BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**🙞 🕮 🙜**

**BÁO CÁO**

**MÁY HỌC NÂNG CAO**

Đề tài

**NHẬN DẠNG KÝ TỰ SỐ**

**SUPPORT VECTOR MACHINE**

**GIẢNG VIÊN HƯỚNG DẪN**

**PGS.TS PHẠM NGUYÊN KHANG**

**tHÀNH VIÊN NHÓM**

**b1609830 lê thanh lương**

**B1611128 lâm thanh hòa**

*Cần Thơ, 06/2020*

**NHẬN XÉT CỦA GIẢNG VIÊN**

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------------------------------------------------------------

Cần Thơ, ngày 22 tháng 06 năm 2020

(GVHD ký và ghi rõ họ tên)

**LỜI CẢM ƠN**

Để có được bài báo cáo này, chúng em xin được bày tỏ lòng biết ơn chân thành và sâu sắc đến Thầy Phạm Nguyên Khang – người đã trực tiếp tận tình hướng dẫn, giúp đỡ chúng em trong việc học tập. Trong suốt quá trình thực hiện báo cáo, nhờ những sự chỉ bảo và hướng dẫn quý giá đó mà bài báo cáo này được hoàn thành một cách tốt nhất.

Chúng em cũng xin gửi lời cám ơn chân thành đến các Thầy Cô Giảng viên Đại học Cần Thơ, đặc biệt là các Thầy Cô ở Khoa CNTT & TT, những người đã truyền đạt những kiến thức quý báu trong thời gian qua.

Chúng em cũng xin chân thành cảm ơn bạn bè cùng với gia đình đã luôn động viên, khích lệ và tạo điều kiện giúp đỡ trong suốt quá trình thực hiện để chúng em có thể hoàn thành bài báo cáo một cách tốt nhất.

Tuy có nhiều cố gắng trong quá trình thực hiện bài báo cáo, nhưng không thể tránh khỏi những sai sót. Chúng em rất mong nhận được sự đóng góp ý kiến quý báu của quý Thầy và các bạn để bài báo cáo hoàn thiện hơn.

Cần Thơ, ngày 21 tháng 06 năm 2020

**MỤC LỤC**

[PHẦN GIỚI THIỆU 5](#_Toc43720611)

[1. Đặt vấn đề 5](#_Toc43720612)

[2. Lịch sử giải quyết vấn đề 5](#_Toc43720613)

[3. Mục tiêu đề tài 5](#_Toc43720614)

[4. Đối tượng và phạm vi nghiên cứu 5](#_Toc43720615)

[5. Phương pháp nghiên cứu 6](#_Toc43720616)

[6. Kết quả đạt được 6](#_Toc43720617)

[7. Bố cục bài báo cáo 6](#_Toc43720618)

[PHẦN NỘI DUNG 8](#_Toc43720619)

[CHƯƠNG 1: MÔ TẢ BÀI TOÁN 8](#_Toc43720620)

[1. Mô tả chi tiết bài toán 8](#_Toc43720621)

[2. Vấn đề và giải pháp liên quan đến bài toán 10](#_Toc43720622)

[2.1. Thư viện OpenCV 10](#_Toc43720623)

[2.2. Thuận toán HOG 11](#_Toc43720624)

[CHƯƠNG 2 : THIẾT KẾ VÀ CÀI ĐẶT 15](#_Toc43720625)

[1. Thiết kế hệ thống 15](#_Toc43720626)

[2. Cài đặt giải thuật 15](#_Toc43720627)

[CHƯƠNG 3: GIAO DIỆN 17](#_Toc43720628)

[1. Giao diện nhận dạng ký tự số 17](#_Toc43720629)

[*2.* Giao diện kết quả nhận dạng 18](#_Toc43720630)

[PHẦN KẾT LUẬN 19](#_Toc43720631)

[1. Kết quả đạt được 19](#_Toc43720632)

[2. Hướng phát triển 19](#_Toc43720633)

[TÀI LIỆU THAM KHẢO 20](#_Toc43720634)

**DANH MỤC HÌNH**

*Hình 1: Siêu phẳng với lề cực đại cho một SVM………………………………….09*

*Hình 2: Quy trình thuật toán HOG………………………………………………..13*

*Hình 3: Sơ đồ hoạt động của chương trình giao diện…………………………..15*

*Hình 4: Sơ đồ quá trình tiền xử lý dữ liệu………………………………………..15*

*Hình 5: Các chỉ số xây dựng mô hình…………………………………………….16*

*Hình 6: Các chỉ số đánh giá mô hình………………………………………………16*

*Hình 7: Giao diện chương trình nhận dạng với SVM…………………………..17*

*Hình 8: Giao diện kết quả…………………………………………………………..18*

**DANH MỤC CÁC CHŨ VIẾT TẮT**

|  |  |
| --- | --- |
| SVM | Support Vector Machine |
| HOG | Histogram of Oriented Gradients |
| KNN | K-nearest neighbors |
| CNN | Convolutional Neural Network |

**TÓM TẮT**

Trong bài viết này, chúng tôi sử dụng giải thuật máy vector hỗ trợ (SVM) để cho nhận dạng ký tự số viết tay. Đầu tiên chúng tôi thực hiện tải dữ liệu, sau đó tạo một hog\_feature lưu dữ liệu, thực hiện xây dựng mô hình và đánh giá mô hình để tìm một mô hình tốt nhất. Cuối cùng là xây dựng giao diện sử dụng các thư viện OpenCV để đọc dữ liệu ảnh, Tkinter xây dựng giao diện tương tác với người dùng, và thuật toán HOG để trích xuất các đặc trưng của phát hiện đối tượng cần nhận dạng trong hình ảnh. Kết quả nhận diện trên ảnh đầu vào bất kỳ gồm một số hay dãy số đều cho độ chính xác tương đối cao.

# PHẦN GIỚI THIỆU

## 1. Đặt vấn đề

Ngày nay công nghệ thông tin ngày càng phát triển và có vai trò hết sức quan trọng không thể thiếu trong cuộc sống hiện tại. Con người ngày càng tạo ra những cỗ máy thông Minh có khả năng tự nhận biết và xử lí được các công việc một cách tự động, phục vụ cho lợi ích của con người. Trong những năm gần đây, một trong những bài toán được sự quan tâm và tốn nhiều công sức nhất của lĩnh vực công nghệ thông tin, đó chính là bài toán nhận dạng. Tuy mới xuất hiện chưa lâu nhưng nó đã rất được quan tâm vì tính ứng dụng thực tế của bài toán cũng nhưng sự phức tạp của nó. Trong bài báo cáo này dùng SVM (support vecter machine) để nhận dạng chữ số viết tay trên bộ dữ liệu MNIST.

## 2. Lịch sử giải quyết vấn đề

Các hệ thống trước đây như (LeCun et al, 1998), (Simard et al, 2003), (Kégl & BusaFekete, 2009) đều sử dụng các đặt trưng cơ bản từ ảnh ký tự như đường biên, cạnh, độ dày, giá trị mức xám, haarr-like, với các xử lý đặt thù như lấy mẫu, dao động các điểm ảnh, biến đổi ảnh, thêm dữ liệu ảo. Sau đó hệ thống nhận dạng huấn luyện các mô hình học tự động như K láng giềng (KNN), mạng nơ-ron, boosting, Deep learning. Các hệ thống này cơ bản đáp ứng yêu cầu của người sử dụng, tuy nhiên với mong muốn tìm ra giải pháp để nhận dạng số một cách hiệu quả nên bài cáo nghiên cứu kỹ thuật máy vector hỗ trợ (SVM), đây là phương pháp có thể giải quyết các bài toán với kích thước đầu vào lớn.

## 3. Mục tiêu đề tài

- Xây dựng và cài đặt mô hình SVM với độ chính xác cao

- Xây dựng thành công giao diện nhận dạng ký tự số với đầu vào là ảnh ký tự số.

## 4. Đối tượng và phạm vi nghiên cứu

- Các ký tự số viết tay.

- Kỹ thuật SVM

- Ảnh quét số viết tay.

- Các số tự nhiên từ 0 - 9

- Các phương pháp, giải thuật về nhận dạng.

- Sử dụng thư viện xử lý ảnh openCV.

- Sử dụng thư viện Tkinter trong Python

- Sử dụng thuật toán HOG để phát hiện đối tượng.

## 5. Phương pháp nghiên cứu

- Đọc tài liệu chuyên ngành về máy học nâng cao, xử lý ảnh, …

- Xây dựng giải thuật và thiết kế chương trình giao diện

- Lập trình và kiểm thử.

## 6. Kết quả đạt được

- Xây dựng thành công giao diện nhận dạng ký tự số.

- Vận dụng được thuật toán SVM vào xây dựng mô hình nhận dạng.

- Vận dụng được các thư viện đặt ra.

- Nhận dạng đúng ký tự số.

## 7. Bố cục bài báo cáo

**Phần giới thiệu**

Giới thiệu tổng quát về đề tài.

**Phần nội dung**

* **Chương 1** : Mô tả bài toán.
* **Chương 2** : Thiết kế, cài đặt giải thuật
* **Chương 3** : Kiểm thử hệ thống và đánh giá độ chính xác.

**Phần kết luận**

Trình bày kết quả đạt được và hướng phát triển hệ thống.

# PHẦN NỘI DUNG

# CHƯƠNG 1: MÔ TẢ BÀI TOÁN

## Mô tả chi tiết bài toán

**1.1 Tổng quan về SVM**

Một máy vectơ hỗ trợ xây dựng một [siêu phẳng](https://vi.wikipedia.org/wiki/Si%C3%AAu_ph%E1%BA%B3ng) hoặc một tập hợp các siêu phẳng trong một không gian nhiều chiều hoặc vô hạn chiều, có thể được sử dụng cho phân loại, hồi quy, hoặc các nhiệm vụ khác. Một cách trực giác, để phân loại tốt nhất thì các siêu phẳng nằm ở càng xa các điểm dữ liệu của tất cả các lớp (gọi là hàm lề) càng tốt, vì nói chung lề càng lớn thì [sai số tổng quát hóa](https://vi.wikipedia.org/w/index.php?title=Sai_s%E1%BB%91_t%E1%BB%95ng_qu%C3%A1t_h%C3%B3a&action=edit&redlink=1) của thuật toán phân loại càng bé.

Trong nhiều trường hợp, không thể phân chia các lớp dữ liệu một cách tuyến tính trong một không gian ban đầu được dùng để mô tả một vấn đề. Vì vậy, nhiều khi cần phải [ánh xạ](https://vi.wikipedia.org/wiki/%C3%81nh_x%E1%BA%A1) các điểm dữ liệu trong không gian ban đầu vào một không gian mới nhiều chiều hơn, để việc phân tách chúng trở nên dễ dàng hơn trong không gian mới. Để việc tính toán được hiệu quả, ánh xạ sử dụng trong thuật toán SVM chỉ đòi hỏi [tích vô hướng](https://vi.wikipedia.org/wiki/T%C3%ADch_v%C3%B4_h%C6%B0%E1%BB%9Bng) của các vectơ dữ liệu trong không gian mới có thể được tính dễ dàng từ các tọa độ trong không gian cũ. Tích vô hướng này được xác định bằng một hàm hạt nhân *K*(*x*,*y*) phù hợp. Một siêu phẳng trong không gian mới được định nghĩa là tập hợp các điểm có tích vô hướng với một vectơ cố định trong không gian đó là một hằng số. Vectơ xác định một siêu phẳng sử dụng trong SVM là một tổ hợp tuyến tính của các vectơ dữ liệu luyện tập trong không gian mới với các hệ số *αi*. Với siêu phẳng lựa chọn như trên, các điểm *x* trong không gian đặc trưng được ánh xạ vào một siêu mặt phẳng là các điểm thỏa mãn:

Σ*i* *αi* *K* (*xi*, *x*) = hằng số.

Ghi chú rằng nếu *K* (*x*, *y*) nhận giá trị ngày càng nhỏ khi *y* xa dần khỏi *x* thì mỗi số hạng của tổng trên được dùng để đo độ tương tự giữa *x* với điểm *xi* tương ứng trong dữ liệu luyện tập. Như vậy, tác dụng của tổng trên chính là so sánh khoảng cách giữa điểm cần dự đoán với các điểm dữ liệu đã biết. Lưu ý là tập hợp các điểm *x* được ánh xạ vào một siêu phẳng có thể có độ phức tạp tùy ý trong không gian ban đầu, nên có thể phân tách các tập hợp thậm chí không lồi trong không gian ban đầu

**1.2 SVM tuyến tính:**

Ta có một tập huấn luyện D gồm n điểm có dạng: {\displaystyle {\mathcal {D}}}

Với mang giá trị 1 hoặc -1, xác định lớp của điểm . Mỗi là một vector thực p-chiều. Ta cần tìm siêu phẳng có lề lớn nhất chia tách các điểm có =1 và các điểm có = -1. Mỗi siêu phẳng đều có thể được viết dưới dạng một tập hợp các điểm **x** thỏa mãn

,



*Hình 1: Siêu phẳng với lề cực đại cho một SVM phân tách dữ liệu thuộc hai lớp. Các ví dụ nằm trên lề được gọi là các vectơ hỗ trợ*

Với ký hiệu cho tích vô hướng và **w** là một vectơ pháp tuyến của siêu phẳng. Tham số xác định khoảng cách giữa gốc tọa độ và siêu phẳng theo hướng vectơ pháp tuyến **w**.

Chúng ta cần chọn w bà b để cực đại hóa lề, hay khoảng cách giữa hai siêu mặt song song ở xe nhau nhất có thể trong khi vẫn phân chia dữ liệu. Các siêu mặt ấy được xác định bằng:

Để ý rằng nếu dữ liệu huấn luyện có thể được chia tách một cách tuyến tính, thì ta có thể chọn hai siêu phẳng của lề sao cho không có điểm nào ở giữa chúng và sau đó tăng khoảng cách giữa chúng đến tối đa có thể. Bằng hình học, ta tìm được khoảng cách giữa hai siêu phẳng là . Vì vậy ta muốn cực tiểu hóa giá trị . Để{\displaystyle \|\mathbf {w} \|} đảm bảo không có điểm dữ liệu nào trong lề, ta thêm vào các điều kiện sau:

Với mỗi ta có:

cho thuộc lớp thứ nhất hoặc

cho thuộc lớp thứ hai.

## 2. Vấn đề và giải pháp liên quan đến bài toán

### 2.1. Thư viện OpenCV

1. **Khái niệm:**

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOs lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần.

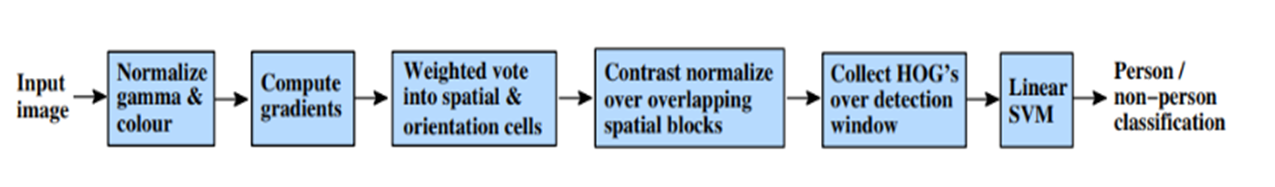
1. **Ứng dụng:**

* Nhận dạng ảnh
* Xử lý hình ảnh
* Phục hồi hình ảnh/video
* Thực tế ảo
* Các ứng dụng khác

### 2.2. Thuận toán HOG

1. **Giới thiệu về HOG**
   * Các khái niệm về HOG được nêu ra từ năm 1986 tuy nhiên đến năm 2005 mới được sử dụng rộng rãi sau khi Navneet Dalal và Bill Triggs công bố bài báo.
   * HOG là một bộ mô tả đặc trưng (feature descriptor) được xử dụng trong computer vision và xử lý ảnh.
     + Feature descriptor: là một thuật toán nhận đầu vào là ảnh và đầu ra là các vector. feature descriptor mã hóa các thông tin quan trọng thành một chuỗi các số, nó như là một dạng vân tay số cho các đặc tính ảnh có thể phân biệt lẫn nhau. Một cách lý tưởng, các thông tin này sẽ bất biến qua các phép biến đổi ảnh, do vậy mà ta vẫn có thể tìm được các đặc tính mặc dù ảnh đã bị biến đổi. Các ví dụ về feature descriptor bên cạnh HOG như: SIFT, SURF, …
   * Giải thuật HOG nằm ở bước thứ 2 trong giải thuật phát hiện đối tượng.
   * HOG lấy ra 2 ma trận: độ lớn gradient (gradient magnitude) và phương của gradient (gradient orientation). Sau đó kết hợp 2 thông tin này để ra biểu đồ phân phối histogram, trong đó độ lớn gradient được đếm theo các nhóm bins của phương gradient. Cuối cùng là thu được vector đặc trưng HOG đại diện cho histogram.
2. **Giải thuật HOG**

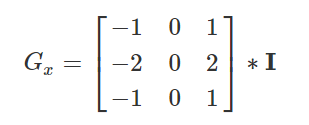
* Đọc ảnh
* Tiền xử lý (preprocessing): hình ảnh được đưa qua bước tiền xử lý để thực hiện các thao tác như cân bằng sáng, làm mờ, …
* Trích đặc trưng ảnh (feature extraction): bằng cách sử dụng các phương pháp rút trích đặc trưng ảnh ta sẽ thu được vector đặc trưng của ảnh. Nói một cách đơn giản đó chính là bạn sẽ mã hóa hình ảnh thành một vector, và vector này mang những đặc trưng (các số thực) đại diện cho ảnh đó.
* Huấn luyện mô hình học máy (training): với phương pháp truyền thống, ta thường sử dụng mô hình SVM trong machine learning để phân tách các vector đặc trưng thành các lớp cần phân loại.
* Kiểm thử (validation): sau khi huấn luyện xong mô hình học máy bạn cần phải đánh giá mô hình mình đã huấn luyện đạt độ chính xác là bao nhiêu phần trăm trên tập kiểm thử này. Khi đã hài lòng với kết quả kiểm thử, ta có thể dừng quá trình huấn luyện.

1. **Các bước thực hiện thuật toán HOG**

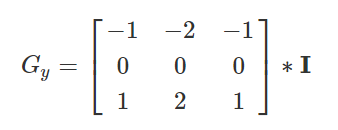
*Hình 2: Quy trình thuật toán HOG*

Tính toán Gradient.

* + Sử dụng phương pháp mặt nạ đạo hàm rời rạc (discrete derivative mask) theo một hoặc cả hai chiều ngang và dọc. Cụ thể phương pháp sẽ lọc ma trận cường độ ảnh với các bộ lọc như Sobel mask hoặc scharr
  + Để tính bộ lọc Sobel, phép tích chập của kernel kích thước 3x3 được thực hiện với ảnh ban đầu.
    - Đạo hàm theo chiều ngang:



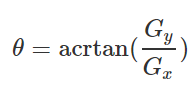
* + - Đạo hàm theo chiều dọc:



* + - Độ lớn gradient:



* + - Phương gradient:



* Ứng dụng của thuật toán:

- Tập train gồm 60000 dữ liệu.

- Tập test gồm 10000 dữ liệu

* Thuật toán HOG làm một trong những bộ mô tả đặc trưng mạnh giúp mã hóa hình ảnh thành một véctơ
* Nguyên lý hoạt động của thuật toán là dựa trên biểu diễn véctơ histogram của độ lớn gradient theo các bins của phương gradient áp dụng trên những vùng ảnh cục bộ.
* Các phương pháp chuẩn hóa được áp dụng giúp véctơ histogram tổng hợp trở nên bất biến với sự thay đổi về cường độ màu sắc của các bức ảnh có cùng nội dung nhưng khác nhau về cường độ màu sắc.
* Trong object detection, thuật toán tỏ ra khá hiệu quả khi ứng dụng tốt để phát hiện ký tự số với nhiều kích thước khác nhau.
* Hiệu quả hơn CNN với những tập dữ liệu nhỏ do CNN có thể hoạt động không chính xác do tập ảnh huấn lyện không đủ bao quát các khả năng.
* Đỡ tốn tài nguyên tính toán hơn CNN

Chính vì thế sử dụng HOG cho bài toán lần này.

# CHƯƠNG 2 : THIẾT KẾ VÀ CÀI ĐẶT

## Thiết kế hệ thống

Chọn ảnh nhận diện

Result Image ( ảnh gốc, và ảnh nhận diện)

*Hình 3: Sơ đồ hoạt động của chương trình giao diện*

## 2. Cài đặt giải thuật

**a) Quá trình tiền xử lý dữ liệu**

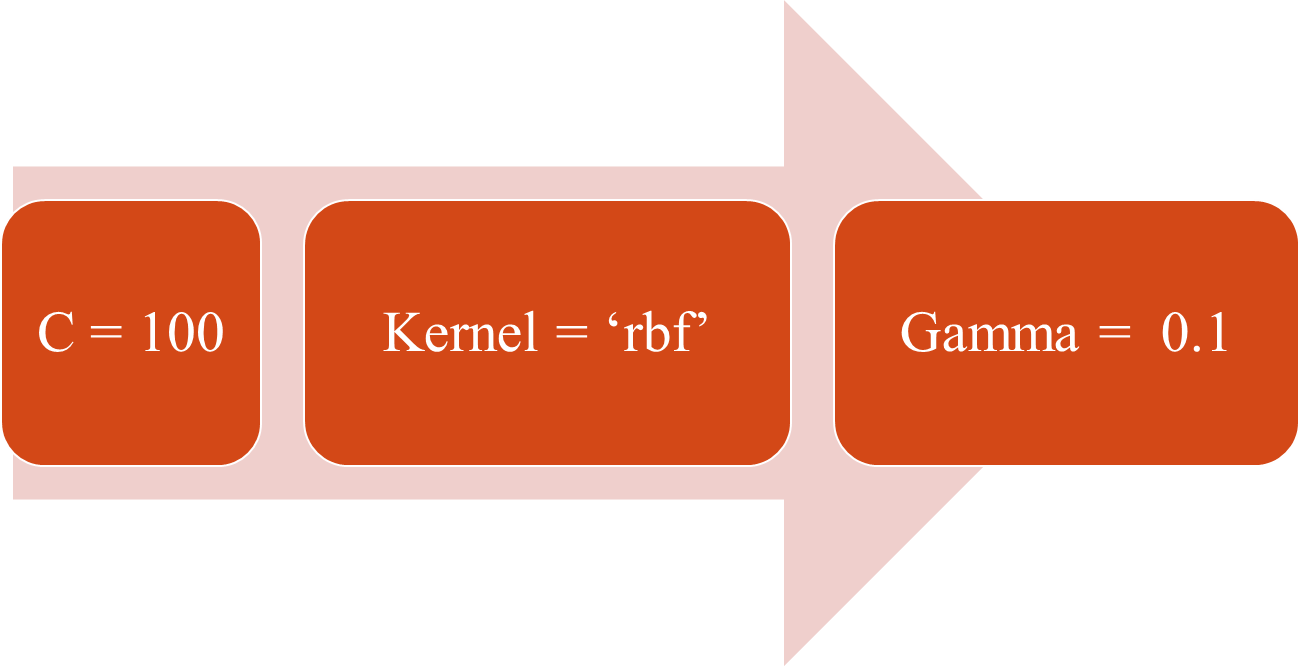
Read Image

Covert Gray Image

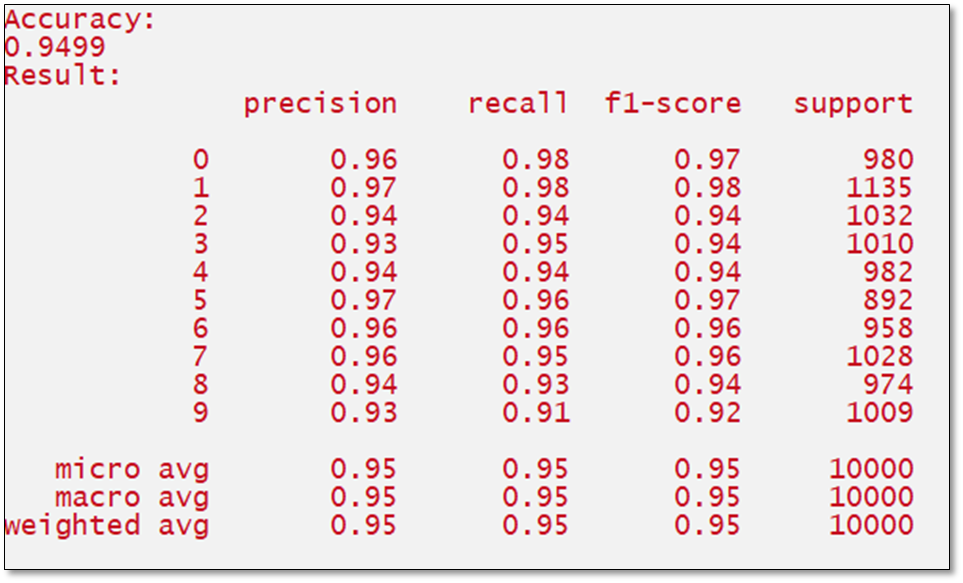
Làm mịn ảnh với GaussianBlur

Covert sang ảnh nhị phân

*Hình 4: Sơ đồ quá trình tiền xử lý dữ liệu*

* Sử dụng thư viện opencv đọc dữ liệu file ảnh từ thư mục Training và Testing
* Chuyển ảnh đầu vào thành ảnh có màu thang xám
* GaussianBlur được sử dụng để giãm nhiễu trong ảnh nói cách khác là làm mịn ảnh
* Chuyển ảnh về dạng ảnh nhị phân
* Ứng dụng thuật toán HOG để khoanh vùng đối tượng cần nhận dạng và nhận dạng đối tượng đó
  + - 1. **Xây dựng mô hình**
* Mô hình được xây dựng với các thông số C = 100
* Sử dụng kernel là ‘rbf’
* Chỉ số gamma = 0.1

*Hình 5: Các chỉ số xây dựng mô hình*

* + - 1. **Đánh giá mô hình**
* Mô hình được xây dựng có độ chính xác khoảng 94%
* Mô hình được đánh với các chỉ số precision, recall, f1-score, support

*Hình 6: Các chỉ số đánh giá mô hình*

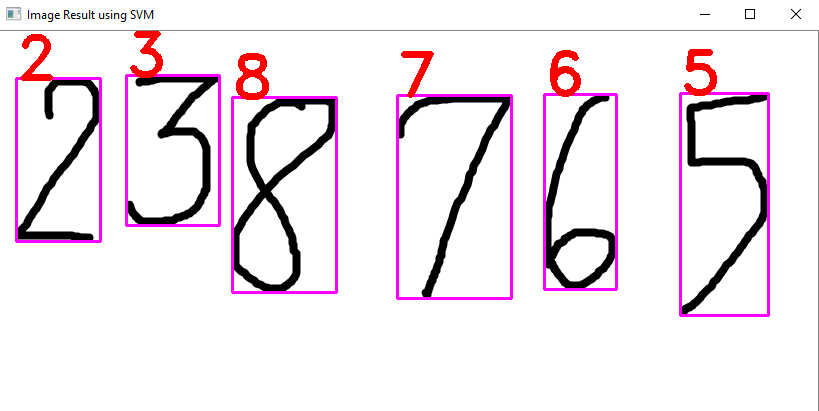
# CHƯƠNG 3: GIAO DIỆN

## Giao diện nhận dạng ký tự số

*Hình 7: Giao diện chương trình nhận dạng với SVM*

**- Chương trình giao diện gồm:**

* Phần 1: Tên đề tài, tên giảng viên hướng dẫn, thông tin thành viên nhóm thực hiện đề tài
* Phần 2: Chức năng của chương trình gồm chức năng chọn ảnh nhận dạng ( khi người dùng chọn ảnh muốn nhận dạng kết quả nhận được là một ảnh ở đầu ra với các số nhận dạng được), cuối cùng là chức năng thoát chương trình

1. Giao diện kết quả nhận dạng

*Hình 8: Giao diện kết quả*

* Kết quả nhận dạng : với ảnh đầu vào chứa các số là **2 3 8 7 6 5** ta nhận dạng được kết quả đầu ra như sau: 2 3 8 7 6 5

# PHẦN KẾT LUẬN

## 1. Kết quả đạt được

* Xây dựng được mô hình SVM với độ chính xác cao
* Từ mô hình đó, ta xây dựng thành công giao diện nhận dạng ký tự số với đầu vào là ảnh gồm một số hay một dãy số.

## 2. Hướng phát triển

* Xây dựng giao diện tối ưu hơn cho người dùng
* Xây dựng và áp dụng mô hình SVM để nhận dạng ký tự chữ viết tay
* Xây dựng giao diện nhận dạng trên nhiều ngôn ngữ lập trình khác nhau C#, C++, JAVA…..
* Ứng dụng vào thực tế xây dựng các chương trình nhận dạng biển số xe, nhận dạng mặt người….

# TÀI LIỆU THAM KHẢO

1. Giáo trình bài giảng trên lớp chuyên ngành Khoa học máy tính học phần máy học nâng cao.

2. Phạm Nguyên Khang, Đỗ Thanh Nghị” Nhận dạng ký tự số viết tay bằng giải thuật máy học” Tạp chí khoa học Trường Đại học Cần thơ, Phần A: Khoa học Tự nhiên, Công nghệ và Môi trường: 27 (2013): 64-71.