tốt trong các hệ thống nhận dạng văn bản số. Đề tài giúp nhóm sinh viên hiểu rõ hơn về quy trình phát triển một mô hình học sâu, từ việc xử lý dữ liệu, thiết kế mô hình, huấn luyện và đánh giá kết quả. Trong tương lai, mô hình này có thể được mở rộng cho các bài toán phức tạp hơn như nhận diện ký tự tiếng Việt, chữ viết tay hoặc biển số xe.

# VI. Danh Sách Phân Công

|  |  |
| --- | --- |
| **Tên Thành Viên** | **Nhiệm Vụ** |
| Trần Minh Phúc | Viết code |
| Châu Gia Kiệt | Train + model |
| Nguyễn Hoàng Khanh | Viết file báo cáo |
| Võ Đình Ngọc Bình | Tìm tập dữ liệu |

# TÀI LIỆU THAM KHẢO

Wikipedia. (2021, 10 14). *Cơ sở dữ liệu MNIST*. (MNIST, Nhà sản xuất) Đã truy lục 03 31, 2025, từ Wikipedia: https://vi.wikipedia.org/wiki/C%C6%A1\_s%E1%BB%9F\_d%E1%BB%AF\_li%E1%BB%87u\_MNIST