

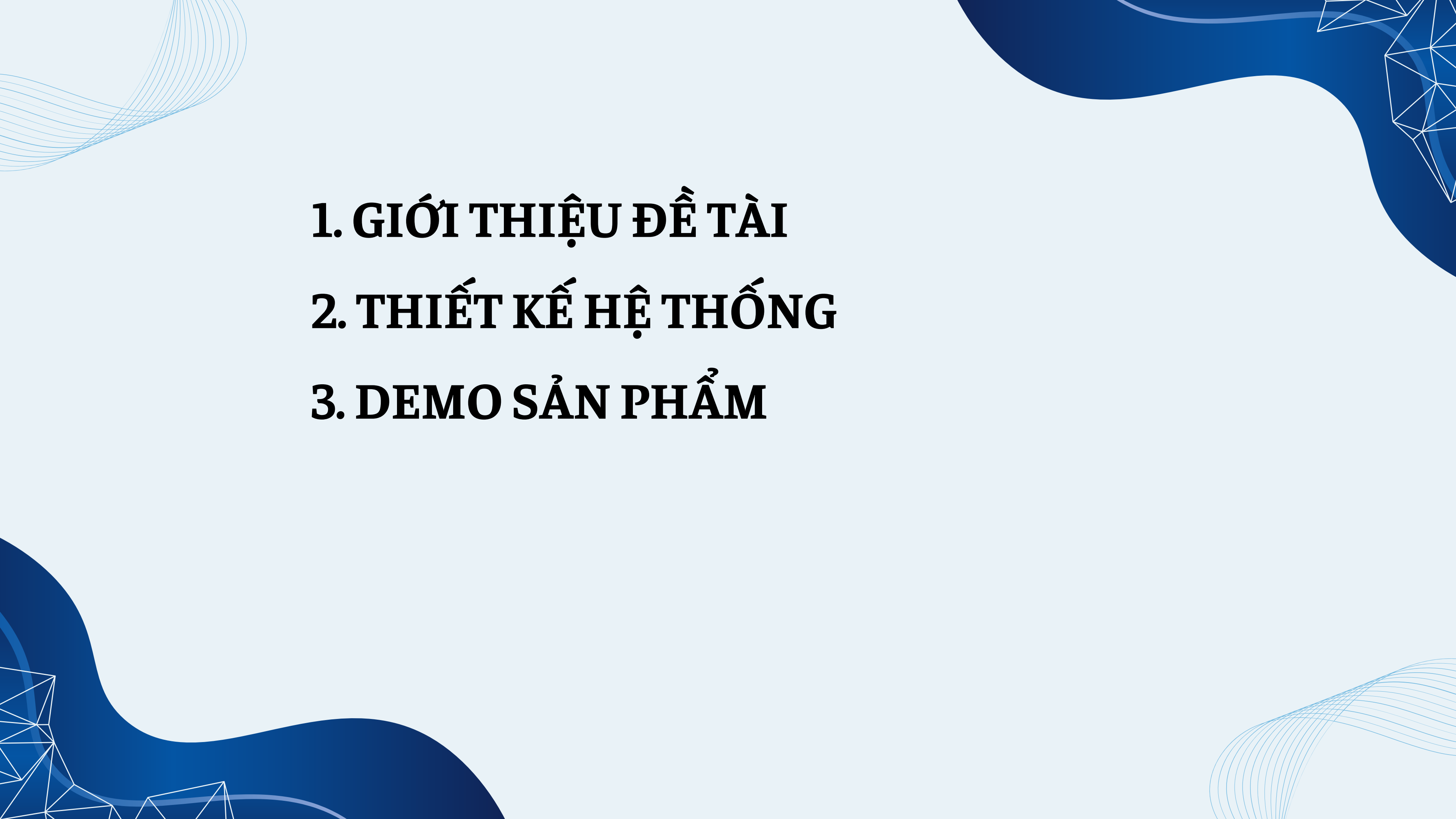
# **Nhận diện chữ viết và hình dạng đơn giản bằng mạng neural**

## **Thành viên nhóm**

Hoàng Việt Anh - B22DCCN017

Phan Văn Thủy - B22DCCN844

Lê Vũ Thành Vinh - B22DCCN904

- 
- 1. GIỚI THIỆU ĐỀ TÀI**
  - 2. THIẾT KẾ HỆ THỐNG**
  - 3. DEMO SẢN PHẨM**

# 1. GIỚI THIỆU ĐỀ TÀI

Lý do, mục tiêu

# LÝ DO CHỌN ĐỀ TÀI

- Computer Vision phát triển mạnh trong thời đại AI.
  - Nhu cầu nhận dạng ảnh: chữ viết tay, hình dạng, ký tự... ngày càng tăng cao
- Đề tài “Xây dựng hệ thống nhận dạng chữ viết tay và hình học sử dụng Mạng Neural Tích chập (CNN)” được lựa chọn và ứng dụng vào 2 bài toán
- Nhận dạng chữ số viết tay
  - Nhận dạng hình học

# MỤC TIÊU HỆ THỐNG

- Huấn luyện mô hình CNN cho từng bài toán nhận dạng
- Nhận dạng chữ số viết tay từ 0 đến 9 dựa trên bộ dữ liệu MNIST
- Nhận dạng các hình học cơ bản như hình tròn, hình vuông, tam giác ,...

## 2. THIẾT KẾ HỆ THỐNG

Xử lý ảnh, mô hình

# KIẾN TRÚC HỆ THỐNG

- Hệ thống được thiết kế theo mô hình phân tầng (3-Tier Architecture):
  - Presentation Layer (Streamlit): Giao diện tương tác (Vẽ Canvas, Upload).
  - Business Logic Layer:
    - `image_processor.py`: Tiền xử lý ảnh.
    - `shape_generator.py`: Sinh dữ liệu huấn luyện.
    - Controller điều phối luồng dữ liệu.
  - Model & Data Layer: Chứa các file model nhận diện và dữ liệu dataset.



## TIỀN XỬ LÝ ẢNH

- Chuyển ảnh sang grayscale
- Resize về kích thước chuẩn
  - 28x28 cho MNIST model
  - 64x64 cho Shape model
- Chuẩn hóa pixel về [0,1]
- Reshape thành tensor  $H \times W \times 1$ 
  - 28 x 28 x 1 cho MNIST model
  - 64 x 64 x 1 cho Shape model



# LUỒNG HOẠT ĐỘNG CỦA HỆ THỐNG KHI VẼ TAY

- Người dùng vẽ lên st\_canvas (Streamlit).
- Canvas trả về ảnh RGBA dưới dạng NumPy array.
- Tách thành ảnh xám duy nhất (lấy kênh alpha hoặc kênh màu).
- Tiền xử lý ảnh
- Đưa ảnh vào mô hình tương ứng (MNISTModel hoặc ShapeModel).
- Đưa ra kết quả dự đoán

# LUỒNG HOẠT ĐỘNG CỦA HỆ THỐNG KHI UPLOAD ẢNH

- Người dùng upload file ảnh
- File ảnh được đọc vào bộ nhớ thông qua thư viện PIL (Python Imaging Library)
- Chuyển ảnh sang NumPy array.
- Tiền xử lý ảnh
- Đưa ảnh đã xử lý vào mô hình tương ứng
- Mô hình dự đoán và trả kết quả.

# KIẾN TRÚC MÔ HÌNH CNN MNIST MODEL

- Input: ảnh kích thước 28 x 28 x 1
- Khối trích xuất đặc trưng 1
  - Conv2D(32, 3×3) + ReLu
  - MaxPool2D(2×2)
  - BatchNormalization
    - Học các đặc trưng cơ bản như góc, cạnh
- Khối trích xuất đặc trưng 2
  - Conv2D(64, 3×3) + ReLu
  - MaxPool2D(2×2)
  - BatchNormalization
    - Học các đặc trưng phức tạp hơn như kết hợp các cạnh thành hình dạng

- Khối trích xuất đặc trưng 3
  - Conv2D(64, (3, 3)) + ReLu
  - BatchNormalization
  - Học được hình dạng tổng thể của đối tượng
- Khối phân loại
  - Flatten → Dropout(0.5) → Dense(128) → Dropout(0.3) → Dense(10, Softmax)
  - Đưa ra kết quả dự đoán xem ảnh thuộc số nào.

## MÔ HÌNH SINH DỮ LIỆU HÌNH HỌC (SHAPEGENERATOR)

- Do thiếu hụt các bộ dữ liệu hình học vẽ tay chuẩn, class ShapeGenerator được thiết kế để tự động sinh dữ liệu huấn luyện tổng hợp
- Dữ liệu được sinh bằng thư viện Python PIL (Pillow) kết hợp với Numpy
  - Các hàm: Circle, Square, Triangle, Pentagon, Hexagon, v.v.
  - Các hình được vẽ dưới dạng khối đặc màu trắng trên nền đen
  - Sử dụng kỹ thuật tăng cường dữ liệu
    - Xoay ảnh ngẫu nhiên
    - Thêm nhiễu hạt để mô phỏng nhiễu thực tế

# KIẾN TRÚC MÔ HÌNH CNN SHAPE MODEL

- Input: ảnh kích thước  $64 \times 64 \times 1$
- Khối trích xuất đặc trưng 1
  - Conv2D(32)×2 + ReLU
  - MaxPool2D(2×2)
  - BatchNorm + Dropout(0.25)
  - Học được các đặc trưng cơ bản như cạnh thẳng, cạnh cong,...
- Khối trích xuất đặc trưng 2
  - Conv2D(32)×2 + ReLU
  - MaxPool2D(2×2)
  - BatchNorm + Dropout(0.25)
  - Học được các đặc trưng mức trung như ghép các cạnh thành hình dạng,...



- Khối trích xuất đặc trưng 3
  - Conv2D(128)×2 + ReLU
  - MaxPool2D(2×2)
  - BatchNorm + Dropout(0.25)→ Học được cấu trúc toàn cục, nhận diện hình dạng hoàn chỉnh
- Khối phân loại
  - Flatten → Dense(256) → Dropout(0.5) → Dense(128) → Dropout(0.3)  
→ Dense(8, Softmax)→ Đưa ra kết quả dự đoán là hình dạng gì trong 8 loại hình dạng



### **3. DEMO SẢN PHẨM**



**THANK YOU**