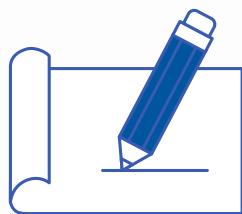
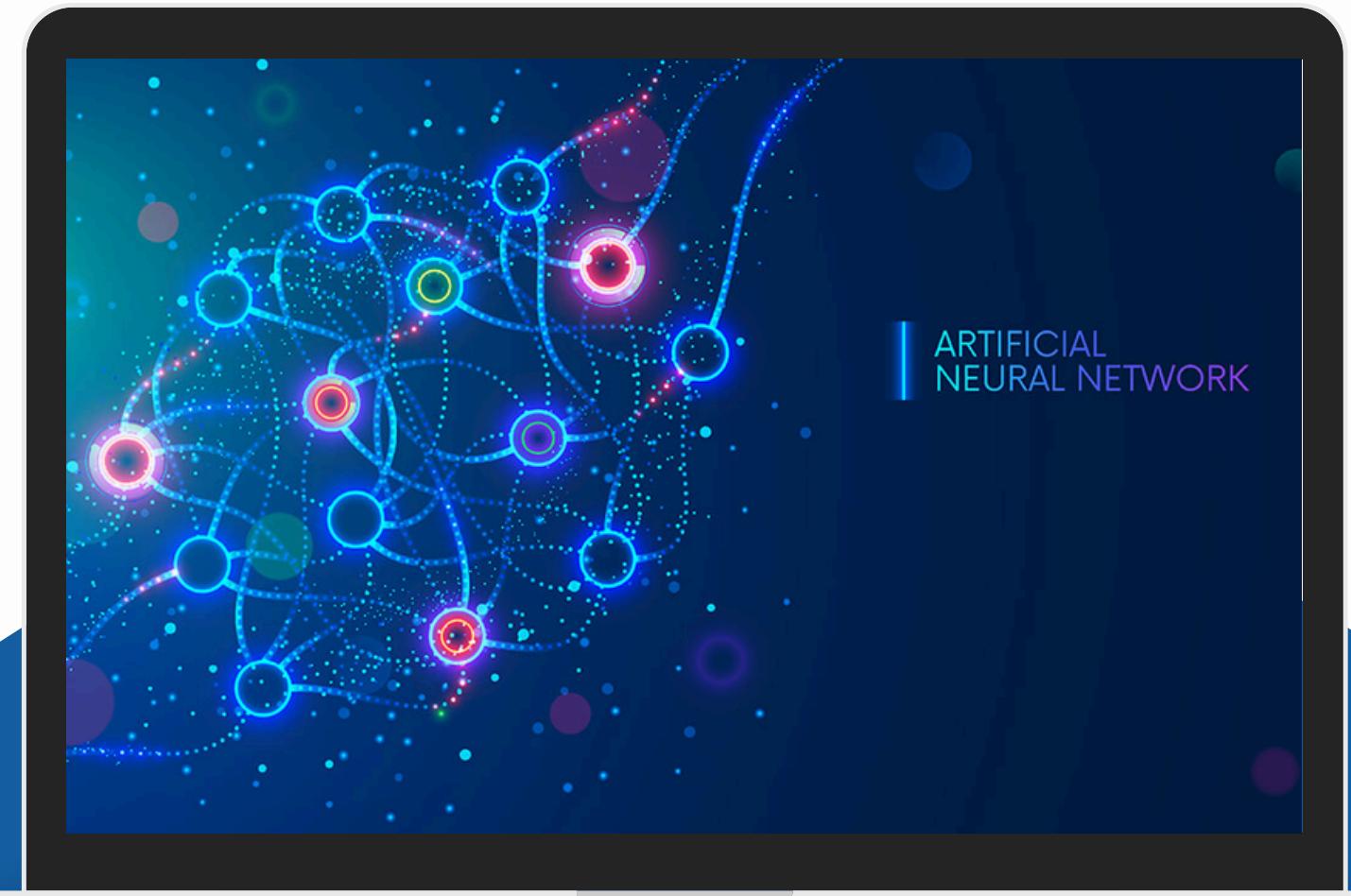


# NHẬN DẠNG CHỮ VIẾT VÀ HÌNH DẠNG ĐƠN GIẢN BẰNG MẠNG NEURAL



xử lý ảnh

Nhóm 7 – D22CNPM02



# Thành viên



Nguyễn Thị Khánh Vân - B22DCCN892



Trần Mai Hương - B22DCCN424

# Đề tài

I

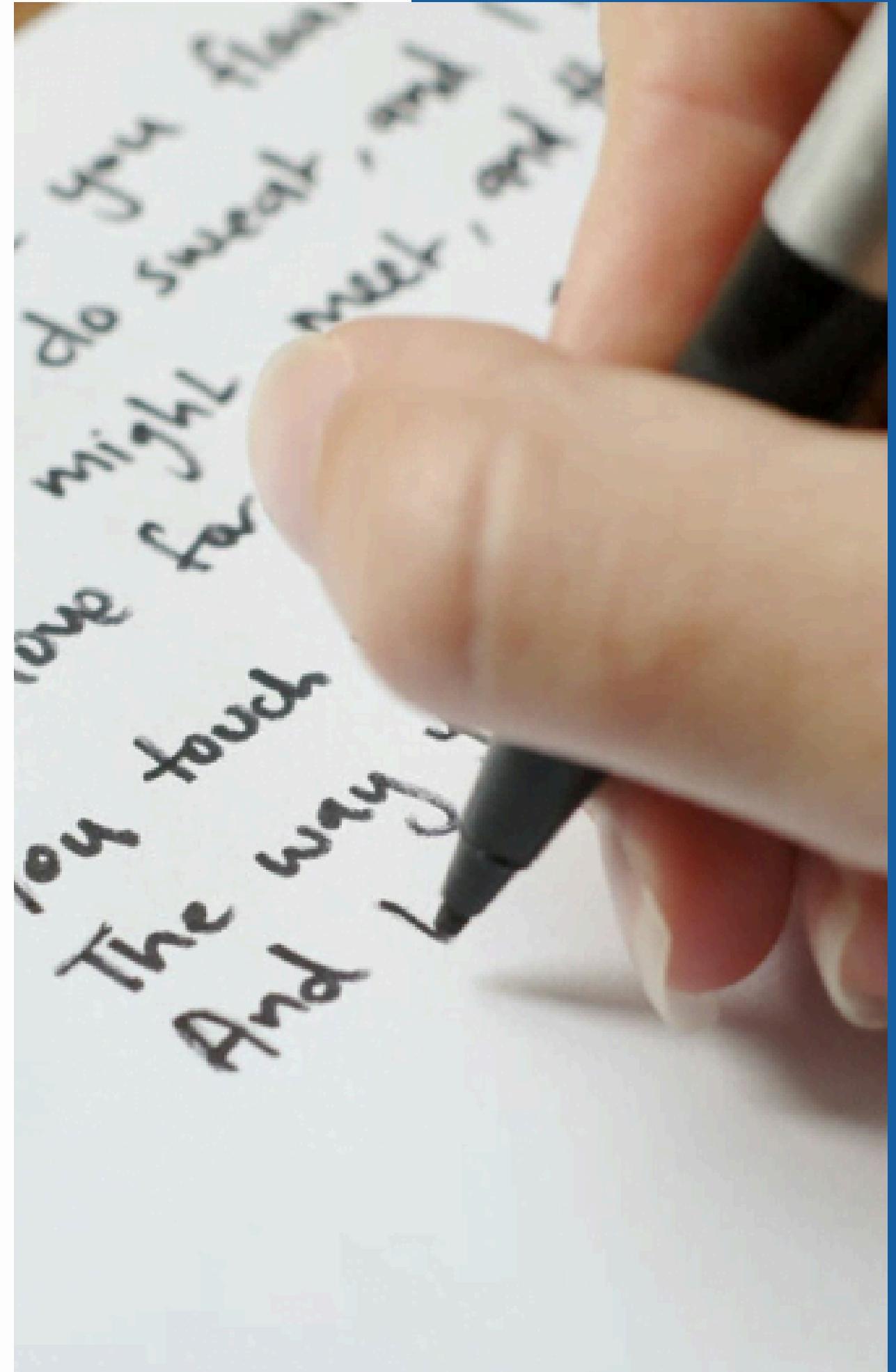
Nhận dạng chữ viết

II

Nhận dạng hình dạng đơn giản

# I. Nhận dạng chữ viết

- ▶ 1. Giới thiệu đề tài
- ▶ 2. Giải thích thuật toán
- ▶ 3. Mô tả dữ liệu
- ▶ 4. Luồng hoạt động
- ▶ 5. Kết quả



# I. Nhận dạng chữ viết

## 1. Giới thiệu đề tài

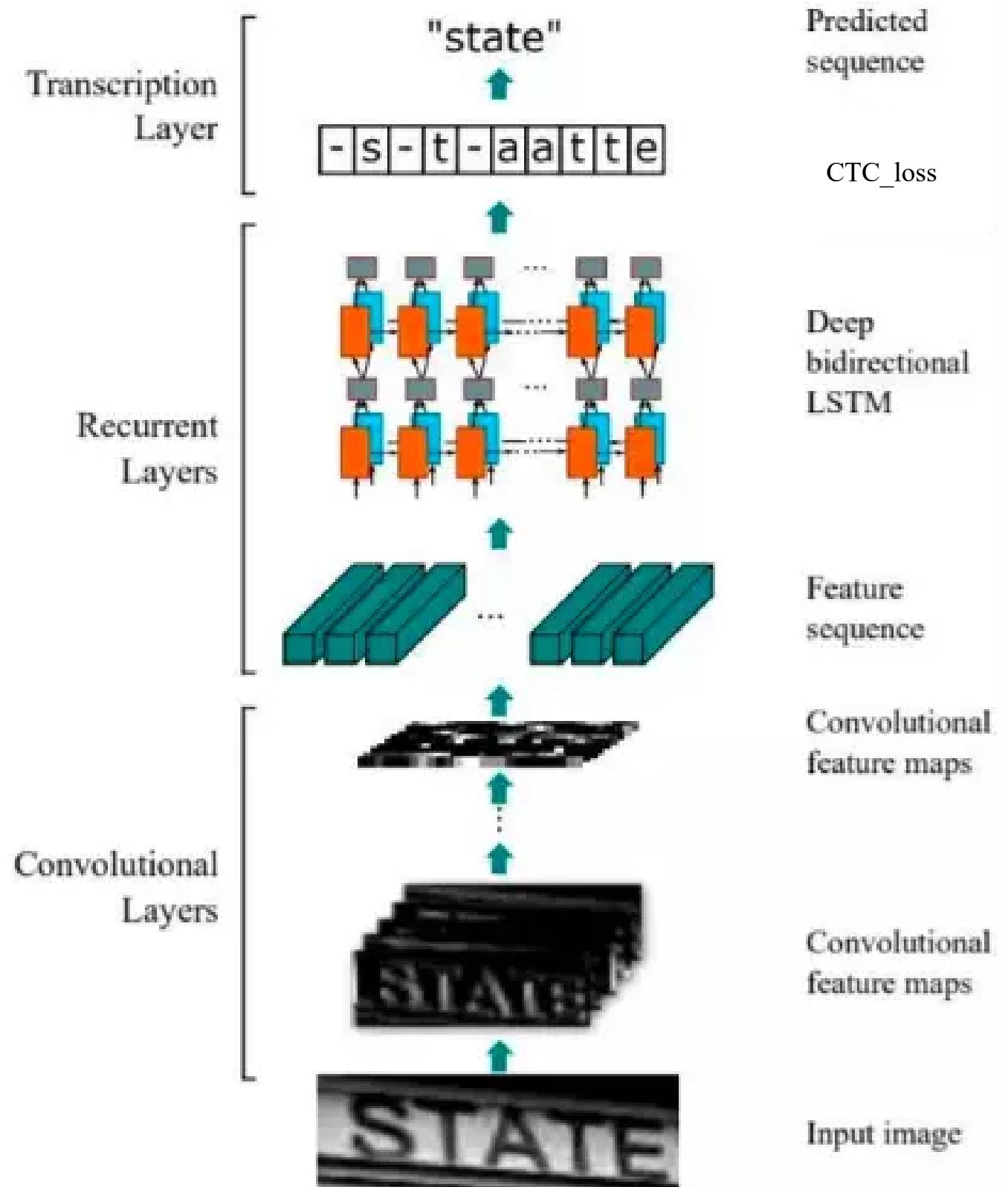
- **Vấn đề:** Nhận diện chữ viết tay (Handwritten Text Recognition - HTR) là thách thức lớn trong OCR, do biến đổi hình dạng chữ, độ nghiêng, và liên kết ký tự.
- **Mục tiêu:** Xây dựng mô hình CRNN có khả năng nhận diện chuỗi ký tự (words) viết tay một cách liên tục, đạt được độ chính xác cao trong việc chuyển đổi ảnh đầu vào thành văn bản số.
- **Công nghệ sử dụng:** CRNN (Conv + RNN + CTC Loss)
- **Ứng dụng:** Ứng dụng trong số hóa tài liệu, trợ lý ảo, giáo dục



# I. Nhận dạng chữ viết

## 2. Giải thích thuật toán

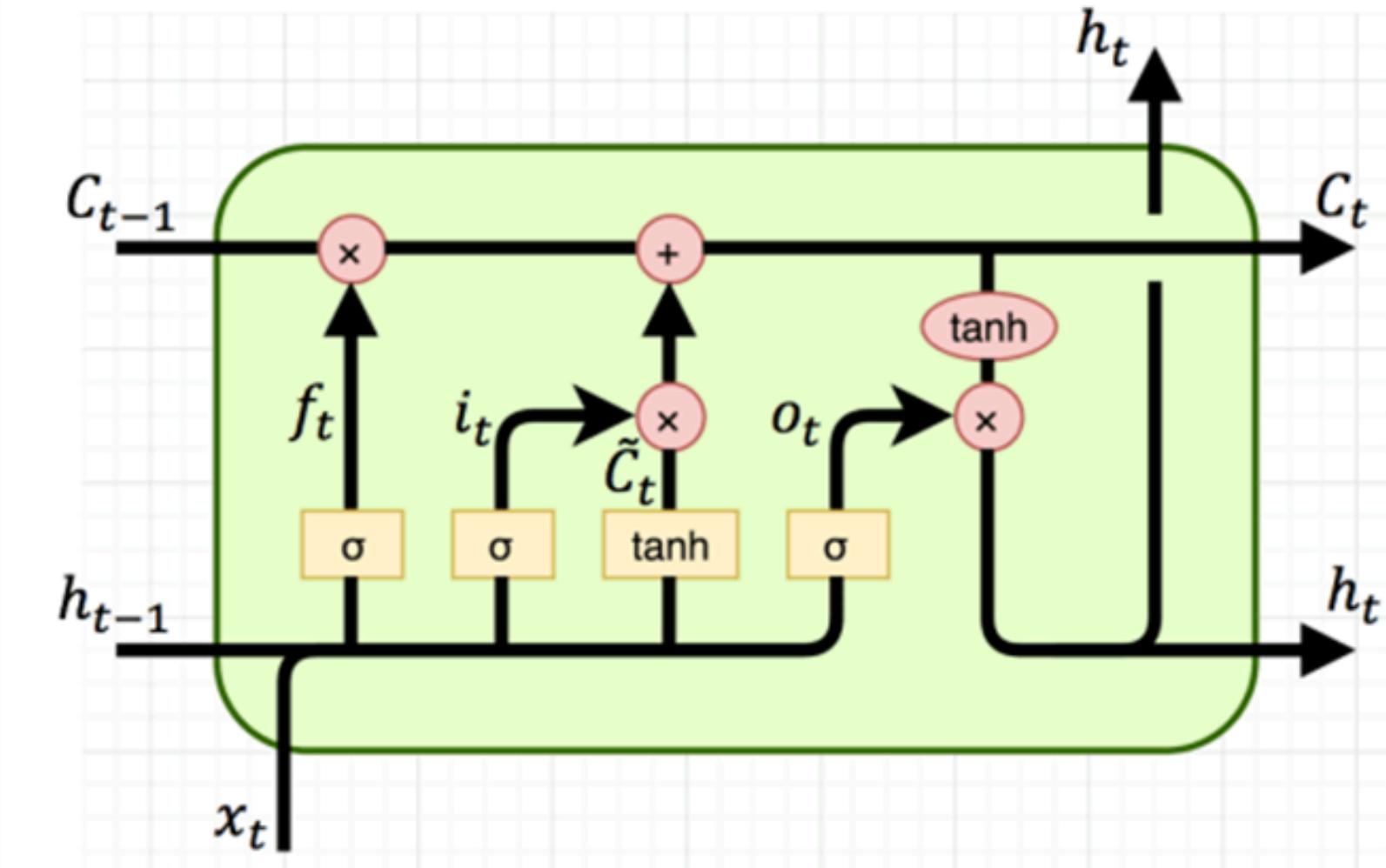
- **CRNN (Convolutional Recurrent Neural Network):**
  - **CNN** (trích xuất đặc trưng hình ảnh)
    - Trích xuất đặc trưng theo các timestep
  - **RNN (LSTM/BiLSTM)**: dựa vào output của CNN nắm bắt thông tin ngữ cảnh để dự đoán đầu ra tuần tự cho từng timestep.
- **CTC Loss**: Xử lý alignment tự động giữa input ảnh và output chuỗi (e.g., bỏ qua khoảng trắng, lặp ký tự).
  - Ưu điểm: Không cần bounding box cho từng chữ; hiệu quả với chuỗi dài.



# I. Nhận dạng chữ viết

## 2. Giải thích thuật toán

- **LSTM (Long Short-Term Memory)** – khắc phục vanishing gradient của RNN bằng Cell State
  - LSTM thêm một thành phần mới: Cell State ( $C_t$ ), đây là đường dẫn thông tin chạy xuyên suốt các bước thời gian.
  - Cell state được truyền gần như không thay đổi (chỉ nhân với các gate  $\approx 1$ )
  - LSTM có 3 gate:
    - Forget Gate – quên thông tin không quan trọng
    - Input Gate – thêm thông tin mới
    - Output Gate – quyết định phần nào dùng để tạo output

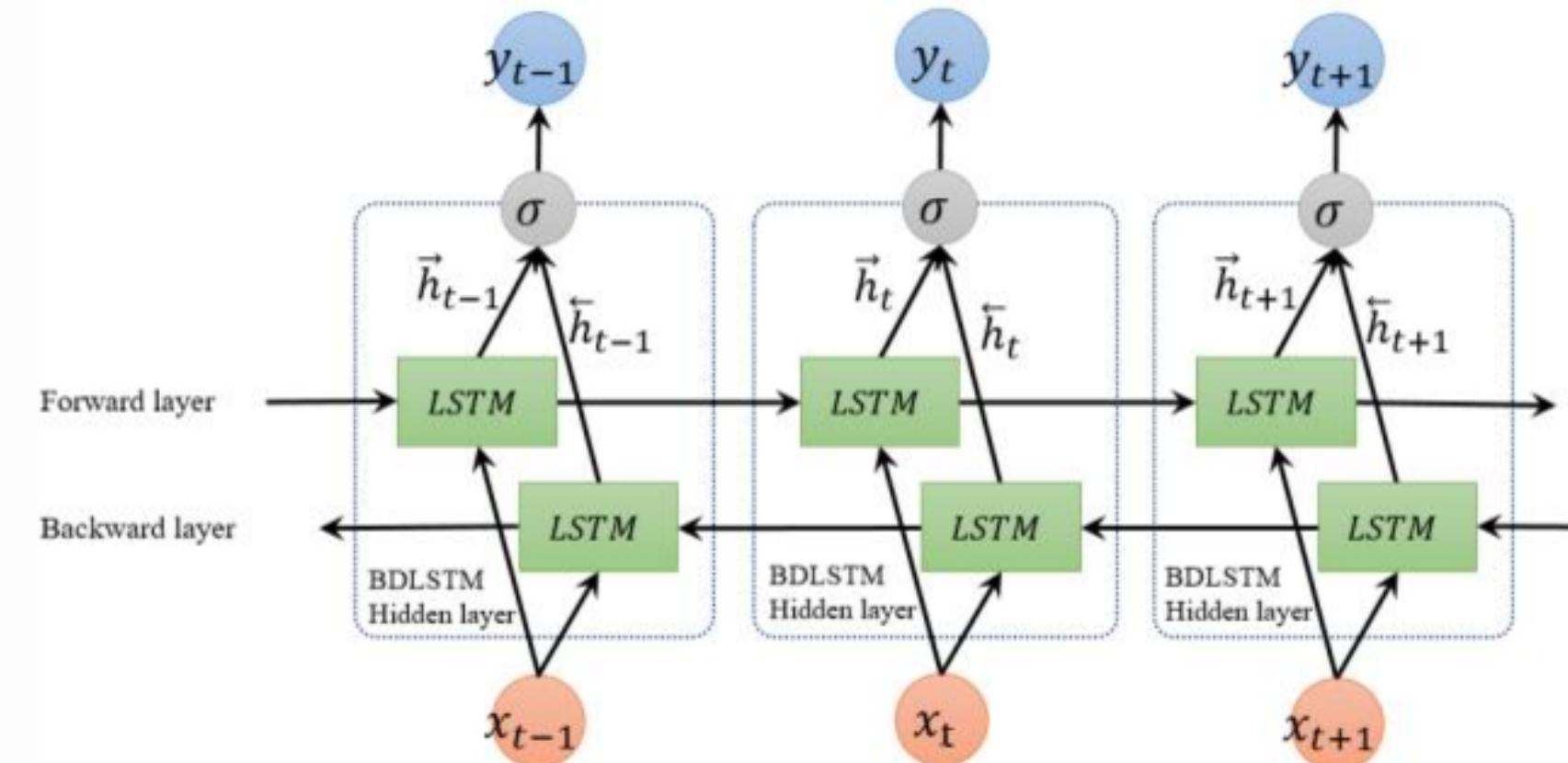


# I. Nhận dạng chữ viết

## 2. Giải thích thuật toán

- **BILSTM (Bidirectional long short term memory)**

- Là mở rộng của LSTM thông thường
- Kết hợp 2 LSTM chạy song song:
- Forward LSTM: Xử lý sequence từ đầu → cuối
- Backward LSTM: Xử lý sequence từ cuối → đầu
- Kết hợp output từ cả 2 chiều để tạo final representation



# I. Nhận dạng chữ viết

## 3. Mô tả dữ liệu

- **Nguồn:** IAM Handwriting Database (Kaggle/ HuggingFace) –
  - ~340k ảnh dòng chữ viết tay tiếng Anh
  - từ 657 người viết.
- **Định dạng:** Ảnh grayscale, nhãn là chuỗi text (UTF-8, uppercase/lowercase + punctuation).
- **Phân chia:** Train (80%), Val (10%), Test (10%)
- **Đặc điểm:**
  - Biến đổi cao (độ nghiêng, kích thước chữ); preprocessing: grayscale, resize theo chiều rộng, pad sequence.
- **Thống kê:**
  - Số ký tự: 30 classes (A-Z, punctuation)

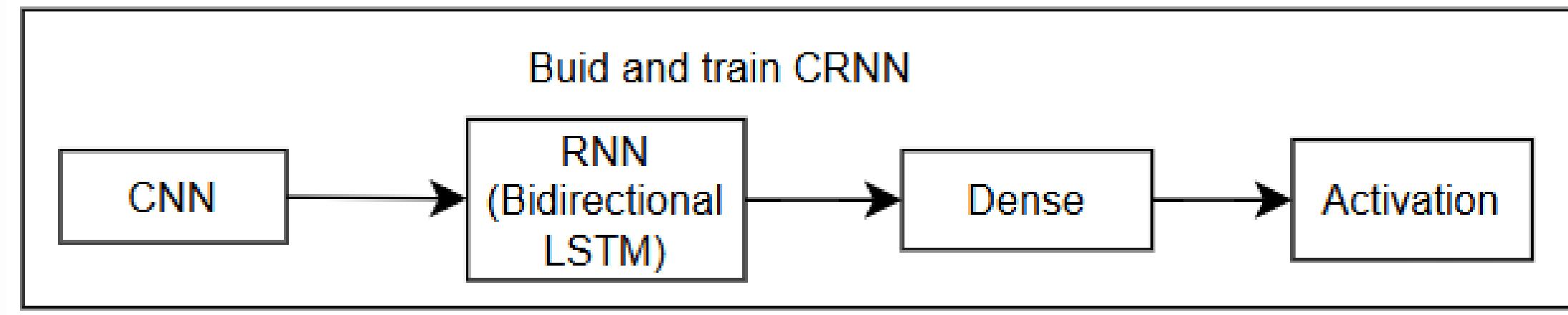
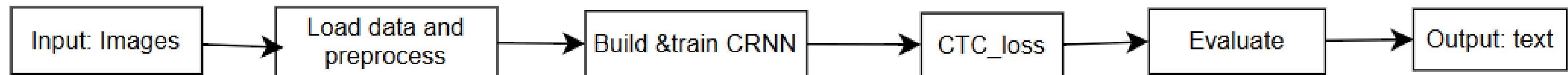
H A N E N T H I R A N

KEVIN

GORTCHAKOFF

# I. Nhận dạng chữ viết

## 4. Luồng hoạt động



# I. Nhận dạng chữ viết

## 5. Kết quả

# I. Nhận dạng hình dạng đơn giản

- ▶ 1. Giới thiệu đề tài
- ▶ 2. Giải thích thuật toán
- ▶ 3. Mô tả dữ liệu
- ▶ 4. Luồng hoạt động
- ▶ 5. Kết quả



# II. Nhận dạng hình dạng đơn giản

## 1. Giới thiệu đề tài

- Vấn đề: Nhận diện hình dạng cơ bản là nền tảng cho các ứng dụng thị giác máy tính như robot, xử lý ảnh y tế, hoặc tự động hóa.
- Mục tiêu: Xây dựng mô hình CNN để phân loại hình dạng đơn giản, kết hợp phát hiện vị trí hình dạng, nhận diện hình dạng đồ vật với độ chính xác cao.
- Công nghệ sử dụng: CNN (Keras/TensorFlow) cho phân loại, OpenCV cho phát hiện contour và bounding box.
- Ứng dụng: giáo dục, trò chơi, hoặc hệ thống giám sát.



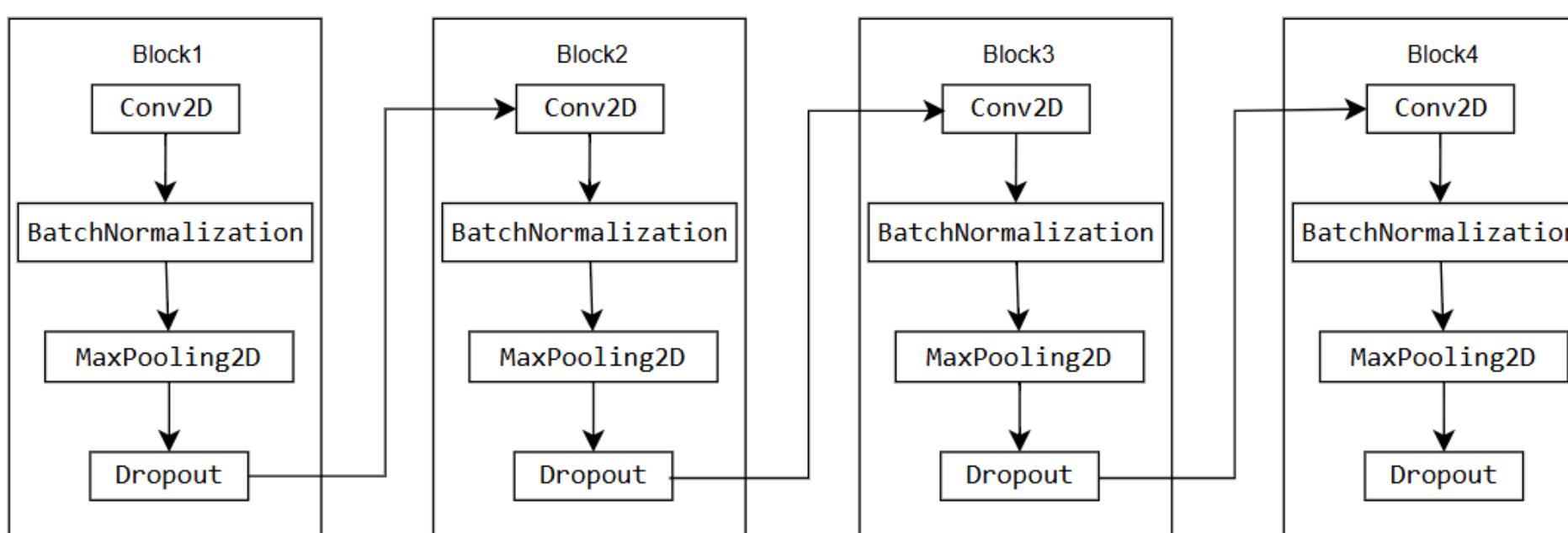
# II. Nhận dạng hình dạng đơn giản

## 2. Giải thích thuật toán

- **CNN (Convolutional Neural Network):**

Mạng nơ-ron tích chập để xử lý ảnh

- Conv2D (trích xuất đặc trưng)
- BatchNormalization (biến mọi giá trị về  $\sim N(0,1)$  với mỗi batch)
- MaxPooling (giảm kích thước)
- Dropout (tránh overfitting)
- Dense (phân loại).



- **Optimizer:**

- Adam (tự điều chỉnh tốc độ học riêng biệt cho từng trọng số theo thời gian).
- Loss: Categorical Crossentropy (Hàm mất mát cho bài toán phân loại nhiều lớp).
- Metrics: Accuracy.

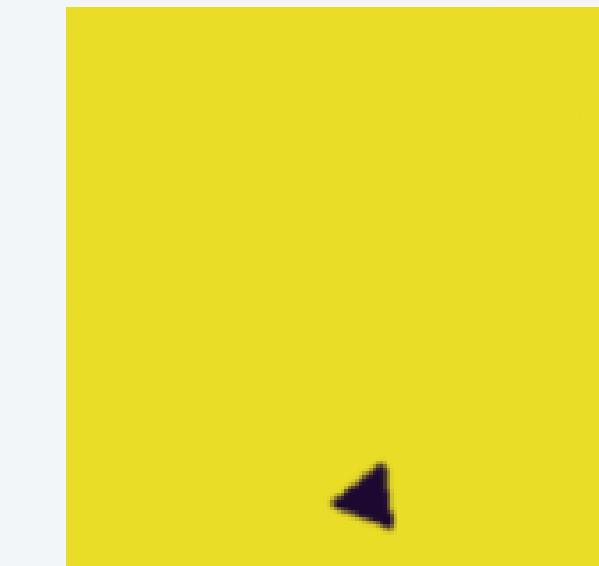
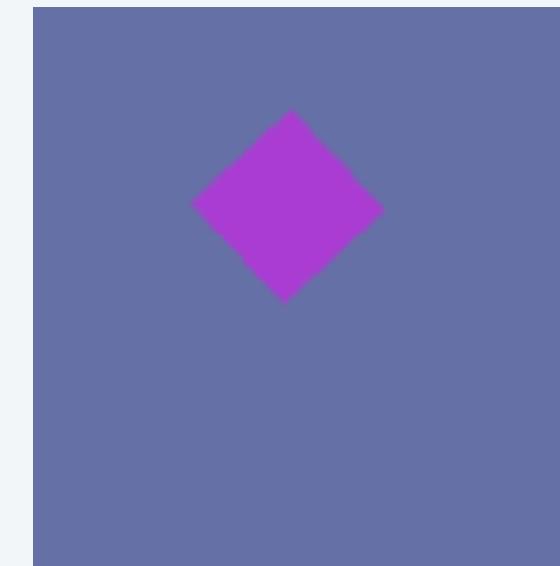
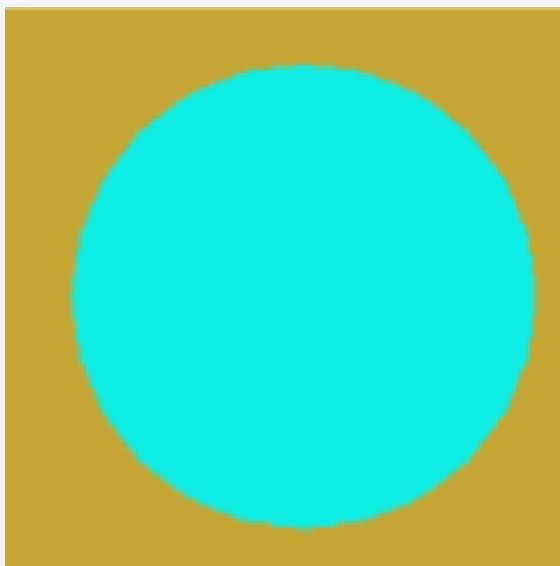
- **Huấn luyện:**

- EarlyStopping (ngừng huấn luyện khi mô hình ngừng cải thiện).
- ReduceLROnPlateau (giảm learning rate khi mô hình ngưng cải thiện).

## II. Nhận dạng hình dạng đơn giản

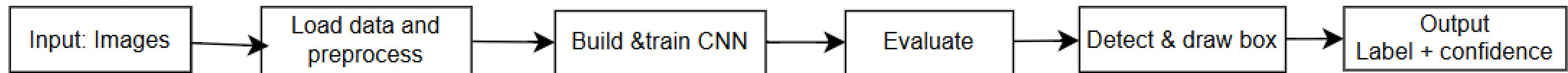
### 3. Mô tả dữ liệu

- **Nguồn:** Dataset từ Kaggle ([/kaggle/input/geometric-shapes-dataset/](#)),
  - 3 lớp: Circle, Square, Triangle. Mỗi lớp có 10,000 ảnh).
  - Định dạng: PNG
- **Phân chia:** Train (64%), Val (16%), Test (20%).
- **Đặc điểm:** Ảnh đơn giản, hình dạng rõ nét
- Có phần demo với ảnh tự tạo (có thể là vật thật với nhiều hình dạng).



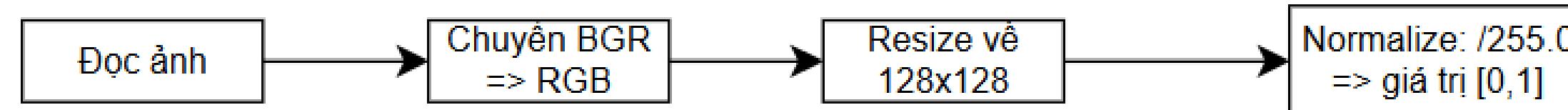
# II. Nhận dạng hình dạng đơn giản

## 4. Luồng hoạt động

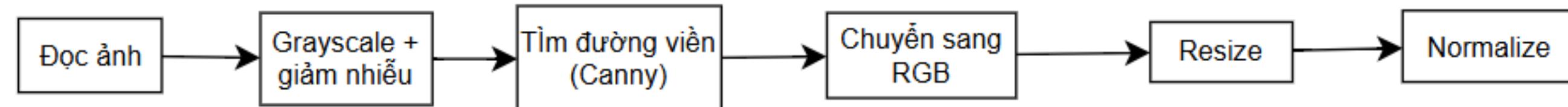


### Tiền xử lý

#### Ảnh vẽ



#### Ảnh thực tế



## **II. Nhận dạng hình dạng đơn giản**

### **5. Demo và kết quả**



# THANK YOU!

Nhóm 7 - D22CNPM02