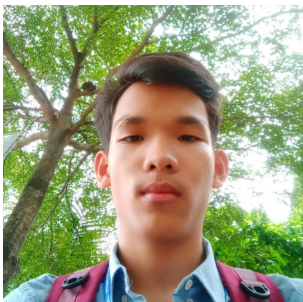


## PHẦN 1: THÔNG TIN TÓM TẮT

<b>Tên đề tài (IN HOA)</b>	NHẬN DẠNG SỐ VIẾT TAY
<b>Họ và tên (IN HOA)</b>	NGUYỄN ANH KHOA
<b>Lớp - MSSV</b>	CS114.K21.KHTN - 18520923
<b>Ảnh</b>	
<b>Link Github chứa repos CS114.K21</b>	<a href="https://github.com/anhkhoa039/CS114.K21.KHTN">https://github.com/anhkhoa039/CS114.K21.KHTN</a>
<b>Điểm đánh giá giữa kỳ (A B C D)</b>	- C
<b>Thành tích để tính điểm bonus</b>	- Không có
<b>Tóm tắt Bài tập quá trình</b>	<ul style="list-style-type: none"><li>- Số lần nộp bài tập Quá trình trên Classroom: 36/36</li><li>- Số lần nộp bài Thực hành trên Classroom: 6/7</li><li>- Tự đánh giá (90/100):</li></ul>
<b>Tóm tắt Đồ án Cuối kỳ (không quá 500 từ)</b>	<ul style="list-style-type: none"><li>- Mô tả bài toán: Input: Ảnh gồm 1 hoặc 1 số số viết tay Output: Dự đoán đó là số mấy</li></ul>

	<ul style="list-style-type: none"> <li>- Các thách thức: <ul style="list-style-type: none"> <li>- Khó khăn khi xây dựng bộ dataset: làm sao để chụp được nhiều hình 1 lúc để thu thập nhanh hơn. Dataset bị nhiễu không sạch do chụp không rõ</li> <li>- Dataset tự xây dựng quá nhỏ dẫn đến độ chính xác model không cao</li> <li>- Khó khăn khi nhận diện với ảnh gồm nhiều số</li> </ul> </li> <li>- cách giải quyết: <ul style="list-style-type: none"> <li>- Sử dụng opencv nhận diện cạnh và cắt ảnh theo cạnh. Chuyển về ảnh xám và lưu dưới dạng png để lọc ra những ảnh nhiễu</li> <li>- Tăng độ lớn của tập dữ liệu</li> <li>- Tách từng số trong ảnh gồm nhiều số thành các ảnh riêng biệt</li> </ul> </li> <li>- kết quả: <ul style="list-style-type: none"> <li>- với bộ dataset tự xây dựng 2000 ảnh: 0.10333333333333333</li> </ul> </li> <li>- Tự đánh giá (70/100):</li> </ul>
<b>Link khác</b>	<ul style="list-style-type: none"> <li>- Link đến báo cáo chi tiết (pdf):</li> <li>- Link đến báo cáo slides (pdf):</li> <li>- Link đến báo cáo video (YouTube)</li> </ul>

## PHẦN 2: BÁO CÁO TÓM TẮT ĐỒ ÁN CUỐI KỲ

### I. Mô tả bài toán:

**Đề tài: nhận dạng số viết tay**

**Lý do chọn đề tài:** dataset có thể tự thu thập, đề tài khá gần gũi, ở mức độ cơ bản, dễ dàng vận dụng được kiến thức đã học

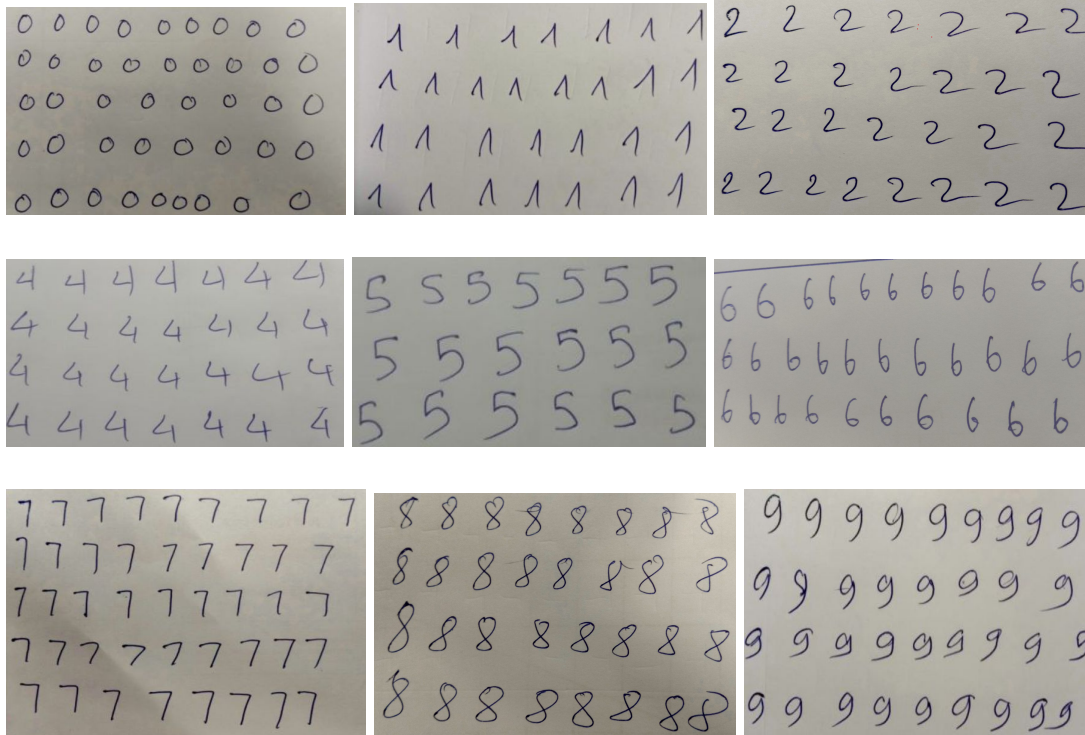
- **Input:** ảnh của 1 hoặc 1 số chữ số viết tay
- **Output:** dự đoán các số đó là số mấy

### II. Mô tả dữ liệu:

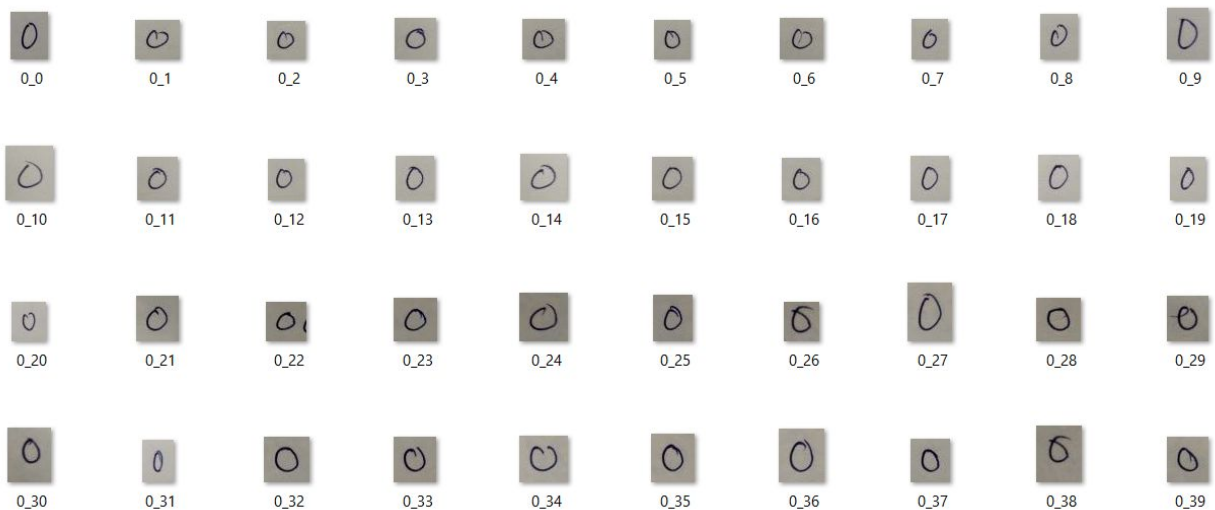
- Dữ liệu được 2 bạn tự thu thập: em (Nguyễn Anh Khoa -18520923) và Phan Gia Huy 18520068
- Dữ liệu gồm 2000 tấm ảnh các số chia đều từ 0-9.

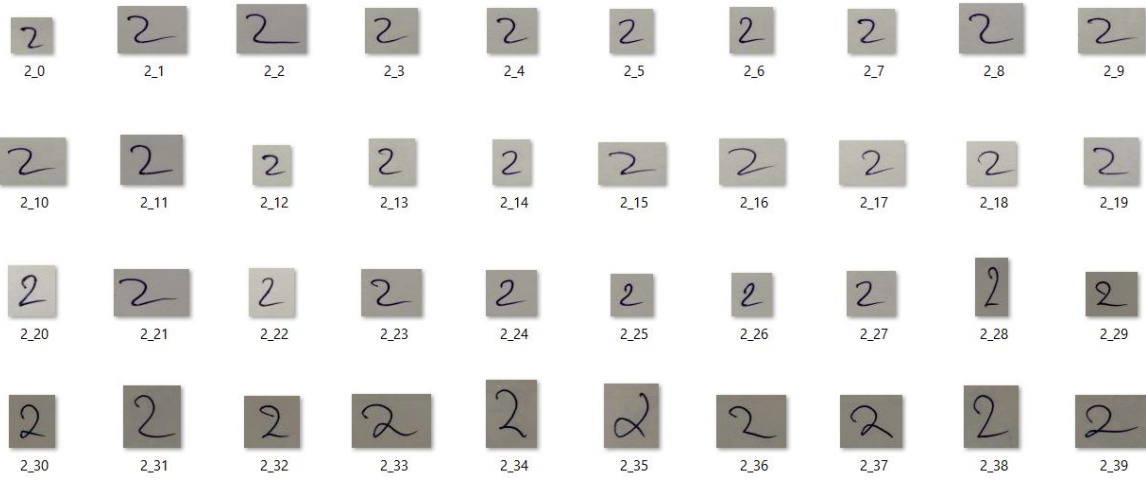
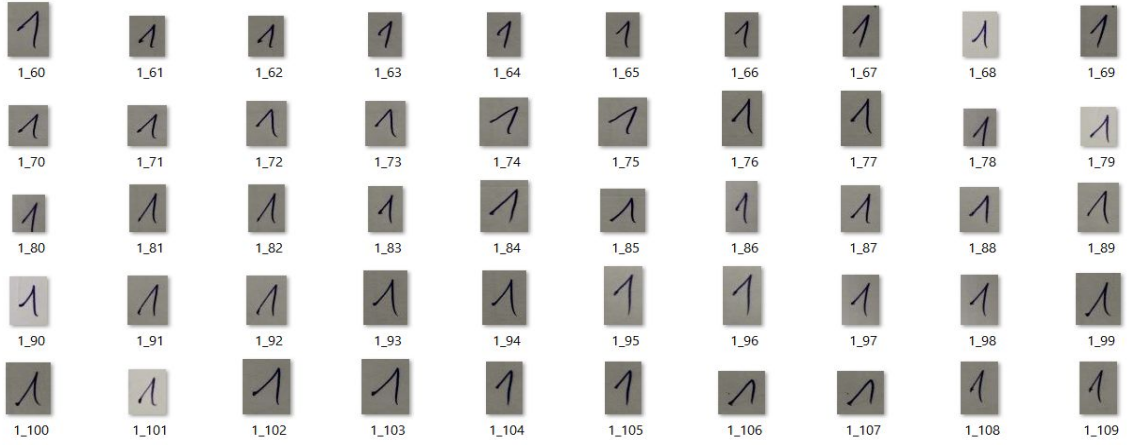
### III. Thu thập và Tiền xử lý dữ liệu:

- Ý tưởng: tạo một bộ dataset tương tự như MNIST dataset (bộ dữ liệu MNIST gồm 60k bức ảnh nhị phân kích thước 28x28 )
- ❖ Viết tay số vào giấy A4 và chụp lại bằng điện thoại:



❖ Sử dụng opencv cắt ra theo từng số, loại bỏ những ảnh cắt sai:

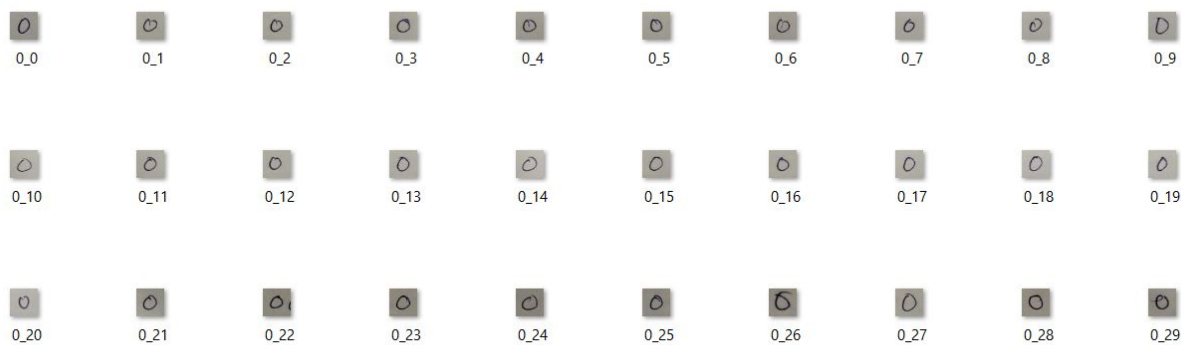




Một số ảnh bị cắt sai bị loại bỏ:



Resize ảnh về kích thước 28x28:



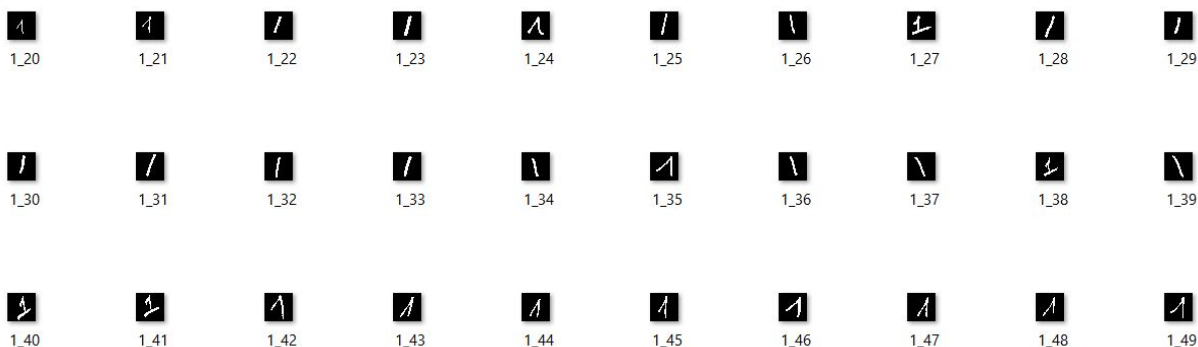
Từ ảnh RGB chuyển sang ảnh xám rồi chuyển sang nhị phân:

### ảnh sau khi chuyển về nhị phân

```
In [7]: img_b1 = cv_to_binary(path1)
         output1(img_b1)
```

[illegible]

Ảnh thu được: (ảnh nhị phân lưu dưới định dạng png nhằm mục đích hiển thị)



### Chuyển về dạng vector 1 chiều :

## chuyển ảnh về vector

```
In [9]: vec = cv_to_vector(img_b1)
```

```
print (len(vec))
print (vec)
```

[illegible]

## Gắn nhãn và xuất ra file CSV :

5 dữ liệu đầu:

Out[35]:

[illegible]

5 rows x 785 columns

Đọc file Data thu được:



	label	1x1	1x2	1x3	1x4	1x5	1x6	1x7	1x8	1x9	1x10	1x11	1x12	1x13	1x14	1x15	1x16	1x17	1x18	1x19	1x20	1x21	1x22	1x23	1x24	1x25	1x26	1x27	1x28	2x1	2x2	2x3
	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1996	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1997	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1998	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1999	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2000 rows × 785 columns																																

❑ File chi tiết: [make\\_data](#)

❑ File ảnh : [file\\_anh](#)

❑ File CSV: [data.csv](#)

#### IV. Trích xuất đặc trưng:

- Có 2 cách lấy đặc trưng: vector hóa và trích xuất đặc trưng sử dụng HOG (Histogram of Oriented Gradients)
- Cả 2 phương pháp lấy đặc trưng đều cho độ chính xác tương tự nhau. Ở đây em trích xuất đặc trưng sử dụng vector hóa
- Đối với ảnh nhỏ dùng để train, thì chưa cần dùng đến HOG

#### V. Mô hình

Model: SVM và KNN.

Với bài toán này, KNN là thuật toán đầu tiên được nghĩ tới, bởi vì đây là bài toán phân lớp, KNN là thuật toán cơ bản.

Với việc sử dụng SVM, SVM về bản chất không có khả năng thực hiện trực tiếp việc dự đoán nhiều class (Multiclass Classification) như một số model như Random Forest hay naive Bayes mà nó sẽ thực hiện dự đoán nhiều class thông qua

các Binary Classification. Có nhiều chiến lược (strategy) có thể được dùng để dự đoán Multiclass Classification thông qua Binary Classification như OvA(one-versus-all) hay OvO(one-versus-one), trong đó SVM sử dụng OvO, có nghĩa là train Binary Classifier cho mỗi cặp class, tức là sẽ có  $C^2_n$  classifier(cụ thể trong bài toán này có 10 class, vậy có 45 classifier sẽ được train), và instance sẽ lần lượt qua số classifier này, sau đó nó sẽ dự đoán class cho instance dựa trên class mà được lựa chọn nhiều nhất trong các Binary classifier

## VI. Nhận xét và Đánh giá

SVC					KNN				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	1.00	0.30	0.47	23	0	1.00	0.30	0.47	23
1	0.00	0.00	0.00	20	1	0.00	0.00	0.00	20
2	0.00	0.00	0.00	29	2	0.00	0.00	0.00	29
3	0.00	0.00	0.00	23	3	0.00	0.00	0.00	23
4	0.00	0.00	0.00	22	4	0.00	0.00	0.00	22
5	0.00	0.00	0.00	20	5	0.00	0.00	0.00	20
6	0.07	1.00	0.13	13	6	0.00	0.00	0.00	13
7	0.00	0.00	0.00	18	7	0.09	1.00	0.17	18
8	0.00	0.00	0.00	17	8	0.00	0.00	0.00	17
9	0.00	0.00	0.00	15	9	0.00	0.00	0.00	15
ccuracy			0.10	200	accuracy			0.12	200
cro avg	0.11	0.13	0.06	200	macro avg	0.11	0.13	0.06	200
ted avg	0.12	0.10	0.06	200	weighted avg	0.12	0.12	0.07	200

- Mô hình không tốt, accuracy chỉ ~12%. Tốt nhất xấp xỉ 96%. Những nguyên nhân gây nên là:
  - Dataset quá nhỏ
  - Dữ liệu không đủ tốt, vẫn chưa loại bỏ hết được dữ liệu nhiễu
  - Ảnh tự chụp không đảm bảo chất lượng
- Cách khắc phục:
  - Xây dựng bộ dữ liệu lớn hơn
  - Sử dụng bộ dữ liệu MNIST

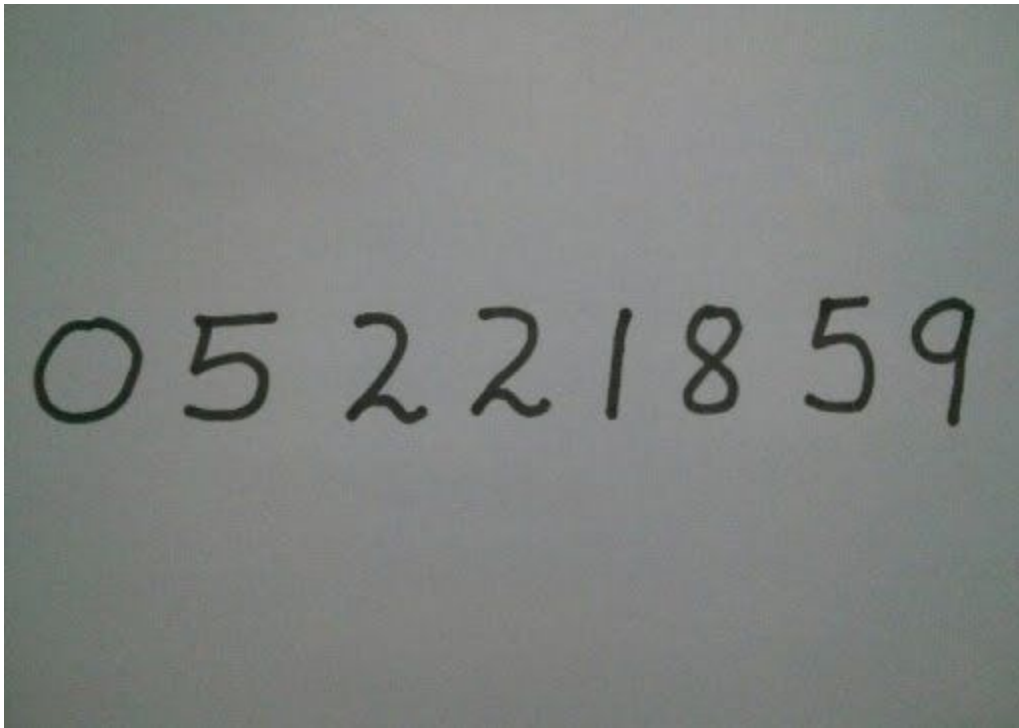
- Chọn cách trích xuất đặc trưng và model hiệu quả hơn, sử dụng mạng nơ-ron

## **V. Tổng kết:**

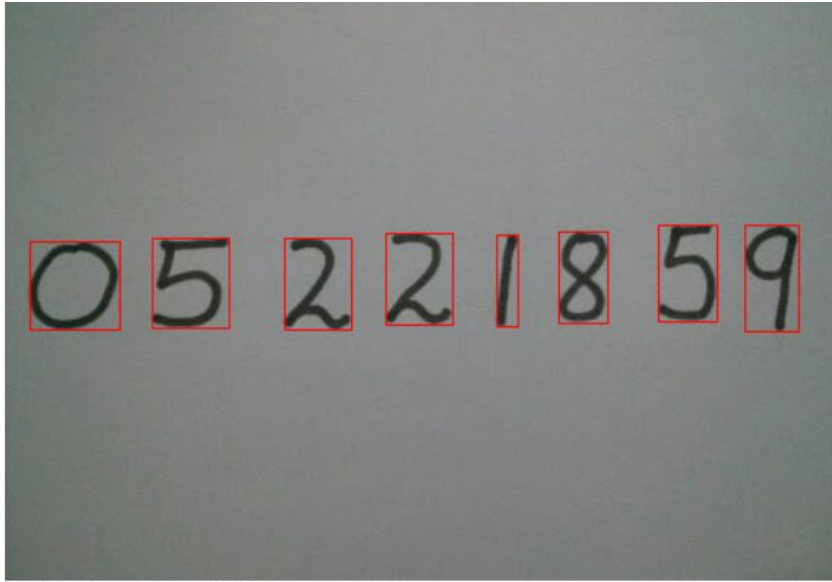
Qua đề tài này, dù model không thực sự hiệu quả, nhưng qua đó học được các bước cơ bản để xây dựng một mô hình học máy, trải nghiệm quá trình thu thập và tiền xử lý dữ liệu, cách chọn đặc trưng và chọn model, nhận xét đánh giá và khắc phục những khuyết điểm

## **VI. DEMO:**

- Từ ảnh input đầu vào, sử dụng cv2 tạo viền bao xung quanh mỗi chữ số và cắt ra từng ảnh nhỏ, sau đó nhận diện từng ảnh nhỏ và xuất ra kết quả
- Ảnh ban đầu



Kết quả:



'[0][2][0][0][2][0][0][0]'