BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

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

#   

**MÔN HỌC**

**MÁY HỌC NÂNG CAO**

Đề tài

## **XÂY DỰNG ỨNG DỤNG TÍNH TOÁN ĐƠN GIẢN TỪ ẢNH**

**Sinh viên thực hiện:**

Nguyễn Nhật Khương B1609777

Nguyễn Đăng Khoa B1611129

Nguyễn Hoàng Châu B1609762

**Giảng viên hướng dẫn**

Phạm Nguyên Khang

Cần Thơ, 20/6/2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

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

#   



**MÔN HỌC**

**MÁY HỌC NÂNG CAO**

Đề tài

## **XÂY DỰNG ỨNG DỤNG TÍNH TOÁN ĐƠN GIẢN TỪ ẢNH**

**Sinh viên thực hiện:**

Nguyễn Nhật Khương B1609777

Nguyễn Đăng Khoa B1611129

Nguyễn Hoàng Châu B1609762

**Giảng viên hướng dẫn**

Phạm Nguyên Khang

Cần Thơ, 13/05/2019

NHẬN XÉT CỦA GIẢNG VIÊN

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Phạm Nguyên Khang đã giúp đỡ và hướng dẫn em tận tình trong suốt thời gian thực hiện đề tài này, tạo cho em những tiền đề, những kiến thức để tiếp cận, phân tích và giải quyết vấn đề. Nhờ đó mà em hoàn thành đề tài của mình được tốt hơn. Em cũng xin cảm ơn bạn bè, anh chị đã tận tình chỉ bảo, giúp đỡ em trong quá trình hoàn thành đề tài , giúp em hiểu thêm về những kiến thức mới.

Do kiến thức và kĩ năng còn hạn chế, phương pháp tiếp cận, phân tích, giải quyết vấn đề còn chưa được thấu đáo, chi tiết nên không tránh khỏi những thiếu sót trong quá trình nghiên cứu và trình bày. Rất kính mong được sự đóng góp ý kiến của các thầy cô giáo để đề tài được hoàn chỉnh hơn. Em xin trân trọng cảm ơn sự quan tâm giúp đỡ của các thầy, cô giảng viên trong suôt quá trình nghiên cứu và thực hiện đề tài này này.

Xin trân trọng cảm ơn!

Cần Thơ, ngày 20 tháng 6 năm 2020

Người viết

Nguyễn Nhật Khương

Danh Mục Hình

[Hình 1. Sơ đồ các bước nhận diện và tính toán 3](#_Toc43671360)

[Hình 2. Phát hiện đối tượng trong ảnh 4](#_Toc43671368)

[Hình 3. Trích lọc 3 đối tượng có diện tích lớn nhất 5](#_Toc43671370)

[Hình 4. Mô tả cấu trúc lưu trữ dữ liệu 5](#_Toc43671375)

[Hình 5. Ảnh kết quả thu được sau quá trình nhận diện 7](#_Toc43671394)

[Hình 6. Ảnh kết quả thu được sau quá trình tính toán 7](#_Toc43671397)

[Hình 7 . Sơ đồ xử lý của hàm get\_digit\_predicted. 9](#_Toc43671425)

[Hình 8 . Sơ đồ xử lý của hàm get\_operation 9](#_Toc43671431)

[Hình 9 . Sơ đồ xử lý của hàm kq 9](#_Toc43671436)

[Hình 10 . Sơ đồ xử lý hàm show 9](#_Toc43671440)

[Hình 11 Ma trận confusion trên tập test số 10](#_Toc43671450)

[Hình 12. Ma trận confusion trên tập test dấu 11](#_Toc43671451)

[Hình 13. Kết quả tính toán ảnh thực tế 11](#_Toc43671452)

PHẦN GIỚI THIỆU

1. **Đặt vấn đề**

Công nghệ thông tin ngày càng phát triển, các ứng dụng thông minh ra đời ngày càng nhiều phục vụ cho công việc, đời sống, làm cho mọi thứ ngày càng tiện lợi và nhanh chóng hơn Trong quá trình nghiên cứu và học tập, nhóm chúng em nhận thấy cần xây dựng một ứng dụng giúp người sử dụng có thể giải các bài toán mà không cần tính toán. Bên cạnh đó là áp dựng những kiến thức đã học được vào thực tiến, nhóm chúng em chọn đề tài “Xây dựng ứng dụng tính toán đơn giản” là nền tảng sơ khai ban đầu để phát triển ứng dụng tính toán thông minh sau này.

1. **Mục tiêu đề tài**

Xây dựng ứng dụng tính toán đơn giản từ 2 toán hạng và 1 toán tử. Toán hạng trong đề tài là số nguyên dương a và b (0 ≤ a,b ≤9 ). Toán hoạng trong đề tài là 2 phép toán + và -

* Đầu vào của ứng dụng là một bức ảnh chứa biểu thức dạng a + b hoặc a – b.
* Đầu ra của hệ thống là hình ảnh nhận diện các toán hạng, toán tử và kết quả phép toán.

1. **Phương pháp nghiên cứu**

Về mặt lý thuyết

- Tìm hiểu về giải thuật SVM.

- Tìm hiểu phương pháp phát hiện đối tượng trong ảnh.

- Tìm hiểu phương pháp rút trích đặc trưng HOG

- Tìm hiểu về thư viện opencv, thư viện sklearn và các thư viện có liên quan.

Về mặt giao diện

* Tìm hiểu cách xây dựng giao diện đơn giản từ Tkinter của python.

1. **Kết quả đạt được**

* Xây dựng được ứng dụng tính toán đơn giản
* Thiết kế được giao diện đơn giản dễ sử dụng.

1. **Bố cục báo cáo**

**Phần giới thiệu**

Giới thiệu tổng quát đề tài

**Phần nội dung**

**Chương 1** : Mô tả bài toán

**Chương 2** : Thiết kế, cài đặt giải thuật, trình bày các bước xây dựng hệ thống.

**Chương 3** : Kiểm thử hệ thống và đánh giá mức độ hoàn thiện, tốc độ của hệ thống.

**Phần kết luận**

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

PHẦN NỘI DUNG

CHƯƠNG 1. MÔ TẢ BÀI TOÁN

1. **Mô tả chi tiết bài toán**

Bài toán xây dựng ứng dụng tính toán từ ảnh là bài toán thuộc lĩnh vực máy học (machine – learning). Các bài toán về máy học được áp dụng rất nhiều trong các lĩnh vực như: Giáo dục, kinh tế, giải trí ….

Bài toán được đưa ra nhằm hỗ trợ cho việc tính toán nhanh chóng và đơn giản, người dùng chỉ cần tải lên hình ảnh chứa phép toán và nhân về kết quả.

1. **Vấn đề và giải pháp liên quan đến bài toán**
2. **Vấn đề của bài toán**

* Từ một bức ảnh đầu vào phải phát hiệu được các đối tượng trong ảnh.
* Vẽ bounding box bao quanh đối tượng, tách cách đối tượng từ ảnh.
* Xác định thứ tự các toán hoạng và toán tử theo thứ tự từ trái sang phải (toán hạng - toán tử - toán hạng).
* Từ những đối tựng đó tiến hành nhận diện các toán hạng và toán tử.

1. **Giải pháp giải quyết vấn đề cho bài toán**

* Phát hiện cá đối tựng trong ảnh bằng phương pháp phát hiện contours sử dụng hàm từ thư viện Opencv.
* Vẽ bounding box và tách các đối tượng sử dụng hàm từ thư viện Opencv.
* Từ tọa độ **x** của các bounding box đã vẽ, xác định thứ tự từ trái sang phải.
* Nhận diên các đối tượng bằng giải thuật SVM.

1. **Mô tả quá trình nhận diện và tính toán**

Em xin phép mô tả quá trình nhận diện và tính toán qua sơ đồ sau

Ảnh đầu vào.

Phát hiện đối tượng

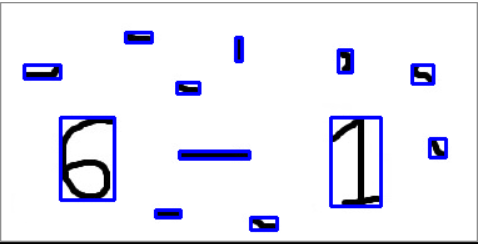
Nhận diện đối tượng

Tính toán và

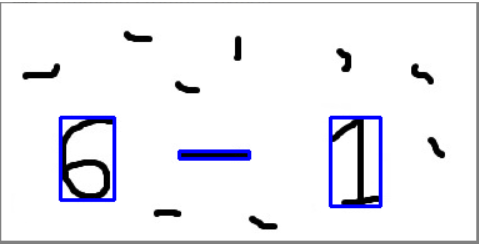
trả về kết quả

***Hình 1. Sơ đồ các bước nhận diện và tính toán***

1. **Phát hiện đối tượng trong ảnh**

* Ảnh mức xám là ảnh mà mỗi điểm ảnh (pixel) được biểu diễn bởi 1 con số đại diện cho độ sáng của điểm ảnh đó. Mỗi điểm ảnh trong ảnh mức xám được biểu thị bằng 1 giá trị a, a được biểu diễn bằng 8 bit, 0 ≤ a ≤ 255. Giá trị a càng tiến gần về 0 là ảnh càng tối, càng tiến gần về 255 là điểm ảnh đó càng sáng. Một bức ảnh có kích thước 128 x 128 tức và có 128 pixel trên mỗi hàng và có 128 hàng. Như vậy bức ảnh đó có 128\*128 giá trị a. Mở rộng thêm, máy tính mô tả 1 bức ảnh bằng một ma trận 3 chiều (h x w x d) với h là chiều cao, w là chiều rộng và d là chiều sâu của ảnh. Đối với ảnh mức xám thì d = 1 vì mỗi điểm ảnh chỉ biểu thị độ sáng tối bằng 1 giá trị duy nhất. Tương tự như vậy đối với ảnh màu RGB, mỗi điểm ảnh được biễu diễn bởi 3 giá trị tương ứng 3 màu (R – Red, G – Green, B – Blue). Vậy nên nếu tính toan trên ảnh thì ta chỉ cần tính toán 1 lần trên ảnh mức xám thay vì 3 lần trên ảnh màu.
* Giảm nhiễu bằng phương pháp Gaussian.
* Ảnh nhị phân là ảnh mà các điểm ảnh (pixel) chỉ có 2 giá trị thể hiện màu sắc là đen và trắng (tương ứng với 0 hoặc 255).
* Phát hiện đối tượng bằng phương pháp phát hiện contours trên ảnh nhị phân.
* Contours là một tập hợp các điểm liên tục tạo thành 1 đường cong kín, trong đó các điểm có thể có cùng hoặc xấp xỉ nhau về màu sắc. Contours là một phương pháp phát hiện đối tượng hiệu quả.
* ****Giai đoạn phát hiện các đối tượng trong ảnh nhằm xác định và khoanh vùng các đối tượng nghi ngờ là các toán hạng và toán tử. Vẽ bounding box bao quanh các đối tượng này, tách đối tượng ra nhằm trách nhiễu và loại bỏ những thông tin không cần thiết.

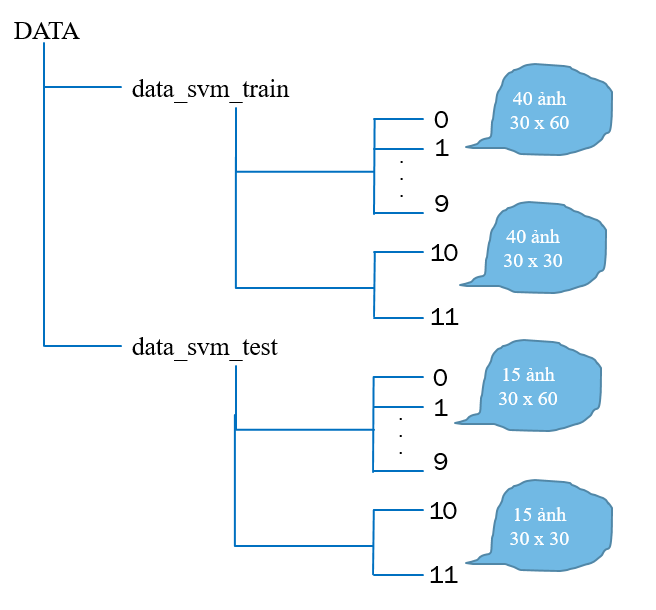
***Hình 2. Phát hiện đối tượng trong ảnh***

Từ các đối tượng đã phát hiện được, tiến hành trích lọc đối tượng để tìm ra các đối tượng có khả năng cao nhất. Ở bước này em đặt điều kiện là 3 đối tượng có diện tích lớn nhất.

***Hình 3. Trích lọc 3 đối tượng có diện tích lớn nhất***

Sắp xếp các contours theo thứ tự từ trái sang phải dựa vào tọa độ của bounding box.

1. **Nhận diện đối tượng**
2. **Xây dựng mô hình nhận diện**

* Xây dựng tập dữ liệu:

***Hình 4. Mô tả cấu trúc lưu trữ dữ liệu***

* ***Tập dữ liệu train*** *480 ảnh* **(Mỗi thư mục 40 ảnh)**
  + - **400 ảnh** các số viết tay từ 0 – 9 được lưu vào 10 thư mục tương ứng, các ảnh có kích thước **30 x 60 pixel** được vẽ bằng phần mềm paint, lưu dưới định dạng .jpg
    - **80 ảnh** các ảnh viết tay của 2 dấu + và - được lưu vào 2 thư mục là “10” và “11”, các ảnh có kích thước **30 x 30 pixel** và được lưu dưới định dạng .jpg
* ***Tập dữ liệu test*** *180 ảnh* **(Mỗi thư mục 15 ảnh)**
  + **150 ảnh** các số viết tay từ 0 – 9 được lưu vào 10 thư mục tương ứng, các ảnh có kích thước **30 x 60** pixel được tạo bằng phần mêm paint, lưu dưới định dạng .jpg
    - **30 ảnh** các ảnh viết tay của 2 dấu + và - được lưu vào 2 thư mục là “10” và “11”, các ảnh có kích thước **30 x 30 pixel** và được lưu dưới định dạng .jpg
* Phương pháp trích chọn đặc trưng HOG (Histogram of Oriented Gradient). HOG chia nhỏ bức ảnh thành các vùng con được gọi là cell, tại mỗi cell ta sẽ tính một biểu đồ về các hướng của các điểm nằm trong cell. Sau đó ghép các cell lại tạo thành bức ảnh ban đầu. (2)

Mục đích của phương pháp rút trích đặc trưng HOG là mô tả hình dạng và sự xuất hiệu của đối tượng trong ảnh.

* Xây dựng mô hình bằng giải thuật SVM

Máy học vector hỗ trợ (Support Vector Machines – SVM [Vapnik, 1995]) là một mô hình học máy hiệu quả cho các vấn đề phân lớp, hồi quy, phát hiện phần tử các biệt. Máy học SVM được áp dụng thành công trong nhiều lĩnh vực về y tế, giáo dục,… giải quyết các vấn đề về nhận dạng mặt người, phân loại văn bản,…

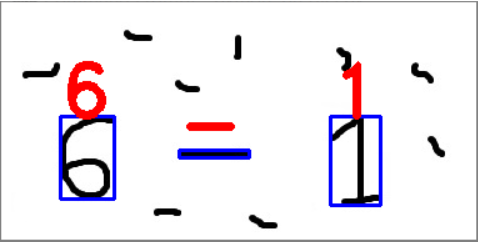
Trong bài báo cáo này, em xây dựng 2 mô hình nhận diện cho toán hạng và toán tử.

model1 để nhận diện toán hạng và model2 để nhận diện toán tử.

Cụ thể em xin trình bày ở phần thiết kế và cài đặt.

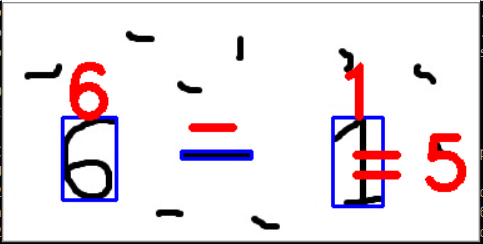
1. **Nhận diện đối tượng từ mô hình đã xây dựng**

* Từ đối tượng đã được tách và sắp xếp theo thứ tự từ phần **2.4** em nhận diện như sau:
* Có tổng cộng 3 đối tượng được sắp xếp theo thứ tự từu trái sang phải tức “toán hạng – toán tử - toán hạng”.
* Tại vị trí toán hạng áp dụng model1, tại vị trí toán tử áp dụng model2



***Hình 5. Ảnh kết quả thu được sau quá trình nhận diện***

1. **Tính toán và trả về kết quả của biểu thức.**

Kết thúc quá trình nhận diện đối tượng ở mục **2.5,** em xác định được 2 toán hạng và 1 toán tử, từ đây dễ dàng trả về kết quả của biểu thức. Chưng trình chỉ xoay quanh 2 phép toán là + và – nên các biểu thức sẽ được tính theo chiều từ trái sang phải, mỗi biểu thức chỉ tồn tại duy nhất 1 phép toán nên không có thứ tự ưu tiên.

***Hình 6. Ảnh kết quả thu được sau quá trình tính toán***

**CHƯƠNG 2****.** **THIẾT KẾ VÀ CÀI ĐẶT**

1. **Xây dựng mô hình huấn luyện bằng giải thuật SVM**

Rút trích đặc trưng HOG :

Mỗi cell được xây dựng bằng 5 x 5 pixel.

Mỗi block được xây dựng bằng 2 x 2 cell.

Số bin = 9 tương ứng 180 độ.

Giải thuật SVM các kernel: Linear, rbf, polynomial

Tham số c: 0.001, 0.01, 0.1, 1, 5, 10, 100, 1000

1. Xây dựng model1 nhận diện số ( 0 – 9 )

* Hàm nhân linear với tham số c = 0.1 (Tham số tối ưu hóa độ chính xác sau quá trình kiểm thử).
* Tập dữ liệu train cho các số bao gồm 400 tấm ảnh kích thước 30 x 60 pixel. Tại mỗi ảnh ta sẽ xây dựng được 6 x 12 cell và 5 x 11 block.
* Như vậy tại mỗi ảnh ta có 1 vector với số phần tử là: 5 x11 x 4 x 9 = 1980.
* Có tổng cộng 400 vertor tương ứng 400 ảnh.

1.2. Xây dựng model2 nhận diện dấu ( + , – )

* Hàm nhân linear với tham số c = 0.001 (Tham số tối ưu hóa độ chính xác sau quá trình kiểm thử).
* Tập dữ liệu train cho các số bao gồm 400 tấm ảnh kích thước 30 x 30 pixel. Tại mỗi ảnh ta sẽ xây dựng được 6 x 6 cell và 5 x 5 block.
* Như vậy tại mỗi ảnh ta có 1 vector với số phần tử là : 5 x 5 x 4 x 9 = 900.
* Có tổng cộng 80 vertor tương ứng 80 ảnh.

1. **Xây dựng hàm get\_digit\_predicted.**

* Đầu vào: Hình ảnh chứa phép toán.
* Đầu ra: Mảng chứa 2 toán hạng và 1 toán tử được sắp xếp theo thứ tự từ trái sang phải (toán hạng - toán tử - toán hạng).
* Xử lý:

+ Từ ảnh đầu vào tiến hành chuyển sang ảnh mức xám, giảm nhiễu bằng GausianBlur.

+ Chuyển ảnh nhị phân (Thresh\_binary\_inv) với ngưỡng 90.

+ Dùng hàm fincontours từ thư viện opencv2 tìm các contours trong ảnh.

+ Lọc 3 contours có diện tích lớn nhất, vẽ bounding box của 3 contours trên.

+ Tại mỗi contours tiến hành resize và áp dụng thuật toán giãn nở dilate.

+ Rút trích đặc trưng hog và dự đoán nhãn.

**Ảnh đầu vào**

Ảnh mức xám

Ảnh nhị phân

Contours

3 contours có diện tích lớn nhất

Resize

Dự đoán nhãn

**Tập ký tự, toạ độ các bounding box**

***Hình 7 . Sơ đồ xử lý của hàm get\_digit\_predicted.***

1. **Xây dựng hàm get\_operation**

* Đầu vào: Tập ký tự và tọa độ các boundingbox từ hàm **get\_digit\_predicted.**
* Đầu ra: Tập các toán hạng, tập phép toán, tọa độ các bounding box
* Xử lý: Xây dựng cấu trúc rẽ nhánh if – else phân biệt toán hạng và toán tử.

**Tập ký tự, tọa độ các bounding box**

Cấu trúc rẽ nhánh

**Tập toán hạng, tập toán tử, tọa độ**

***Hình 8 . Sơ đồ xử lý của hàm get\_operation***

1. **Xây dựng hàm kq (tính kết quả)**

* Đầu vào: Tập toán hạng, toán tử, tọa độ các bounding box
* Đầu ra: Kết quả phép toán = toán hạng[0] toán tử [0] toán hạng[1]

Trong đó toán hạng[0] và toán hạng[1] là 2 số nguyên dương ≥ 0 và ≤ 9, toán tử [0] là phép toán + hoặc -

**Tập ký tự, tọa độ các bounding box**

Cấu trúc rẽ nhánh xác định phép toán

**Kết quả phép toán, tọa độ các bounding box**

# *Hình 9 . Sơ đồ xử lý của hàm kq*

1. **Xây dựng hàm show**

* Đầu vào: Kết quả, toán hạng, phép toán, tọa độ các bounding box.
* Đầu ra: Hình ảnh được vẽ các bounding box và kết quả phép toán.

**Kết quả phép toán, toán hạng, toán tử, tọa độ các bounding box**

Xác định tọa độ các bounding box

**Hình ảnh kết quả sau khi đã vẽ các bounding box**

# *Hình 10 . Sơ đồ xử lý hàm show*

**CHƯƠNG 3.** **KIỂM TRA VÀ ĐÁNH GIÁ**

1. **Mục tiêu**

* Kiểm tra độ chính xác của model1 và model2.
* Xem xét khả năng nhầm lẫn giữa các số.
* Từ kết quả kiểm tra tối ưu hóa model, khắc phục hạn chế và phát triển.

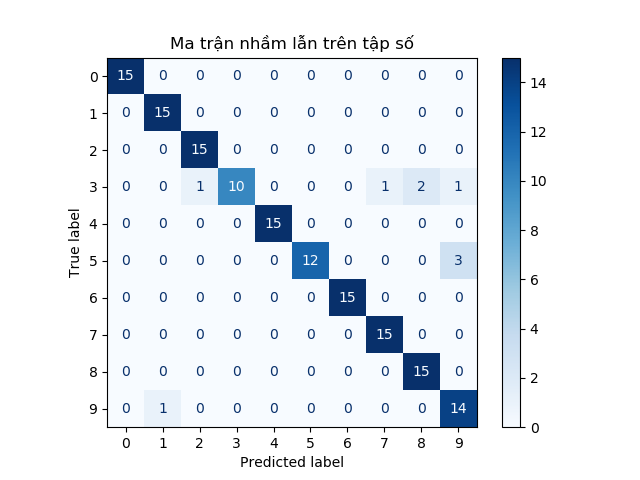
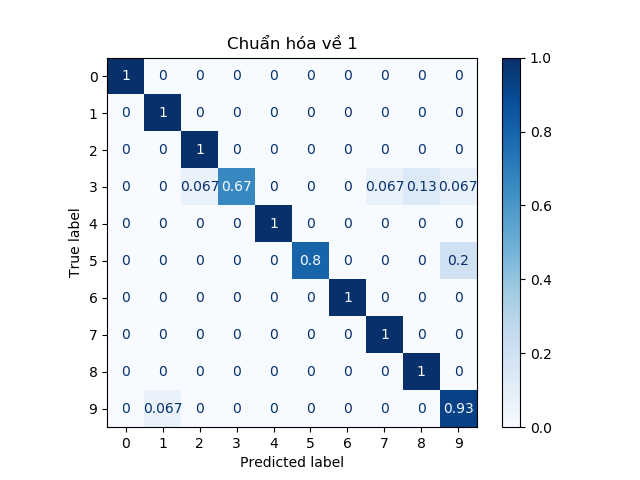
1. **Phương thức kiểm tra**

**Accuracy (độ chính xác)** thể hiện tỉ lệ các bức ảnh dự đoán đúng trên tổng số các bức ảnh trên tập test.

**Confusion matrix** thể hiện cụ thể giá trị dự đoán đúng và sai của từng class. Confusion matrix cho biết có bao nhiêu giá trị được dự đoán đúng vào lớp của nó, bao nhiêu giá trị dự đoán sai và cụ thể là dự đoán sai vào class nào.

1. **Kết quả đạt được**
   1. Kết quả kiểm tra model1

**Độ chính xác**: 94%

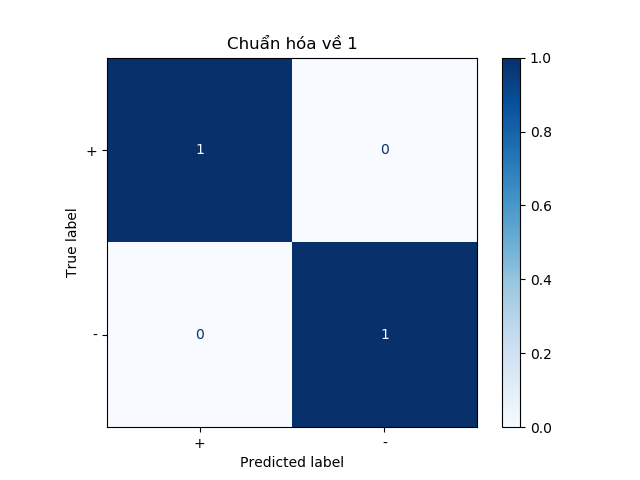
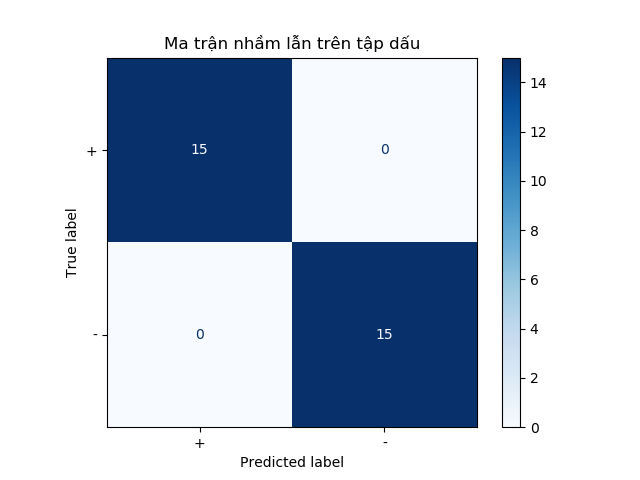
**Confusion matrix**

# *Hình 11 Ma trận confusion trên tập test số*

* 1. Kết quả kiểm tra model2

**Độ chính xác**: 100%

**Confusion matrix**

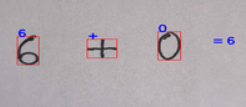
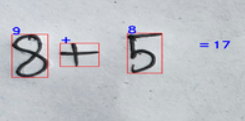
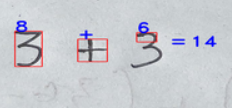
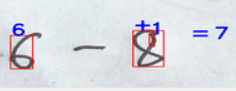
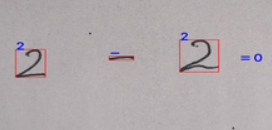
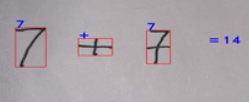
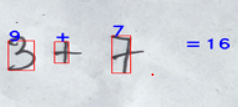
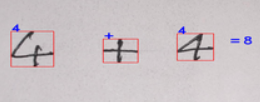
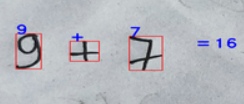
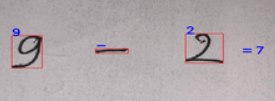


***Hình 12. Ma trận confusion trên tập test dấu***

* 1. Kiểm tra kết quả tính toán thực tế.

Kết quả kiểm tra trên 12 ảnh thực tế chụp bằng camera trên điện thoại di động:

***Hình 13. Kết quả tính toán ảnh thực tế***



*Nhận xét*: Có 8 ảnh dự đoán đúng trên 12 ảnh.

Độ chính xác: 66.66 %.

Trong 4 ảnh dự đoán sai:

* 3 ảnh phát hiện sai contours
* 1 ảnh nhận diện sai số.

PHẦN KẾT LUẬN

* 1. **Kết quả đạt được**
* Xây dựng được chương trình tính toán với giao diện đơn giản.
* Nhận diện được các số và dấu trên ảnh đầu vào, dựa vào đó tính toán được kết quả.
  1. Hạn chế
* Hạn chế về nhận diện: Trong mỗi bức ảnh chỉ trích chọn 3 contours với diện tích lớn nhất.
* Hạn chế về độ phức tạp của phép toán: Chương trình chỉ tính toán kết quả cho phép toán có 2 toán hạng đơn vị và 1 phép toán (+, -) .
  1. **Hướng phát triển**
* Tăng khả năng xác định chính xác contours đề nhận diện chính xác ký tự.
* Cải tiến chương trình cho phép tính toán các phép toán phức tạp hơn.
* Mở rộng giới hạn số lượng toán hạng và toán tử.
* Mở rộng khả năng nhận diện các toán hạng như: Các số nhiều hơn 1 chữ số, số âm, số thập phân…
* Mở rộng khả năng nhận diện các toán hạng: Phép nhân, chia, bình phương, lấy căn….

TÀI LIỆU THAM KHẢO

1. Recommender Systems: Types of Filtering Techniques

International Journal of Engineering Research & Technology (IJERT)

1. Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs