



## NHẬP MÔN CÔNG NGHỆ THÔNG TIN

### ĐỒ ÁN MÔN HỌC – ĐỀ 2

#### DAMH-02: NHẬN DẠNG CHỮ SỐ VIẾT TAY

##### I. Thông tin chung

Mã số bài tập:	DAMH-02-TGMT
Thời lượng dự kiến:	5 tuần
Deadline nộp bài:	TBA
Hình thức:	Đồ án theo nhóm
Hình thức nộp bài:	
GV phụ trách:	Phạm Minh Hoàng
Thông tin liên lạc với GV:	<a href="mailto:pmhoang@fit.hcmus.edu.vn">pmhoang@fit.hcmus.edu.vn</a>

##### II. Chuẩn đầu ra cần đạt

Đồ án này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- G1.2: Tham gia thảo luận, tranh luận theo nhóm trên chủ đề môn học.
- G1.3: Phân tích, tổng hợp và viết tài liệu kỹ thuật theo mẫu cho trước theo cá nhân hoặc cộng tác nhóm.
- G2.1: Biết, hiểu thuật ngữ tiếng Anh thuộc các ngành và chuyên ngành.
- G7.6: Xây dựng một ứng dụng đơn giản, ứng dụng kiến thức ngành Thị giác máy tính.

##### III. Mô tả đồ án

###### 1. Nội dung chính

Xây dựng một ứng dụng nhận diện chữ số viết tay đơn giản.

###### 2. Mục tiêu đồ án

Sinh viên thực hiện đồ án sẽ nắm được các kiến thức

- Các bước của bài toán nhận dạng ảnh (TGMT).

- Các bước xây dựng phần mềm (CNPM).
- Một số công cụ hỗ trợ làm việc trên môi trường Internet (MMT).

Sinh viên thực hiện đồ án sẽ rèn luyện các kĩ năng

- Lập kế hoạch, làm việc nhóm, tổ chức và quản lí nhóm
- Tìm hiểu và viết báo cáo
- Đọc hiểu tiếng Anh và các tài liệu chuyên ngành

### 3. Nội dung chi tiết của đồ án

Sinh viên được cung cấp đoạn code xử lý dữ liệu đơn giản. Hãy thực hiện những yêu cầu sau:

1. Cài đặt thành công môi trường lập trình: ngôn ngữ Python, công cụ lập trình Anaconda, thư viện máy học và xử lý ảnh Scikit-learn.
2. Chuẩn bị dữ liệu bao gồm: tập huấn luyện, tập kiểm thử, load và show dữ liệu
3. Cài đặt một số thuật toán rút trích đặc trưng đơn giản: vector hóa ảnh, down-sampling, histogram
4. Cài đặt các bộ phân lớp: KNN, mẫu trung bình
5. Chạy thử nghiệm trên tập dữ liệu, lập báo cáo kết quả

### IV. Các yêu cầu & quy định chi tiết cho đồ án

#### ❖ Các kết quả cần đạt được

- **Báo cáo tìm hiểu.**
- **Project plan.** Kế hoạch thực hiện.
- **Sản phẩm**

### V. Cách đánh giá

Việc đánh giá đồ án của sinh viên được dựa trên các phần như sau:

- Bản kế hoạch thực hiện đồ án
- Báo cáo quy trình tiến độ thực hiện và các kết quả đạt được từng tuần (theo kế hoạch trên).
- Báo cáo quy trình thực hiện và các kết quả đạt được cuối cùng.
- Demo minh họa.

### VI. Tài liệu tham khảo

- Mastering Machine Learning with scikit-learn
- Scikit-learn Cookbook
- <http://yann.lecun.com/exdb/mnist/>

## VII. Yêu cầu mỗi tuần

- Tuần 1: Cài đặt thành công môi trường lập trình, thư viện, chuẩn bị dữ liệu (load và show được dữ liệu)
- Tuần 2: Cài đặt ít nhất rút trích đặc trưng bằng vector hóa ảnh
- Tuần 3: Cài đặt ít nhất rút trích đặc trưng bằng downsampling, histogram
- Tuần 4: Cài đặt bộ phân lớp kNN, mẫu trung bình
- Tuần 5: Chạy thử nghiệm trên tập kiểm thử, lập báo cáo kết quả

## VIII. Hướng dẫn thực hành

Chuẩn bị dữ liệu. Download tập dữ liệu MNIST tại <http://yann.lecun.com/exdb/mnist/>.

Lưu các file đã download trong thư mục data cùng đường dẫn file mã nguồn python.

Copy đoạn code load và show ảnh và chạy thử

```
import matplotlib.pyplot as plt
import os
import numpy as np
import gzip

def load_mnist(path, kind='train'):
    """Load MNIST data from `path`"""
    labels_path = os.path.join(path, '%s-labels-idx1-ubyte.gz' % kind)
    images_path = os.path.join(path, '%s-images-idx3-ubyte.gz' % kind)

    with gzip.open(labels_path, 'rb') as lbpath:
        lbpath.read(8)
        buffer = lbpath.read()
        labels = np.frombuffer(buffer, dtype=np.uint8)
    with gzip.open(images_path, 'rb') as imgpath:
        imgpath.read(16)
        buffer = imgpath.read()
        images = np.frombuffer(buffer, dtype=np.uint8).reshape(len(labels), 28, 28).astype(np.float64)

    return images, labels

X_train, y_train = load_mnist('data/', kind='train')
print('Rows: %d, columns: %d' % (X_train.shape[0], X_train.shape[1]))

fig, ax = plt.subplots(nrows=2, ncols=5, sharex=True, sharey=True,)
ax = ax.flatten()
for i in range(10):
    img = X_train[y_train == i][0]
    ax[i].imshow(img, cmap='Greys', interpolation='nearest')

ax[0].set_xticks([])
ax[0].set_yticks([])
plt.tight_layout()
plt.show()
```