

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
ĐẠI HỌC KHOA HỌC TỰ NHIÊN**



**BÁO CÁO ĐỒ ÁN  
HỌC SÂU  
NHẬN DẠNG CHỮ VIẾT TAY**

**Giảng viên:**

TS. Nguyễn Tiến Huy  
TS. Lê Thanh Tùng  
ThS. Nguyễn Trần Duy Minh  
ThS. Phan Thị Phương Uyên

**Thực hiện đề tài:**

22C15018: Phạm Minh Thạch

**TP.HCM, ngày 07 tháng 05 năm 2023**

## MỤC LỤC

<b>GIỚI THIỆU ĐỀ ÁN.....</b>	<b>3</b>
<b>TÌM HIỂU TẬP DỮ LIỆU .....</b>	<b>3</b>
<b>TIỀN XỬ LÝ.....</b>	<b>3</b>
<b>PHƯƠNG PHÁP TRAIN MODEL .....</b>	<b>3</b>
<b>MODEL 1 .....</b>	<b>4</b>
<b>MODEL 2 .....</b>	<b>4</b>
<b>MODEL 3 .....</b>	<b>5</b>
<b>TỔNG KẾT.....</b>	<b>6</b>
<b>NGUỒN THAM KHẢO.....</b>	<b>6</b>

## GIỚI THIỆU ĐỒ ÁN

Ở đồ án này, chúng ta sẽ xây dựng 3 mô hình học sâu để giải quyết bài toán nhận dạng chữ viết tay với thư viện tensorflow và keras.

Các thư mục và file của đồ án được tổ chức như sau:

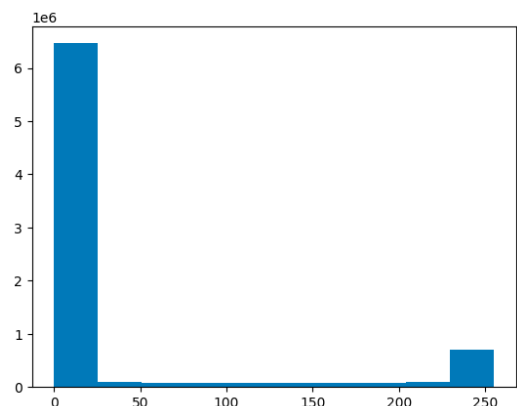
- README.md : Giới thiệu, hướng dẫn cài đặt môi trường
- report.docx : Báo cáo
- report.pdf : Báo cáo
- explore\_data.ipynb : Jupyter notebook tìm hiểu tập dữ liệu
- model\_1.ipynb : Jupyter notebook huấn luyện model 1
- model\_2.ipynb : Jupyter notebook huấn luyện model 2
- model\_3.ipynb : Jupyter notebook huấn luyện model 3

## TÌM HIỂU TẬP DỮ LIỆU

Tập dữ liệu là được chia thành 2 tập: train và test. Tập train gồm 60,000 ảnh trắng đen chữ viết tay từ 0 đến 9. Tập test gồm 10,000 ảnh.

Số lượng mẫu train cho từng nhãn khá đồng đều nhau. Khoảng 10,000 mẫu cho mỗi nhãn.

Cách điểm ảnh có giá trị từ 0 đến 255 và phân bố chủ yếu trong khoảng [0, 25] và [230, 255] như histogram trong **hình 1**. Khoảng [0, 25] là màu sắc của background và [230, 255] là màu sắc của các nét chữ.



Hình 1. Histogram giá trị các điểm ảnh

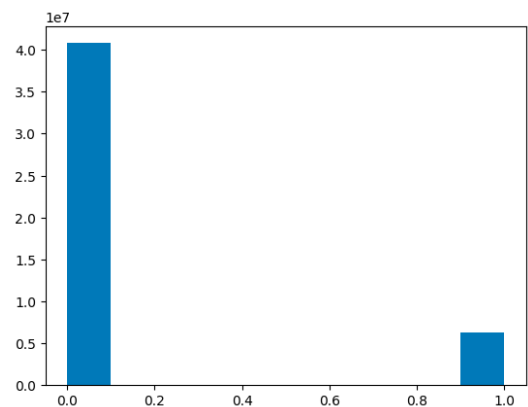
## TIỀN XỬ LÝ

Để thuận lợi cho việc tính toán của các mô hình, ta đưa giá trị của các điểm ảnh về khoảng [0,1] theo công thức sau.

$$q = \begin{cases} 0 & \text{nếu } p < 127 \\ 1 & \text{nếu } p \geq 127 \end{cases}$$

Trong đó, p là giá trị ban đầu của điểm ảnh, q là giá trị mới phục vụ cho việc tính toán sau này.

Sau bước tiền xử lý, ta có histogram giá trị của các điểm ảnh như **hình 2**.



Hình 2. Histogram giá trị các điểm ảnh sau tiền xử lý

## PHƯƠNG PHÁP TRAIN MODEL

Cả 3 model sẽ được train theo cùng một phương pháp với các tham số như sau.

Epoch number	5
Batch size	64
Optimizer	Adam

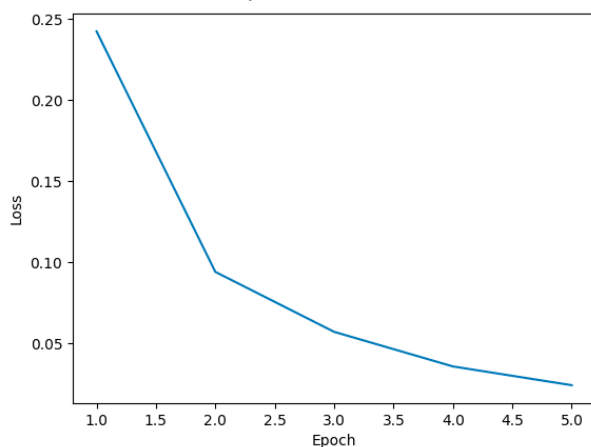
Loss function	Categorical Cross Entropy
Metric	Accuracy

## MODEL 1

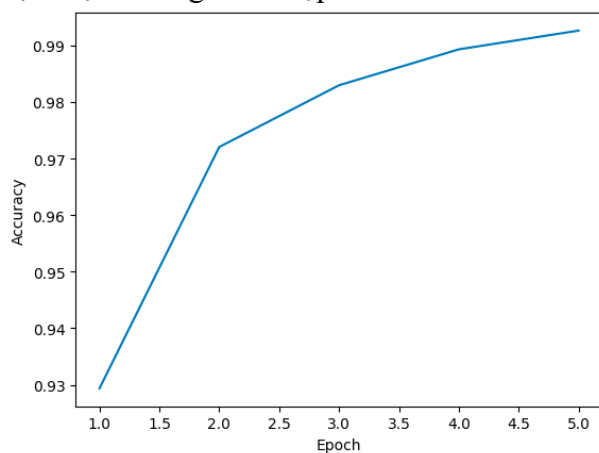
Model 1 chỉ gồm các fully connected layer với cấu trúc như sau.

```
model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(28*28,)))
model.add(Dense(10, activation='softmax'))
```

Độ lỗi và độ chính xác của model trên tập train theo từng epoch được thể hiện như biểu đồ ở **hình 3** và **hình 4**. Độ chính xác của model 1 khi thực hiện đánh giá trên tập test là: 0.9777.



Hình 3. Loss của model 1



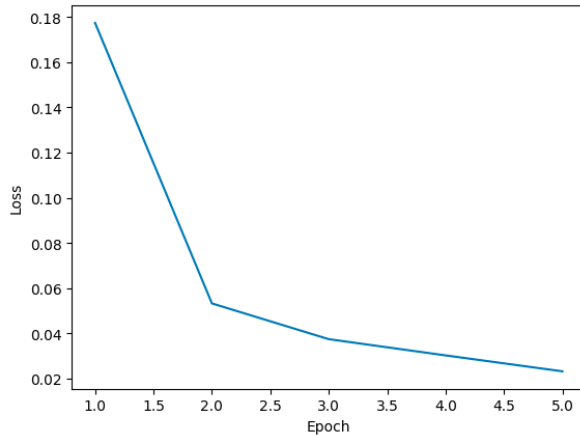
Hình 4. Accuracy của model 1

## MODEL 2

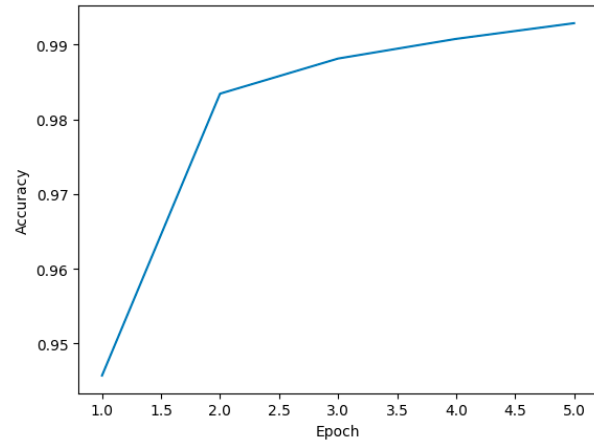
Model 2 kết hợp convolution layer và linear layer với cấu trúc như bảng bên dưới.

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(10, activation='softmax'))
```

Độ lỗi và độ chính xác của model trên tập train theo từng epoch được thể hiện như biểu đồ ở **hình 5** và **hình 6**. Độ chính xác của model 1 khi thực hiện đánh giá trên tập test là: 0.9919.



Hình 5. Loss của model 2



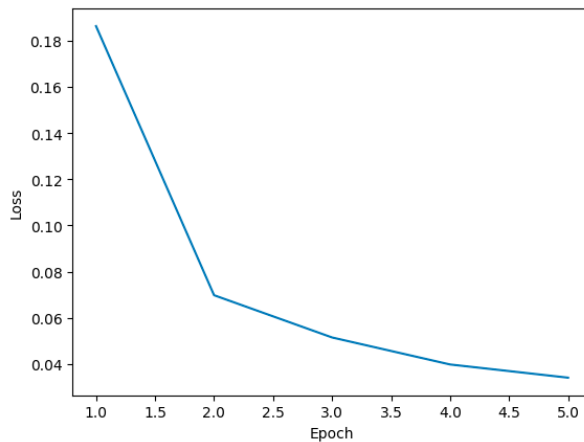
Hình 6. Accuracy của model 2

## MODEL 3

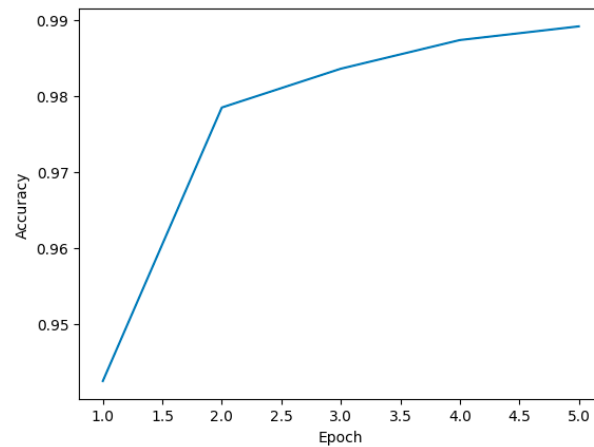
Model 3 kết hợp convolution layer, linear layer và dropout với cấu trúc như bảng bên dưới.

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(28,28,1)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax', name='preds'))
```

Độ lỗi và độ chính xác của model trên tập train theo từng epoch được thể hiện như biểu đồ ở **hình 7** và **hình 8**. Độ chính xác của model 1 khi thực hiện đánh giá trên tập test là: 0.9919.



Hình 7. Loss của model 3



Hình 8. Accuracy của model 3

## TỔNG KẾT

Bảng bên dưới cho sự khác biệt về số lượng tham số, thời gian thực hiện trên tập test và độ chính xác của 3 mô hình.

	Số lượng tham số	Thời gian đánh giá	Độ chính xác
Model 1	407,050	2s	0.9777
Model 2	93,322	2.4s	0.9919
Model 3	1,199,882	2.3s	0.9893

## NGUỒN THAM KHẢO

1. <https://github.com/raghakot/keras-vis>
2. <https://github.com/csbanon/mnist-classifiers>