


PHẦN 1: THÔNG TIN TÓM TẮT

| | |
|---|---|
| Tên đề tài (IN HOA) | NHẬN DẠNG SỐ VIẾT TAY |
| Họ và tên (IN HOA) | PHAN GIA HUY |
| Lớp - MSSV | CS114.K21.KHTN - 18520068 |
| Ảnh |  |
| Link Github chứa repos CS114.K21 | - https://github.com/huyphangia/CS114.K21.KHTN |
| Điểm đánh giá giữa kỳ (A B C D) | - C |
| Thành tích để tính điểm bonus | - Không có |
| Tóm tắt Bài tập quá | - Số lần nộp bài tập Quá trình trên Classroom: 36/36 |

| | |
|---|--|
| trình | <ul style="list-style-type: none"> - Số lần nộp bài Thực hành trên Classroom: 6/7 - Tự đánh giá (90/100): |
| Tóm tắt Đồ án Cuối kỳ (không quá 500 từ) | <ul style="list-style-type: none"> - Mô tả bài toán: Đồ án tự động nhận diện điểm và mã số sinh viên được viết tay <p>Input: Ảnh chụp có gồm hai khung có chứa MSSV và điểm của sinh viên</p> <p>Output: Lưu điểm của sinh viên và mã số sinh viên trong input vào file csv</p> <ul style="list-style-type: none"> - các thách thức: <ul style="list-style-type: none"> - Khó khăn khi xây dựng bộ dataset: Dataset bị nhiễu không sạch do chụp không rõ - Dataset tự xây dựng quá nhỏ dẫn đến độ chính xác model không cao - Đôi khi không nhận diện được các contour do bị nhiễu hoặc ảnh chụp không chất lượng - Đối với điểm có phần thập phân chưa nhận diện được triệt để - cách giải quyết: <ul style="list-style-type: none"> - Xử dụng các feature engineering mới hơn, chụp ảnh trong cường độ ánh sáng tốt hơn, - Tăng độ lớn của tập dữ liệu - Đòi hỏi tập train phải có ảnh kèm thêm dấu . ở sau chữ số ở một số ảnh để có thể nhận diện được nhiều hơn điểm có phần thập phân - kết quả: <ul style="list-style-type: none"> - với bộ dataset tự xây dựng 2000 ảnh: 0.125 - Tự đánh giá (70/100): |
| Link khác | <ul style="list-style-type: none"> - Link đến báo cáo chi tiết (pdf): - Link đến báo cáo slides (pdf): - Link đến báo cáo video (YouTube) |

PHẦN 2: BÁO CÁO TÓM TẮT ĐỒ ÁN CUỐI KỲ

I. Mô tả bài toán:

Đề tài: Đồ án tự động nhận diện điểm và mã số sinh viên được viết tay

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

- Lấy ý tưởng từ vấn đề của một giáo viên dạy chính trị, trong một lần đứng lớp đã kể rằng cô và đồng nghiệp rất hay bị nhầm khi nhập điểm thi, điểm kiểm tra từ bài mà mình chấm vào file trên excel vì số lượng bài rất nhiều, phải mất nhiều công sức để rà soát xem mình nhập đã đúng chưa và đôi khi dẫn đến tình trạng sinh viên khiếu nại, phúc khảo điểm.

- Cá nhân em nghĩ ra ý tưởng xây dựng hệ thống mà chỉ cần thầy/cô chụp hình phần bài gồm MSSV và điểm của học sinh, hệ thống sẽ nhận diện và ghi phần điểm cũng như MSSV của học sinh tương ứng vào từng hàng trong file csv(tương tự như excel)
- Để hệ thống có thể hoạt động một cách bình thường, chúng ta cần thêm một tờ giấy A4 như hình vào tờ làm bài của mỗi thí sinh

Họ tên:

MSSV

Điểm

- Tờ giấy này gồm họ tên, hai khung, khung đầu tiên yêu cầu thí sinh ghi MSSV, và khung thứ hai sẽ là điểm của bài làm mà giáo viên sẽ ghi trực tiếp vào, phần còn trống của tờ giấy có thể được tận dụng để ghi đề các câu hỏi, tuy nhiên thí sinh và người chấm không được tự ý viết gì thêm lên tờ giấy (ý tưởng tạo một tờ giấy đặc biệt để cho hệ thống hoạt động tốt hơn dựa trên ý tưởng tờ giấy trắc nghiệm khi thi THPT quốc gia cũng có những phần đặc biệt để máy có thể nhận biết và chấm)
- **Input:** Ảnh chụp có gồm hai khung có chứa MSSV và điểm của sinh viên
- **Output:** Lưu điểm của sinh viên và mã số sinh viên trong input vào file csv

Các bước thực hiện:

- Bước 1: Xây dựng một model nhận diện chữ số viết tay bằng dữ liệu tự thu thập được
- Bước 2: Nhận diện khung chứa điểm và khung chứa MSSV, sau đó trích xuất hai khung chứa số ra hai ảnh khác nhau
- Bước 3: Nhận diện các số trong dãy, tách chúng thành các số đơn lẻ và trích xuất thành các ảnh tương ứng
- Bước 4: Dùng Model đã train ở bước 1 để nhận diện các ảnh đã cắt ở bước 3 và lưu vào file csv

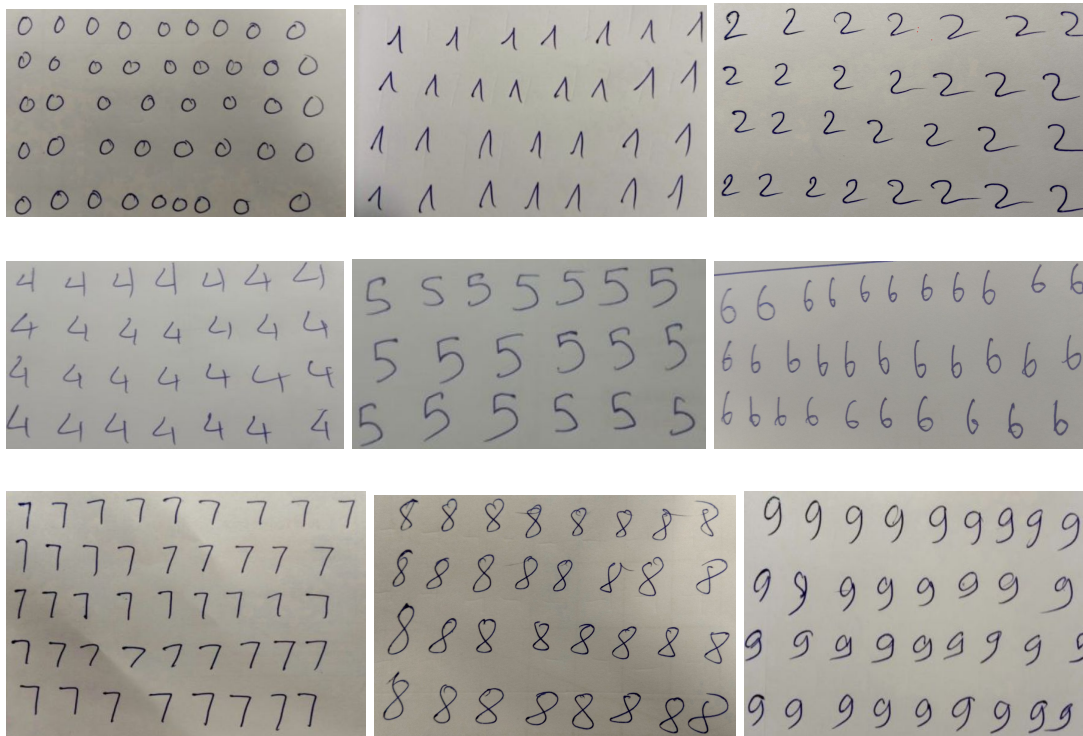
Bước 1: Thu thập dữ liệu và xây dựng model

II. Mô tả dữ liệu:

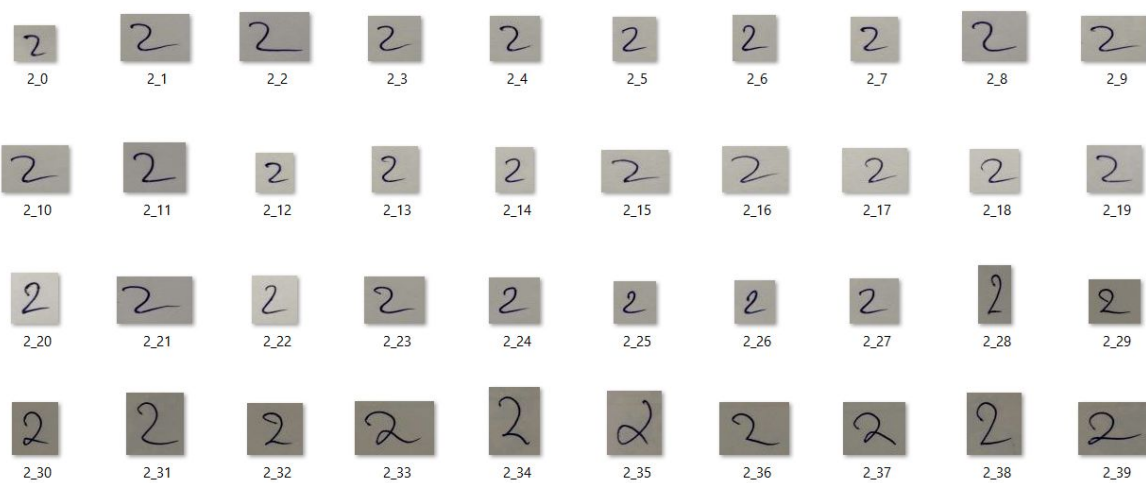
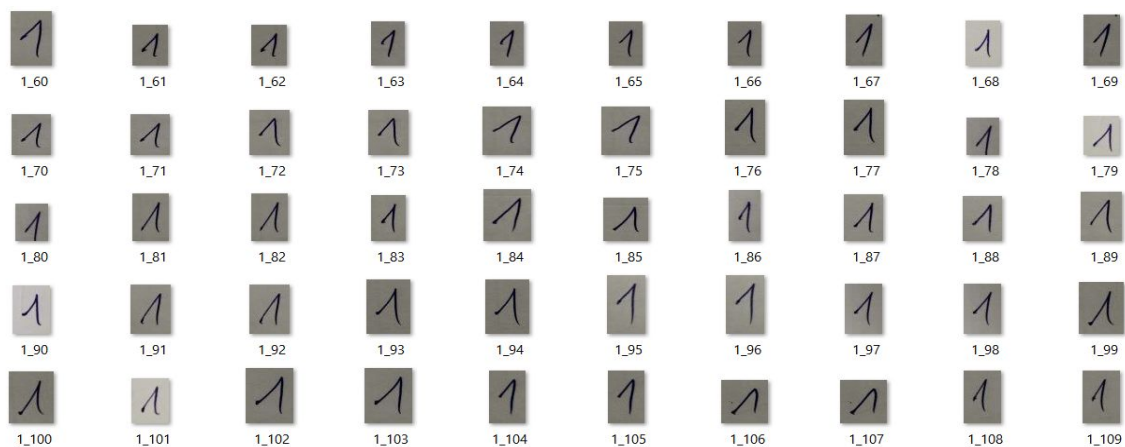
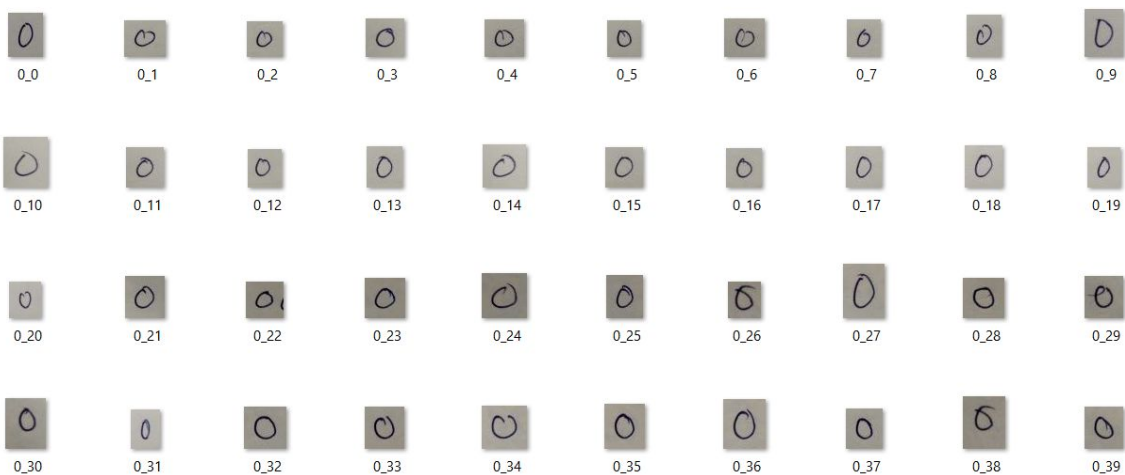
- Dữ liệu được 2 bạn tự thu thập: Nguyễn Anh Khoa -18520923 và Phan Gia Huy-18520068
- Dữ liệu gồm 2000 tấm ảnh các số chia đều từ 0-9.

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:

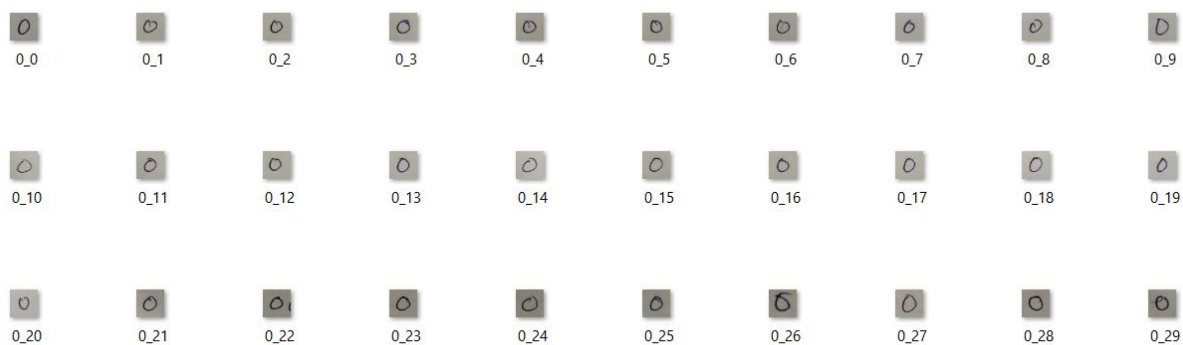




Một số ảnh bị cắt sai bị loại bỏ:



Resize ảnh về kích thước 28x28:



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

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)
- Đối với ảnh nhỏ dùng để train, thì chưa cần dùng đến HOG vì thế ở đây em sẽ không dùng đến HOG
- Chia dữ liệu train-test theo tỉ lệ 9-1

V. Mô hình

- Model: SVM và KNN.
- Lý do chọn SVM: do tập dữ liệu nhỏ và thuật toán SVM rất phù hợp với bộ dữ liệu không quá lớn và đối với những bài toán có số chiều (dimention) lớn, thì SVM là một trong những lựa chọn rất tốt, trong bài toán này có 784 dimention trong vector, khá lớn. Hai lý do trên là nguyên do chính để em chọn SVM
- Khái quát về cách hoạt động của 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_n^2 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

- Lý do chọn KNN : Đối với các bài toán Multiclass Classification, thuật toán KNN tỏ ra rất hữu dụng, đơn giản .

Khái quát về cách thức hoạt động của KNN:

Thuật toán dựa trên các các neighbor gần nhất của nó hay nói rõ hơn là model sẽ dự đoán dựa vào K(số neighbor, đây là hyper-parameter mà ta sẽ điều chỉnh) neighbor gần nhất của nó, và trong K neighbor, class nào có nhiều neighbor thuộc về nó nhất sẽ được gán cho instance cần dự đoán

VI. Đánh giá và Fine-tuning:

- Đánh giá:

- Dùng cross_val_score
- Số fold là 3 cho các loại model
- Dự đoán accuracy của model trên từng fold dựa trên model được train ở các fold còn lại
- Tiết kiệm được dữ liệu và có thể biết được phần nào accuracy của model để đến bước tiếp theo
- Có dùng dữ liệu đã được scale để đánh giá thử, tuy nhiên kết quả đạt được gần giống như dữ liệu chưa được scale, nên trong bước tiếp theo dùng dữ liệu chưa scale

- Fine-tuning:

- Dùng gridsearch
- Tìm các hyper-parameter tối ưu phù hợp với từng thuật toán
- Ở đây, đối với KNN là tìm weights và n_neighbors, sau khi chạy đã tìm được weights là uniform và n_neighbors=1 là những hyper-parameter tối ưu nhất với thuật toán
- Đối với SVM là tìm C, tìm được C=0.001

VII. Test trên tập testing data và kết luận

- Kết quả có tăng tuy nhiên không đáng kể, kết quả vẫn chưa như ý muốn
- Kết luận:

- Do dữ liệu tự thu thập quá nhỏ, dẫn đến hiện tượng overfit trên training data
- Ảnh tự chụp chất lượng không đảm bảo
- Hướng khắc phục:
 - Xây dựng bộ data to hơn
 - Sử dụng bộ data mnist
 - Dùng các kỹ thuật feature engineering mới
 - Sử dụng các thuật toán phức tạp hơn trong deeplearning

Bước 2: Nhận diện khung chứa điểm và khung chứa MSSV, sau đó trích xuất hai khung chứa số ra hai ảnh khác nhau

2.1 Chụp ảnh và load ảnh lên colab (do người dùng thực hiện)

2.2 Tiến hành nhận diện khung và tách ảnh

- Ở đây, trước tiên là xử lý ảnh bằng cách chuyển ảnh xám, sau đó lọc nhiễu bằng GaussianBlur
- Tìm edge bằng method cv2.canny
- Tìm contour dựa vào ảnh đã qua bước xử lý edge(tham số edged.copy() trong cv2.findContours), em đã thử một vài cách nhưng không hiệu quả, như tìm contour thông qua tham số ảnh được chỉnh bằng các method như morphologyEx hay threshold
- Dùng method cv2.approxPolyDP để tính xấp xỉ đa giác
- Vì hai khung cần nhận diện hình chữ nhật, nên tìm contour có số cạnh là 4, em đã thử cách một khung hình chữ nhật, một khung hình tròn và sẽ tùy chỉnh thuật toán để có thể nhận ra nhanh hơn, chính xác hơn tuy nhiên cũng không hiệu quả, vì thế em dùng cách để cả hai khung là hình chữ nhật, và dùng thuật toán trên
- Sau khi xác định contour thỏa điều kiện thì tiến hành cắt hai khung thành hai ảnh nhỏ riêng biệt

Bước 3+4: Từ hai ảnh ở bước 2, nhận diện, tách dãy số thành các số riêng lẻ và predict các số ấy, lưu vào file csv

- Xử dụng hai ảnh đã được cắt từ bước 2
- Các bước xử lý ảnh tương tự như bước 2
- Sau đó tìm contour có trong hai bức ảnh và dùng model đã train để dự đoán các contour đó

- Ghi kết quả dự đoán được vào file Csv

Tổng kết:

Qua quá trình thực hiện đề án đã giúp em tìm hiểu rõ hơn về các bước thực hiện một dự án về máy học trong thực tế, các công đoạn thu thập data, xây dựng model, đánh giá model, tinh chỉnh các tham số chỉ là bề nổi, thực tế trong mỗi bước đòi hỏi có rất nhiều công đoạn nhỏ, mỗi công đoạn ấy tuy vậy lại góp phần không nhỏ vào việc thực hiện dự án, như trong lúc thu thập data phải chỉnh sửa, gán nhãn, lọc nhiễu,.. đây là ví dụ về những công đoạn phụ ẩn trong mỗi bước lớn trong 7 bước lớn để xây dựng dự án máy học mà nếu không bắt tay vào làm thì không thể học hỏi và biết cách xử lý được. Tuy đề án lần này, model em làm chưa được như mong muốn dẫn đến hệ thống có kết quả chưa tốt, nhưng đã giúp em học hỏi thêm rất nhiều thứ từ việc lọc data, chọn model, đánh giá, tinh chỉnh, cách trích xuất hình ảnh,... Quan trọng nhất là kiến thức về Machine Learning được nâng cao hơn, bên cạnh đó là việc tự mình tìm hiểu, tạo ra ,so sánh, đánh giá, tối ưu những các phương án để xử lý vấn đề, tiếp cận với các bước làm dự án máy học và xây dựng hệ thống dựa trên máy học trong thực tế.

- Các thách thức:

- Khó khăn khi xây dựng bộ dataset: Dataset bị nhiễu không sạch do chụp không rõ
- Dataset tự xây dựng quá nhỏ dẫn đến độ chính xác model không cao
- Đôi khi không nhận diện được các contour do bị nhiễu hoặc ảnh chụp không chất lượng
- Đối với điểm có phân thập phân chưa nhận diện được triệt để

- **Cách giải quyết:**

- Xử dụng các feature engineering mới hơn, chụp ảnh trong cường độ ánh sáng tốt hơn,
- Tăng độ lớn của tập dữ liệu
- Đòi hỏi tập train phải có ảnh kèm thêm dấu “.” ở sau chữ số ở một số ảnh để có thể nhận diện được nhiều hơn điểm có phần thập phân