

**TRƯỜNG ĐẠI HỌC HỌC VĂN LANG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO MÔN HỌC**  
**PHÂN TÍCH DỮ LIỆU VÀ HỌC SÂU**  
NGÀNH: CÔNG NGHỆ THÔNG TIN

***Tên Đồ Án:***

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

**SVTH: Trần Minh Phúc - 2274802010694**

**Châu Gia Kiệt – 2274802010449**

**Võ Đình Ngọc Bình - 2274802010066**

**Nguyễn Hoàng Khanh - 2274802010393**

**GVHD: Trần Kim Mỹ Vân**

**LHP: 242\_71ITDS30203\_0103**

**TP. Hồ Chí Minh – 03/2025**

# MỤC LỤC

I. GIỚI THIỆU .....	3
1. Bối Cảnh Hiện Tại .....	3
2. Lý Do Chọn Đề Tài.....	3
3. Mục Tiêu Đề Tài .....	4
II. TỔNG QUAN LÝ THUYẾT .....	4
1. CNN Là Gì? .....	4
2. Kiến Trúc CNN Cơ Bản.....	5
3. Dữ Liệu Hình Ảnh .....	5
III. MÔ HÌNH NHẬN DIỆN .....	6
1. Dữ Liệu .....	6
2. Tiền Xử Lý Dữ Liệu.....	7
3. Kiến Trúc Mô Hình CNN .....	7
IV. KẾT QUẢ & ĐÁNH GIÁ:.....	9
1. Kết Quả Huấn Luyện .....	9
2. Đánh Giá .....	15
V. KẾT LUẬN.....	16
VI. DANH SÁCH PHÂN CÔNG.....	16
TÀI LIỆU THAM KHẢO.....	16

# I. GIỚI THIỆU

## 1. Bối Cảnh Hiện Tại

Trong bối cảnh cuộc Cách mạng Công nghiệp 4.0, trí tuệ nhân tạo (AI) và học sâu (Deep Learning) đã trở thành những công nghệ chủ chốt, được ứng dụng ngày càng rộng rãi trong nhiều lĩnh vực. Một ví dụ tiêu biểu minh họa cho xu hướng này là bài toán **nhận dạng chữ viết tay**, lĩnh vực đã chứng kiến những tiến bộ vượt bậc nhờ việc áp dụng các phương pháp học sâu hiện đại, đặc biệt là các mô hình mạng nơ-ron tích chập (CNN). Bài toán nhận dạng chữ viết tay có tầm quan trọng đặc biệt và đóng vai trò then chốt trong nhiều ứng dụng thực tiễn, bao gồm các hệ thống giao diện người–máy, kiểm tra và hiệu chỉnh dữ liệu số hóa, cũng như phân loại tự động thông tin. Sự quan tâm ngày càng tăng đối với nhận dạng chữ viết tay trong cả cộng đồng học thuật lẫn công nghiệp đã khẳng định tầm vóc của lĩnh vực này trong kỷ nguyên số hiện nay.

## 2. Lý Do Chọn Đề Tài

Tuy nhiên, trong quá trình khảo sát các hệ thống nhận dạng chữ số hiện có, nhóm chúng em nhận thấy rằng việc sử dụng các phương pháp nhận dạng số thông qua hình ảnh tĩnh (ví dụ: ảnh chụp hoặc ảnh scan chứa chữ số) vẫn tồn tại một số hạn chế nhất định. Cụ thể, người dùng cần phải tìm kiếm và cung cấp chính xác các hình ảnh chứa chữ số cần nhận dạng, điều này gây ra sự bất tiện và làm gián đoạn trải nghiệm tương tác, đặc biệt trong các ứng dụng cần thao tác nhanh hoặc mang tính thời gian thực.

Xuất phát từ thực tiễn đó, nhóm chúng em đề xuất một hướng tiếp cận trực quan hơn: xây dựng hệ thống nhận dạng chữ số viết tay theo thời gian thực, cho phép người dùng viết trực tiếp chữ số lên giao diện và hệ thống sẽ tự động nhận dạng, xử lý và hiển thị kết quả tương ứng. Cách tiếp cận này không chỉ đơn giản hóa quy trình tương tác

mà còn phù hợp với xu hướng thiết kế các hệ thống giao tiếp người–máy thân thiện, thông minh và hiệu quả hơn.

### **3. Mục Tiêu Đề Tài**

Nhằm xây dựng một mô hình nhận dạng chữ số viết tay cho phép người dùng viết trực tiếp một chữ số lên giao diện và hệ thống có thể tự động nhận dạng, xử lý và hiển thị kết quả chính xác trên màn hình. Mô hình được thiết kế với mục tiêu đạt tỷ lệ nhận dạng chính xác tối thiểu 50% trong các điều kiện sử dụng cơ bản, làm tiền đề cho việc cải tiến và mở rộng hiệu suất trong các giai đoạn tiếp theo.

Bên cạnh việc kiểm chứng hiệu quả của mô hình trong môi trường thử nghiệm, đề tài cũng hướng đến khả năng ứng dụng trong các tình huống thực tiễn, chẳng hạn như trong giáo dục, khảo sát số liệu, hay giao diện tương tác người–máy thân thiện. Thông qua đề tài này, nhóm em mong muốn đóng góp một giải pháp đơn giản, khả thi và có tiềm năng mở rộng trong lĩnh vực nhận dạng hình ảnh ứng dụng học sâu.

## **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

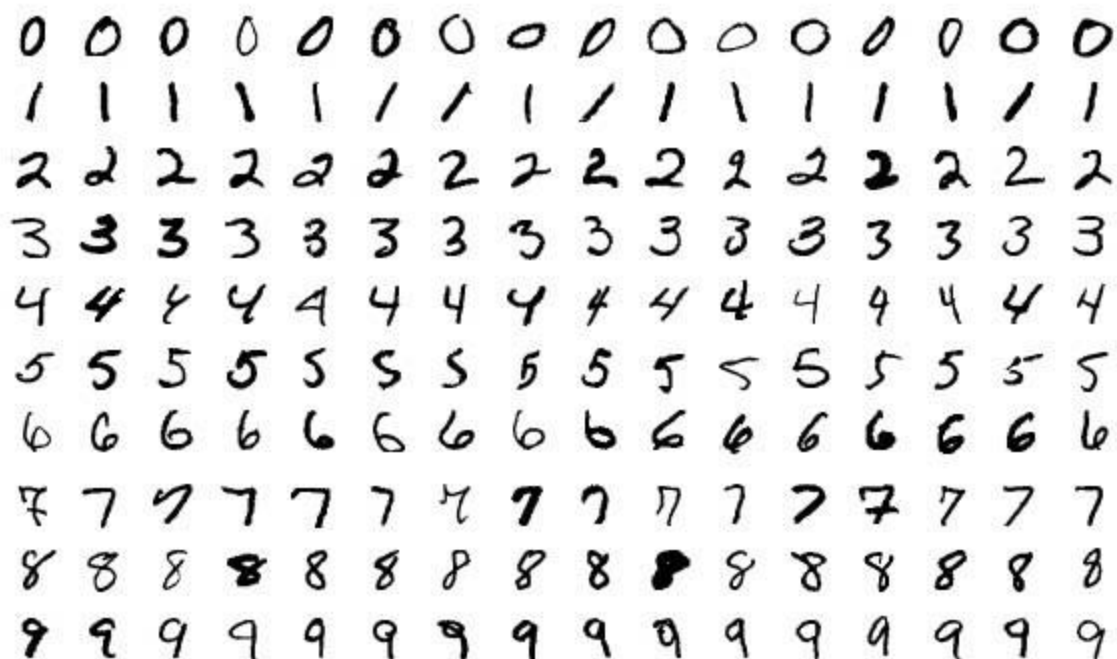
Một mô hình CNN điển hình bao gồm các thành phần chính sau:

- **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:**



**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`).

```
# Tải dữ liệu MNIST gồm ảnh chữ số viết tay 0-9
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
```

## 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].

```
# Reshape dữ liệu ảnh từ (28, 28) → (28, 28, 1) để thêm kênh màu (grayscale)
# 1 kênh màu (grayscale)
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
input_shape = (28, 28, 1)
```

## 3. Kiến Trúc Mô Hình CNN

Trong phạm vi đề tài, nhóm triển khai mô hình nhận dạng chữ số viết tay dựa trên kiến trúc mạng nơ-ron tích chập (Convolutional Neural Network – CNN), vốn đã chứng minh hiệu quả cao trong các bài toán xử lý ảnh. Mô hình đề xuất được thiết kế với kiến trúc đơn giản nhưng hiệu quả, bao gồm các lớp chính như sau:

- **Conv2D:** với 28 bộ lọc có kích thước nhân  $3 \times 3$  và hàm kích hoạt ReLU, dùng để trích xuất đặc trưng cục bộ từ ảnh đầu vào.

```
# Lớp tích chập: 28 kernel (3x3), trích xuất đặc trưng ảnh
model.add(layers.Conv2D(28, kernel_size=(3, 3), input_shape=input_shape))
```

- **MaxPooling2D:** với kích thước cửa sổ  $2 \times 2$  nhằm giảm chiều dữ liệu và làm nổi bật các đặc trưng quan trọng.

```
# Lớp pooling: giảm kích thước ảnh để giảm tính toán
model.add(layers.MaxPooling2D(pool_size=(2, 2)))
```

- **Flatten:** chuyển đầu ra 2D từ các lớp trước về dạng vector 1 chiều để đưa vào các lớp fully-connected.

```
# Chuyển tensor 3D thành vector 1 chiều để đưa vào fully-connected
# các lớp Dense chỉ nhận đầu vào 1 chiều, nên cần flatten trước khi đưa vào.
model.add(layers.Flatten())
```

- **Dense:** với 128 nút ẩn và hàm kích hoạt ReLU, cho phép học các biểu diễn phi tuyến phức tạp.

```
# Lớp Dense ẩn với 128 neuron, dùng hàm kích hoạt ReLU
# dùng hàm kích hoạt ReLU để giới thiệu phi tuyến, giúp mô hình học tốt hơn.
model.add(layers.Dense(128, activation=tf.nn.relu))
```

- **Dropout:** với tỷ lệ 0.2 để giảm hiện tượng overfitting.

```
# Dropout: giảm overfitting bằng cách bỏ ngẫu nhiên 20% neuron
# Điều này giúp mô hình không quá phụ thuộc vào một số neuron cụ thể → tăng khả năng khái quát khi gặp dữ liệu mới.
model.add(layers.Dropout(0.2))
```

- **Dense:** đầu ra với 10 nút (tương ứng 10 lớp chữ số từ 0 đến 9) và hàm kích hoạt softmax, cho phép phân loại đầu ra theo xác suất.

```
# activation = tf.nn.softmax: biến đầu ra thành xác suất
model.add(layers.Dense(10, activation=tf.nn.softmax))
```

Mô hình được biên dịch với các tham số như sau:

- **Loss function:** `sparse_categorical_crossentropy`, phù hợp với bài toán phân loại đa lớp có nhãn dạng số nguyên.
- **Optimizer:** Adam, một thuật toán tối ưu hóa hiệu quả và phổ biến trong huấn luyện mạng nơ-ron.
- **Metrics:** accuracy, nhằm theo dõi hiệu suất phân loại trong quá trình huấn luyện và đánh giá.

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```



Quá trình huấn luyện được thực hiện trong **20 epoch** trên toàn bộ tập dữ liệu huấn luyện, với mục tiêu đạt được độ chính xác cao nhất có thể trên cả tập huấn luyện và tập kiểm tra. Kiến trúc này không chỉ đảm bảo tính đơn giản và dễ triển khai, mà còn tạo nền tảng để mở rộng hoặc tối ưu hóa thêm trong các nghiên cứu tiếp theo.

```
# Huấn luyện mô hình với 20 epoch (lặp 20 lần trên toàn bộ dữ liệu)
model.fit(x=x_train, y=y_train, epochs=20)
```

## IV. KẾT QUẢ & ĐÁNH GIÁ:

### 1. Kết Quả Huấn Luyện

Sau quá trình huấn luyện và thử nghiệm, mô hình đã đạt được kết quả vượt kỳ vọng với độ chính xác trên tập kiểm tra đạt **trên 98%**, cho thấy khả năng học và phân loại vượt trội của kiến trúc CNN được đề xuất. Kết quả này phản ánh rằng mô hình đã nắm bắt tốt các đặc trưng hình học đặc trưng của chữ số viết tay, ngay cả khi có sự khác biệt nhất định về kiểu chữ, nét viết hoặc độ đậm nhạt của từng mẫu.

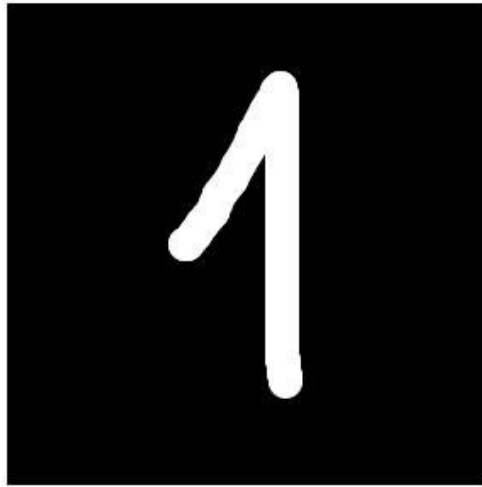
```
# evaluate mô hình trên tập test để kiểm tra độ chính xác
test_loss, test_acc = model.evaluate(x_test, y_test)
print('Độ chính xác trên tập test: %.2f%%' % (test_acc * 100))
```

- Kết quả trả về:

```
Độ chính xác trên tập test: 98.39%
```

- Kết quả những lần train:

## Nhận dạng chữ số



Xóa

Dự đoán

**Kết quả: 1**

## Nhận dạng chữ số



Xóa

Dự đoán

**Kết quả: 2**

### Nhận dạng chữ số



Xóa

Dự đoán

**Kết quả: 3**

### Nhận dạng chữ số

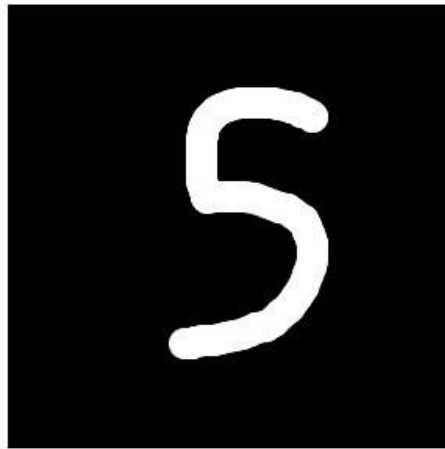


Xóa

Dự đoán

**Kết quả: 4**

## Nhận dạng chữ số



Xóa

Dự đoán

**Kết quả: 5**

## Nhận dạng chữ số

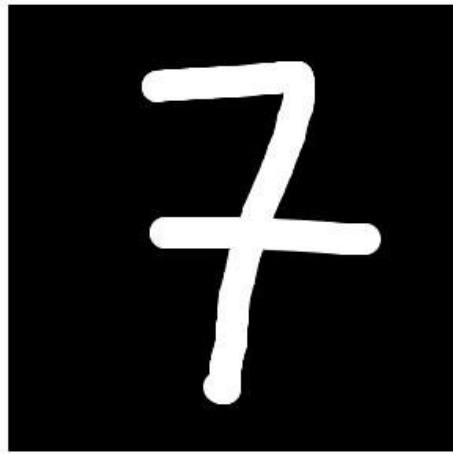


Xóa

Dự đoán

**Kết quả: 6**

## Nhận dạng chữ số

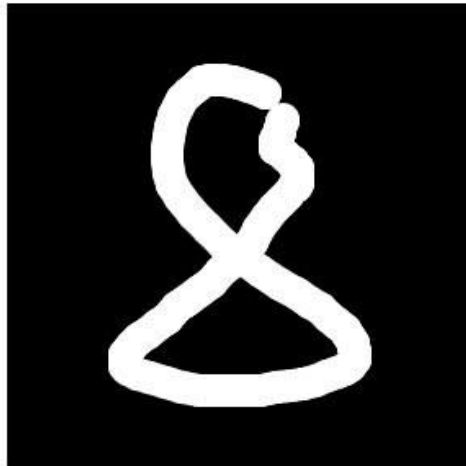


Xóa

Dự đoán

**Kết quả: 7**

## Nhận dạng chữ số

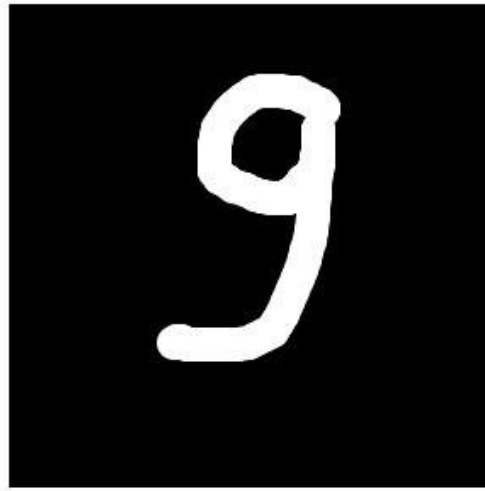


Xóa

Dự đoán

**Kết quả: 8**

## Nhận dạng chữ số



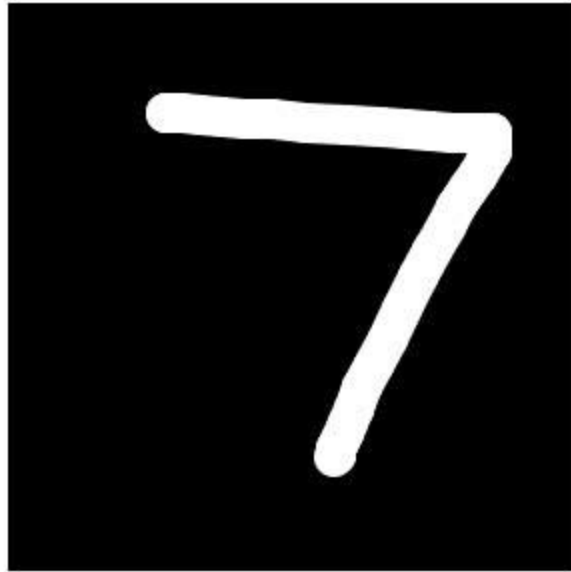
Xóa

Dự đoán

**Kết quả: 9**

Tuy nhiên, bên cạnh những kết quả tích cực, mô hình vẫn còn tồn tại một số hạn chế. Một số mẫu chữ số có nét viết bất thường hoặc không rõ ràng dẫn đến nhầm lẫn trong quá trình phân loại. Điều này cho thấy mô hình chưa đạt hiệu quả tối đa và vẫn cần được cải thiện thêm, thông qua các hướng như tăng kích thước và đa dạng dữ liệu huấn luyện, tinh chỉnh siêu tham số hoặc áp dụng các kỹ thuật tăng cường dữ liệu (data augmentation). Nhóm cũng ghi nhận rằng việc phân tích sâu hơn các trường hợp phân loại sai sẽ là cơ sở quan trọng để tối ưu mô hình trong các nghiên cứu tiếp theo.

## Nhận dạng chữ số



Xóa

Dự đoán

**Kết quả: 2**

## 2. Đánh Giá

Đánh giá của nhóm em sau đã chạy chương trình:

- Mô hình có tốc độ hội tụ nhanh, ổn định qua từng epoch.
- Biểu đồ loss giảm dần, accuracy tăng đều, không dao động mạnh.
- Sau 20 epochs, mô hình đạt độ chính xác >98% trên tập kiểm tra.
- Thời gian huấn luyện ngắn, phù hợp với các mô hình đơn giản hoặc ứng dụng thời gian thực.
- Tuy nhiên, mô hình vẫn còn nhầm lẫn với một số chữ số viết không rõ, cần cải thiện thêm để tăng độ chính xác và tính ổn định.

## V. KẾT LUẬN

Thông qua quá trình nghiên cứu và triển khai, nhóm đã xây dựng thành công một mô hình CNN đơn giản nhưng hiệu quả cho bài toán nhận diện chữ số viết tay trên tập dữ liệu MNIST. Mô hình cho thấy khả năng huấn luyện nhanh, độ chính xác cao và hiệu suất ổn định, phù hợp cho các ứng dụng thực tiễn trong nhận dạng văn bản số.

Đề tài không chỉ mang lại kết quả kỹ thuật khả quan mà còn giúp nhóm sinh viên nắm vững quy trình phát triển một mô hình học sâu, từ tiền xử lý dữ liệu, thiết kế kiến trúc mạng, huấn luyện, đến đánh giá mô hình.

Trong tương lai, hướng phát triển tiếp theo có thể là mở rộng mô hình cho các bài toán phức tạp hơn như **nhận dạng chữ viết tay tiếng Việt**, **ký tự tự do**, hoặc **biển số xe**, góp phần giải quyết các bài toán thực tế trong lĩnh vực thị giác máy tính.

## VI. DANH SÁCH PHÂN CÔNG

Tên Thành Viên	Nhiệm Vụ
Trần Minh Phúc	Viết code
Châu Gia Kiệt	Viết code
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](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u_MNIST)