TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

TÊN MÔN HỌC: MÁY HỌC

BÁO CÁO ĐỒ ÁN MÔN HỌC

**ĐỀ TÀI: NHẬN DIỆN CHỮ SỐ VIẾT TAY**

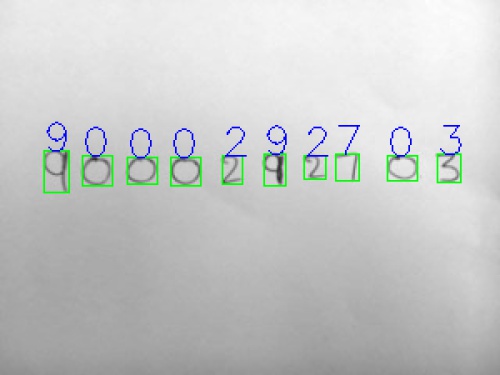
THÀNH VIÊN THỰC HIỆN:

NGUYỄN CÔNG HẬU

HUỲNH VĂN HUÂN

TRẦN HÀ NY

ĐINH THÀNH THUẬN



MỤC LỤC

[PHẦN I: CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG 3](#_Toc533482373)

[1.1. OpenCV: 3](#_Toc533482374)

[1.2. TensorFlow: 3](#_Toc533482375)

[1.3. CNN-Keras: 3](#_Toc533482376)

[1.4. Random Forest Classifier: 4](#_Toc533482377)

[1.5. MNIST dataset: 4](#_Toc533482378)

[PHẦN 2: TẬP DỮ LIỆU TRAIN & CÔNG NGHỆ TRAIN DỮ LIỆU 5](#_Toc533482379)

[PHẦN 3:XỬ LÝ DỮ LIỆU & NGUYÊN TẮC HOẠT ĐỘNG 7](#_Toc533482380)

[3.1. CNN-Keras 7](#_Toc533482381)

[3.2. RandomForestClassification 8](#_Toc533482382)

[3.3. kNN 8](#_Toc533482383)

[PHẦN 4: DEMO + TÀI LIỆU THAM KHẢO 9](#_Toc533482384)

# PHẦN I: CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG

* 1. OpenCV:

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và [máy học](https://techmaster.vn/khoa-hoc/25511/machine-learning-co-ban), và các tính năng tăng tốc GPU trong hoạt động thời gian thực. OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực.

* 1. TensorFlow:

Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lí tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu. Tensorflow cung cấp một loại dữ liệu mới được gọi là Tensor. Trong thế giới của Tensorflow, mọi kiểu dữ liệu đều được quy về một mối được gọi là Tensor hay trong Tensorflow, tất cả các loại dữ liệu đều là Tensor. Vậy nên có thể hiểu được phần nào cái tên Tensorflow là một thư viện mô tả, điều chỉnh dòng chảy của các Tensor.

* 1. CNN-Keras:

Keras là một thư viện được phát triển bởi François Chollet, là một kỹ sư nghiên cứu deep learning tại google. Là một hệ thống mã nguồn mở cho Neural Network được xây dựng trên nền tảng python. Keras là một API bậc cao có thể sử dụng chung với các thư viện deep learning nổi tiếng như tensorflow.

CNN-Keras:

Convolutional Neural Network (Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning được tạo ra để xây dựng được những hệ thống nhận diện với độ chính xác cao như những hệ thống xử lý ảnh (khuôn mặt, biển số, chữ viết, số viết tay,… )

Các Layer trong Keras hổ trợ sử dụng CNN:

**Pooling-Layer** trong Sub-model Layers của Keras, Layer này chứa các layer dùng trong mạng CNN:

* MaxPooling1D, MaxPooling2D dùng để lấy feature nổi bật(dùng max) và giúp giảm parameter khi training
* AveragePooling1D, AveragePooling2D
* GlobalMaxPooling1D, GlobalMaxPooling2D

**Convolutional Layers**: chứa các layer trong mạng nơ ron tích chập hai

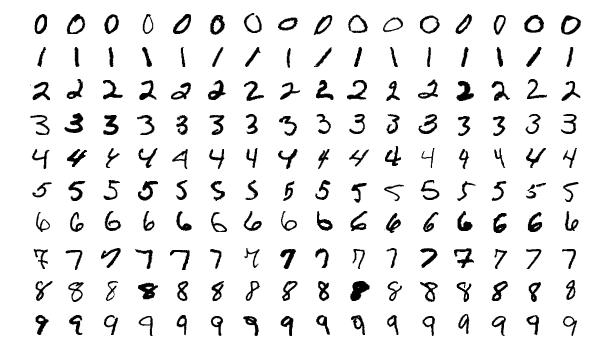
* Conv1D,Conv2D là convolution layer dùng để lấy feature từ image. tham số cần chú ý:
  + filters số filter của convolution layer
  + kernel\_size size window search trên image
  + strides bước nhảy mỗi window search
  + padding same là dùng padding,valid là ko
  + data\_format format channel ở đầu hay cuối
* UpSampling1D,UpSampling2D Ngược lại với convolution layer
  + size (size(2,2)) có nghĩa mỗi pixel ban đầu sẽ thành 4 pixel
* ZeroPadding1D,ZeroPadding2D dùng để padding trên image.
  + padding số pixel padding
  1. Random Forest Classifier:

Dựa trên ý tưởng thuật toán cây quyết định (Decision-Tree), đây là một thuật toán Machine Learning cho phép cải thiện độ chính xác khi phân loại hình ảnh.

Thuật toán được sử dụng trên nền tảng Scikit-Learn.

* 1. MNIST dataset:

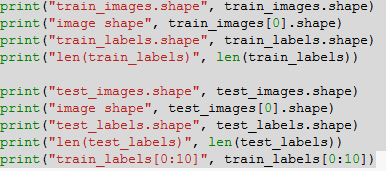
MNIST là một tập dữ liệu chữ số viết tay từ 0 đến 9 bao gồm 60000 mẫu cho tập huấn luyện (training set) và 10000 mẫu cho tập kiểm tra (test set). MNIST được giới thiệu năm 1998 bởi Yann Lecun và cộng sự nhằm đánh giá các mô hình phân lớp.



Nguồn: Wikipedia

# PHẦN 2: TẬP DỮ LIỆU TRAIN & CÔNG NGHỆ TRAIN DỮ LIỆU

Sử dụng API được cung cấp sẵn trong Keras để truy cập vào dữ liệu của MNIST một cách tự động.



Trong đó:

 **train\_images** : là một ma trận có kích thước (60000, 28 , 28). Đây là ma trận chứa 60000 hình training với mỗi hình có kích thước là 28 x 28 pixel.

 **train\_labels**: là một vector có kích thước 60000 dữ liệu. Đây là nhãn của từng hình tương ứng trong tập train.

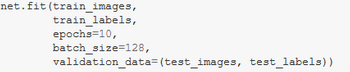
 **test\_images** và **test\_labels** :10000 hình test với mỗi hình là 28 x 28 pixel.

Tiếp theo, sử dụng mạng fully-connected để thực hiện nhận diện chữ số viết tay. Mạng fully-connected sẽ có đầu vào là 784 nơ ron tương ứng với 784 pixels ( 28 x 28 pixels) và đầu ra là 10 giá trị ứng với số từ 0 đến 9.

Sau đó tiếp tục xử lý dữ liệu từ đầu và là 784 neuron bằng cách vector hóa hình thành một vector bằng hàm **reshape** trong Numpy. Bên cạnh đó, chúng ta cũng sẽ chuyển dữ liệu về kiểu dữ liệu là **float32** và chuyển về miền giá trị của từng phần tử là [0, 1].

Dữ liệu nhãn (label) của chúng ta hiện tại là một giá trị số nguyên, từ 0 đến 9. Trong khi đó, đầu ra của mô hình là 10 nơ ron. Cần phải chuyển nhãn về dạng One hot vector. One hot vector là một vector với các giá trị đều bằng 0, ngoại trừ một giá trị bằng 1 tại vị trí có index bằng với giá trị của nhãn.

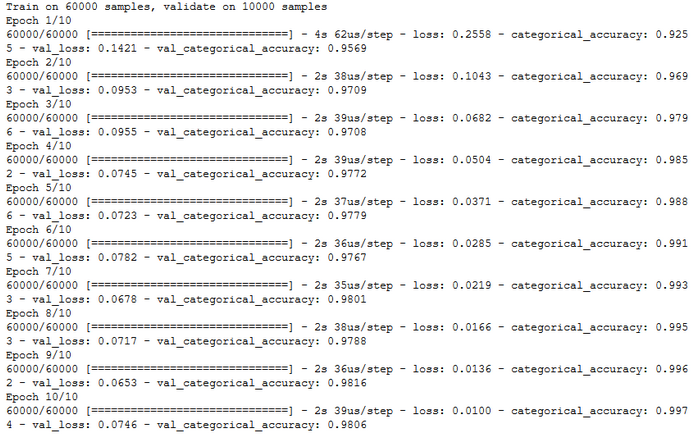
Huấn luyện mô hình: hàm **net.fit** là sẽ tính hành huấn luyện mô hình như sau.



Trong đó:

 **Epochs** là một lần quá trình huấn luyện học qua tất cả các dữ liệu trong tập huấn luyện.

 **batch\_size:** bộ nhớ của máy tính / card đồ họa, số lượng dữ liệu một lần tính gọi là **batch\_size**.



Kết quả thực hiện hàm net.fit

Trong đó:

* **Epoch 10/10:** Đây là epoch thứ 10
* **loss: 0.01:** Đây là độ lỗi trên tập train trong epoch này
* **categorical\_accuracy: 0.9975**: Đây là độ chính xác trên tập train. Đây có nghĩa là mạng đã đạt được 99.75% trên tập train.
* **val\_loss: 0.0746** : Đây là độ lỗi trên tập test trong epoch này.
* **val\_categorical\_accuracy: 0.9806:** Đây là độ chính xác trên tập test. Đây có nghĩa là mạng đã đạt được 98.06% trên tập test.

# PHẦN 3:XỬ LÝ DỮ LIỆU & NGUYÊN TẮC HOẠT ĐỘNG

* 1. CNN-Keras



Thuật toán CNN nhận một ảnh input đầu vào;

* Bước 1: Tạo Convolution layer thứ nhất với “Activation function: relu”

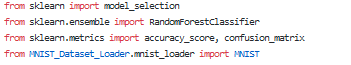
Tiếp tục thực hiện lọc Pooling với pool-size:2x2

Kết thúc ta sẽ có được lớp Convolution đầu tiên

* Bước 2: Lặp lại bước 1 để có lớp Convolution thứ hai
* Bước 3: Tiếp tục thực hiện để có lớp Convolution thứ 3
* Bước 4: Tạo lớp Fully-Connected
* Bước 5: Dùng Activation function: softmax để phân lớp dữ liệu
* Bước 6: Kết xuất kết quả
  1. RandomForestClassification

Random Forest áp dụng phối hợp nhiều thuật toán phân loại khác nhau (như là kNN, Logistic Regression, SVM, Decision Tree) để đạt được kết quả phân loại tốt nhất.

Thư viện được sử dụng: Scikit-Learn là một thư viện mã nguồn mở dành cho học máy, chứa hầu hết các thuật toán machine learning hiện đại nhất.



Sử dụng Dataset MNIST để thực hiện phân loại các bức ảnh.

C:\Users\Thuan Gay\Desktop\BaoCaoMayHoc\43.PNG

Chương trình sử dụng thuật toán RandomForestClassification với độ lớn của cây quyết định n-estimators=100.

* 1. kNN

K-NN là phương pháp để phân lớp các đối tượng dựa vào khoảng cách gần nhất giữa đối tượng cần xếp lớp (Query point) và tất cả các đối tượng trong Training Data. Một đối tượng được phân lớp dựa vào K láng giềng của nó. K là số nguyên dương được xác định trước khi thực hiện thuật toán.

Thư viện Scikit-Learn cung cấp công cụ sử dụng thuật toán kNN, sử dụng dataset MNIST làm dữ liệu cho thuật toán.

C:\Users\Thuan Gay\Desktop\BaoCaoMayHoc\45.PNG

C:\Users\Thuan Gay\Desktop\BaoCaoMayHoc\46.PNG

Sử dụng thuật toán kNN với số láng giềng gần nhất n-neighbors=5.

# PHẦN 4: DEMO + TÀI LIỆU THAM KHẢO

Link youtube:

<https://www.youtube.com/watch?v=LCUlFCX58KM>

Link download:

<https://bitbucket.org/MachineGunHaunter/machinelearning_digitrecognition/src>

or

<https://drive.google.com/open?id=1yO2epmSASRWw7uVbBR91rs5Cnd-kT0yH>

Tài liệu tham khảo:

<https://github.com/anujdutt9/Handwritten-Digit-Recognition-using-Deep-Learning>

<https://opencv.org/>

<https://conda.io/docs/user-guide/install/download.html>

<https://www.google.com/>

<https://www.youtube.com/>