Trích chọn đặc trưng GIST kết hợp với máy học VÉC-TƠ HỖ TRỢ để nhận dạng ký tự số viết tay

Phạm Viết Đạo1, Đỗ Thanh Nghị1 và Phạm Nguyên Khang1

1 Khoa CNTT&TT, Trường Đại học Cần Thơ

| ***Thông tin chung:****Ngày nhận: 31/08/2015**Ngày chấp nhận: 26/09/2015****Title:***Handwritten digit recognition using Gist feature and support vector machines***Từ khóa:*** Chữ số viết tay, đặc trưng GIST, máy học véc-tơ hỗ trợ***Keywords:***Handwritten digit recognition, GIST feature, support vector machines | ABSTRACTThis paper proposes a model of handwritten digits recognition using the GIST features and support vector machines (SVM). The GIST approach to extract a set of perceptual dimensions that represent the spatial structure of handwriting digits. The number of features obtained by the GIST method could be large. Thus, we propose to use the SVM algorithm that is usually suited for classifying very-high-dimensional datasets, typically the handwritten digit image representation based on the GIST method. The experimental results on USPS, MNIST datasets show that our proposal has very high accuracy compared to state-of-the-art algorithms.TÓM TẮTBài báo này đề xuất một mô hình nhận dạng ký tự số viết tay đạt được độ chính xác cao dựa trên phương pháp trích chọn đặc trưng GIST kết hợp với máy học véc tơ hỗ trợ (SVM) để nhận dạng. Phương pháp GIST rút trích từ ảnh tập hợp các đặc trưng quan trọng như tính tự nhiên, mở rộng, độ nhám, độ chắc chắn, cho phép trình bày cấu trúc không gian của một cảnh. Mỗi ảnh ký tự số được rút trích các đặc trưng GIST, biểu diễn bởi véc tơ có 960 chiều. Chúng tôi đề xuất sử dụng giải thuật học tự động có khả năng làm việc tốt trên dữ liệu có số chiều lớn sinh ra từ bước tiền xử lý, chẳng hạn như máy học SVM, mạng nơ-ron, Adaboost, rừng ngẫu nhiên. Kết quả thực nghiệm trên . Kết quả thử nghiệm trên tập dữ liệu thực USPS và MNIST cho thấy phương pháp chúng tôi đề xuất đạt độ chính xác cao khi so sánh với các phương pháp hiện có. |
| --- | --- |
|  |

# GIỚI THIỆU

Nhận dạng chữ là một lĩnh vực đã được quan tâm, nghiên cứu từ nhiều năm nay. Trong những năm gần đây, do sự thúc đẩy của quá trình tin học hoá, bài toán nhận dạng chữ đã được chú trọng phát triển và có nhiều ứng dụng vào thực tế. Nhận dạng chữ được áp dụng vào quá trình tự động hoá các công việc văn phòng phục vụ cho việc nhập liệu, lưu trữ văn bản, đọc và xử lý các chứng từ, hoá đơn, phiếu ghi, bảng điểm,… với mục đích giúp tăng tốc độ và chất lượng nhập thông tin vào máy tính. Nhận dạng chữ được chia ra làm hai hướng chính là nhận dạng chữ in và nhận dạng chữ viết tay. Đối với bài toán nhận dạng chữ in đã được giải quyết gần như trọn vẹn với các sản phẩm tiêu biểu như: FineReader 9.0, OmniPage, VNDOCR 4.0. Bài toán nhận dạng chữ viết tay hiện nay vẫn còn là vấn đề thách thức lớn đối với các nhà nghiên cứu. Chúng ta chưa tìm được giải pháp đủ tốt để giải quyết hết những khó khăn của bài toán. Những khó khăn thường gặp như:

- Kích thước chữ viết tay không đồng đều: lúc cao, lúc thấp, lúc to, lúc nhỏ,...

- Mỗi người viết theo mỗi kiểu khác nhau: người viết chữ nghiêng, người viết chữ đứng,…

- Các ký tự nhiều khi viết thiếu nét, thừa nét thậm chí dính vào nhau.

Do những khó khăn trên, nên khi giải quyết bài toán nhận dạng chữ viết tay chúng ta buộc phải giới hạn bài toán trong một phạm vi cụ thể đối với mẫu chữ nhận dạng. Bài toán chúng tôi đặt ra ở đây là ứng dụng đặc trưng GIST vào nhận dạng ký tự viết số tay dựa trên hai tập dữ liệu mẫu là MNIST (National Institute of Standard and Technology of the United States) và USPS (United States Postal Service).

Trích chọn đặc trưng đóng vai trò rất quan trọng trong một hệ thống nhận dạng. Có nhiều phương pháp trích chọn đặc trưng hiệu quả có thể áp dụng cho bài toán nhận dạng chữ viết tay như: Trọng số vùng (Zoning), biểu đồ chiếu (Projection histograms), trích chọn theo chu tuyến (Contour profiles), đặc trưng wavelet Haar,… Trong bài báo này, chúng tôi sử dụng ý tưởng của phương pháp trích chọn đặc trưng GIST cho mô hình nhận dạng ký tự viết tay. Khác với đặc trưng SIFT [4], GIST là một đặc trưng toàn cục biểu diễn nội dung ảnh được Oliva & Torralba [1][9][12] đề xuất năm 2001. Đặc trưng GIST thể hiện dưới dạng một véctơ và mỗi véctơ này được tính toán từ kết quả của việc áp dụng các bộ lọc Gabor lên ảnh. Từ dữ liệu ảnh đầu vào, sau khi trích chọn đặc trưng GIST sẽ cho ra một véctơ có 960 chiều. Đặc trưng GIST rút trích từ ảnh tập hợp các đặc trưng quan trọng như: tính tự nhiên (Degree of Naturalness), tính cởi mở (Degree of Openness), tính mở rộng (Degree of Expansion), độ nhám (Degree of Roughness) và độ chắc chắn (Degree of Ruggedness) cho phép trình bày cấu trúc không gian của một cảnh. Mặt khác, đặc trưng GIST được trích chọn theo các hướng với các tần số khác nhau từ đó làm nổi bật thêm cấu trúc của ảnh.

SVM (Support Vector Machines) là phương pháp máy học tiên tiến đã và đang được áp dụng rộng rãi trong lĩnh vực khai phá dữ liệu và thị giác máy tính. Phương pháp SVM được Vapnik [17] đề xuất năm 1995, dựa trên lý thuyết học thống kê nên có một nền tảng toán học chặt chẽ đảm bảo kết quả đạt được là tối ưu. Vì vậy, chúng tôi đề xuất phương pháp phân lớp SVM cho bài toán nhận dạng ký tự viết tay. Trong thực nghiệm, chúng tôi cũng kết hợp GIST với Adaboost [5], GIST với rừng ngẫu nhiên (Random Forest) [7] và GIST với mạng nơ-ron (Neuron Network) [2][8], sau đó so sánh kết quả.

Phần còn lại của bài báo này có cấu trúc như sau: Phần 2 phác hoạ mô hình, phương pháp trích chọn đặc trưng GIST từ ảnh. Phần 3 tóm tắt những ý tưởng cơ bản của phương pháp phân lớp SVM. Phần 4 là kết quả thực nghiệm trên tập dữ liệu mẫu MNIST và USPS. Cuối cùng phần 5 là phần kết luận.

# TRÍCH CHỌN ĐẶC TRƯNG GIST

## Ý tưởng của phương pháp trích chọn đặc trưng GIST

Ảnh đầu vào sau khi được tiền xử lý sẽ được tách ra thành 3 kênh màu Red-Green-Blue riêng biệt. Sau đó, áp dụng phép biến đổi Fourier [8] trên mỗi kênh màu. Ứng với mỗi ảnh Fourier áp dụng lần lượt 20 bộ lọc Gabor lên ảnh. Cuối cùng, kết quả của mỗi bộ lọc được đưa qua phép biến đổi Fourier ngược, sau đó chia thành 16 vùng bằng nhau và trích đặc trưng. Kết quả của mỗi vùng là một đặc trưng. Như vậy, số chiều của véc-tơ đặc trưng GIST là: 3\*(8+8+4)\*16 = 960 chiều.



Hình 1: Mô hình trích đặc trưng GIST.

## Áp dụng phương pháp trích chọn đặc trưng GIST vào chữ viết tay

### Ảnh đầu vào

Ảnh dùng để trích chọn đặc trưng GIST thường là ảnh màu hoặc ảnh mức xám.

### Tiền xử lý

*a. Chuẩn hoá kích thước*

Đặc trưng GIST chủ yếu được thực hiện trong miền tần số thông qua phép biến đổi Fourier. Do đó, ảnh đầu vào sẽ được chuẩn hoá kích thước về ảnh vuông theo dạng 2n×2n (n ∈ N). Chúng tôi sử dụng phương pháp nội suy Bi-cubic [3] để chuẩn hoá kích thước ảnh về 32×32.

****

Hình 2: Sơ đồ áp dụng đặc trưng GIST vào chữ viết tay



Hình 3: Chuẩn hoá kích thước ảnh.

*b. Tách ba kênh màu*

Mỗi điểm ảnh thường được cấu thành từ ba thành phần màu cơ bản: Red, Green, Blue. Nếu sử dụng 8 bit biểu diễn giá trị màu sắc cho ảnh, thì giá trị màu của mỗi điểm ảnh biến thiên trong đoạn [0, 28-1].



Hình 4: Tách ba kênh màu RGB

*c. Biến đổi Log*

Mục đích của phép biến đổi Log [10] là để mở rộng các giá trị của các điểm ảnh tối trong một ảnh trong khi nén các giá trị cấp xám cao lại.

Mục đích của phép biến đổi Log [10] là để mở rộng các giá trị của các điểm ảnh tối trong một ảnh trong khi nén các giá trị cấp xám cao lại.

$$g(x,y)=c\*Log(1+f(x,y))$$

Với:

* $g(x,y)$: giá trị cường độ sáng thu được sau phép biến đổi $Log$.
* $c$: hằng số ($c$ = 1).
* $f\left(x,y\right)$: giá trị cường độ sáng của ảnh đầu vào tại toạ độ $\left(x,y\right)$.



Hình 5: Áp dụng phép biến đổi Log

*d. Add padding (AP)*

Để giảm hiệu ứng biên [13] của ảnh khi lọc trong miền Fourier, ta đệm thêm 5 pixel vào bốn cạnh của ảnh, sau đó lấy đối xứng giá trị.



Hình 6: Add padding

*e. Lọc trắng ảnh (whitening filter - WF)*

Mục đích của lọc trắng ảnh là để cân bằng năng lượng phổ (power spectrum). Để thực hiện lọc trắng, trước hết ta dùng phép biến đổi Fourier thuận (DFT) để chuyển ảnh Red, Green và Blue từ miền không gian fR,G,B(x,y) sang miền tần số FR,G,B(u,v). Sau đó xây dựng bộ lọc trắng như sau:

$$Tần số theo hướng X: f\_{X}\left(x,y\right)=m-\frac{N}{2}$$

$$Tần số theo hướng Y: f\_{Y}\left(x,y\right)=n-\frac{N}{2}$$

$$Kernel:h\_{w}\left(f\_{X},f\_{Y}\right)=e^{\frac{-(f\_{X}\left(x,y\right)^{2}+f\_{Y}\left(x,y\right)^{2})}{s^{2}}}$$

$$Với s=\frac{4}{\sqrt{Log(2)}} (4 cycles/image)$$

$$Bộ lọc trắng:$$

$$H\_{W}\left(u,v\right)=Shift\left[h\_{w}\left(f\_{X},f\_{Y}\right)\right]$$

Với: N kích thước ảnh, 0 ≤ m, n, x, y < N.

Áp dụng lọc trắng ảnh

$$G\_{R,G,B}\left(u,v\right)=H\_{W}\left(u,v\right)\*F\_{R,G,B}\left(u,v\right)$$

Biến đổi Fourier ngược (IDFT)

$$g\_{R,G,B}\left(x,y\right)=IDFT\left[G\_{R,G,B}\left(u,v\right)\right]$$



Hình 7. Lọc trắng ảnh

*f. Lọc chuẩn hoá độ tương phản (contrast normalisation filter - CNF)*

Sau khi lọc trắng ảnh, năng lượng phổ (power spectrum) của ảnh bây giờ khá bằng phẳng. Do đó, ta chỉ cần lọc thông thấp (lowpass) để năng lượng phổ trở nên cân bằng hơn. Để làm điều này, ta tiến hành chuẩn hoá độ tương phản cục bộ (local contrast normalisation). Bộ lọc chuẩn hoá độ tương phản được xây dựng như sau:

$$V\_{R,G,B}(x,y)=f\_{R,G,B}\left(x,y\right)-g\_{R,G,B}(x,y)$$

Trong đó:

* $f\_{R,G,B}\left(x,y\right)$: giá trị cường độ sáng của ảnh Red, Green, Blue tại toạ độ (x, y) sau khi Add padding.
* $g\_{R,G,B}(x,y)$: giá trị cường độ sáng của ảnh Red, Green, Blue tại toạ độ (x, y) sau khi lọc trắng ảnh.

$$Kernel:h\_{c}\left(x,y\right)=\left[\frac{\sum\_{}^{}V\_{R,G,B}\left(x,y\right)}{3}\right]^{2}$$

$$Bộ lọc chuẩn hoá$$

$$H\_{C}\left(u,v\right)=DFT\left[h\_{c}\left(x,y\right)\right]$$

$$Áp dụng lọc chuẩn hoá$$

$$G\_{C}\left(u,v\right)=H\_{C}\left(u,v\right)\*H\_{W}\left(u,v\right)$$

Biến đổi Fourier ngược (IDFT)

$$g\_{c}\left(x,y\right)=IDFT\left[G\_{C}\left(u,v\right)\right]$$

Xây dựng lại ảnh sau khi chuẩn hoá

$$f\_{R,G,B}\left(x,y\right)= \frac{V\_{R,G,B}(x,y)}{0.2+\sqrt[4]{g\_{c\\_real}\left(x,y\right)^{2}+ g\_{c\\_img }\left(x,y\right)^{2}}}$$



Hình 8: Lọc chuẩn hoá độ tương phản

*g. Remove padding (RP)*

Việc remove padding chỉ đơn giản là cắt bỏ 5 pixel ở bốn cạnh của ảnh Red, Green, Blue mà ta đã thêm vào để cho ảnh trở về đúng kích thước ban đầu (kích thước sau khi chuẩn hoá 32×32).



Hình 9: Xoá padding

## Sinh tập bộ lọc Gabor

Lọc Gabor [6][16] được sử dụng rộng rãi trong phân tích dữ liệu và nó tỏ ra rất hiệu quả trong việc phân tích kết cấu ảnh theo các hướng và các tỉ lệ khác nhau. Trong trích đặc trưng GIST sử dụng 20 bộ lọc Gabor. Tập các bộ lọc Gabor được sinh ra ở 3 dãy (scale) và 8 hướng. Trong đó, dãy 1 và dãy 2 gồm 8 bộ lọc, dãy 3 gồm 4 bộ lọc.

$$G\left(u, v\right)=exp\left[-10\*0.35\left(\frac{f\_{r}(u,v)}{Nf\_{s}}-1\right)^{2}-2πpδ\_{θ}^{2}\right]$$

Trong đó:

* $f\_{r}(u,v)$: giá trị chuẩn hóa bán kính tần số (Normalised radius frequency)

$$f\_{r}\left(u,v\right)=Shift(\sqrt{f\_{X}(x,y)^{2}+f\_{Y}(x,y)^{2}})$$

* $N$: kích thước của bộ lọc (32×32).
* $f\_{s}$: tần số ứng với từng scale:

$f\_{s}=k^{-s}f\_{max} $, $(s=0, …, S-1)$

$k$ : hằng số $k=1.85$

$s$ : scale thứ $s$, $S$ = 3 🠖 $s$ = 0, 1, 2

$f\_{max}$ : tần số cực đại $f\_{max}=0.3$

* $p$: băng thông của bộ lọc:

$$ p=\frac{16 \* O\_{s}^{2}}{N^{2}}$$

$ O\_{s}:$ số hướng ứng với từng scale s

* $δ\_{θ}$ là các hướng của bộ lọc:

$$θ=π\frac{n}{O} ,(n=0,…,O-1)$$



Hình 10: Bộ lọc Gabor

## Trích đặc trưng GIST

### Áp dụng lọc Gabor

Trước tiên thực hiện phép biến đổi Fourier thuận (DFT) trên từng ảnh Red, Green, Blue đã được tiền xử lý để chuyển ảnh về miền tần số.

$$F\_{R,G,B}\left(u,v\right)=DFT\left[f\_{R,G,B}\left(x,y\right)\right]$$

Sau đó, áp dụng lần lượt 20 bộ lọc Gabor lên từng ảnh màu bằng cách nhân giá trị của bộ lọc Gabor với giá trị phức (phần thực và phần ảo) của từng ảnh ở vị trí (u, v) tương ứng.

$$G\_{R,G,B}\left(u,v\right)=Gabor\_{1..20}\left(u,v\right)\*F\_{R,G,B}\left(u,v\right)$$

Cuối cùng, thực hiện phép biến đổi Fourier ngược (IDFT) trên từng ảnh để chuyển về miền không gian, xây dựng lại ảnh và trích đặc trưng.

$$g\_{R,G,B}\left(x,y\right)=IDFT\left[G\_{R,G,B}\left(u,v\right)\right]$$

$$f\_{R,G,B}\left(x,y\right)=\sqrt{g\_{R,G,B\\_real}\left(x,y\right)^{2}+g\_{R,G,B\\_imaginary }\left(x,y\right)^{2}}$$

****

Hình 11: Áp dụng lọc Gabor

### Chia vùng, tính đặc trưng

Ảnh sau khi qua các bộ lọc Gabor và phép biến đổi Fourier ngược sẽ tiến hành trích đặc trưng. Để trích đặc trưng, trước hết ta chia ảnh thành 16 vùng riêng biệt bằng nhau. Sau đó tính giá trị trên mỗi vùng bằng cách lấy tổng giá trị của các điểm ảnh trên mỗi vùng chia cho số điểm ảnh của vùng. Ta làm lần lượt cho 20 ảnh Red, 20 ảnh Green và 20 ảnh Blue. Cuối cùng, ta kết hợp các giá trị này lại thành một đặc trưng có 960 chiều.



Hình 12: Chia vùng, tính đặc trưng

# Mô hình máy học véc-tơ hỗ trợ

Mô hình máy học véc-tơ hỗ trợ (Support vector machines hay viết tắt là SVM) là một mô hình học tự động do (Vapnik et al., 1995) đề xuất và phát triển. Xét bài toán phân lớp tuyến tính nhị phân. Cho tập huấn luyện gồm m phần tử: $\left\{x^{\left(1\right)}, x^{\left(2\right)}, …, x^{\left(m\right)}\right\}$, mỗi phần tử là một véc-tơ trong không gian n chiều. Mỗi phần tử thuộc về một trong hai lớp (+1: lớp dương, -1: lớp âm) như hình 13. Nhiệm vụ của bài toán phân lớp tuyến tính là tìm một siêu phẳng tách rời hai lớp sao cho các phần thuộc cùng lớp nằm về một phía của siêu phẳng. Cũng cùng một mục tiêu đó, mô hình SVM cho bài toán phân lớp tuyến tính nhị phân cũng tìm một siêu phẳng tách rời hai lớp dữ liệu. Tuy nhiên, để tăng cường khả năng tổng quát hoá, mô hình SVM cố gắng tìm một siêu phẳng tối ưu trong tất cả các siêu phẳng có khả năng tách rời tập dữ liệu. Siêu phẳng tối ưu theo mô hình SVM là siêu phẳng mà khoảng cách từ nó đến phần tử gần nó nhất là lớn nhất. Để tìm được siêu phẳng tối ưu, ta định nghĩa hai siêu phẳng hỗ trợ song song nhau: một cho lớp dương (d+) và một cho lớp âm (d-). Không giảm tính tổng quát, ta luôn có thể giả sử tất cả các phần tử thuộc lớp âm nằm về bên trái của siêu phẳng d- và tất cả các phần tử thuộc lớp dương nằm về phía bên phải của siêu phẳng d+. Ta sẽ tìm đồng thời d+ và d- sao cho khoảng cách giữ chúng là lớn nhất. Khoảng cách giữa hai siêu phẳng được gọi là *lề* (margin):

$$margin=\frac{1}{\left‖w\right‖}$$



Hình 13: Ví dụ phân lớp SVM trong R2

Bài toán tối ưu của SVM chính là bài toán quy hoạch toàn phương:

$$\min\_{w,b}\frac{1}{2}\left‖w\right‖^{2}$$

với ràng buộc:

$$y^{\left(i\right)}\left(w^{T}x^{\left(i\right)}+b\right)\geq 1$$

trong đó $y^{\left(i\right)}$ là nhãn hay lớp của phần tử i.

Giải bài toán tối ưu này ta thu được w và. Để dự báo nhãn của một phần tử mới x, ta xét dấu của $w^{T}x^{\left(i\right)}+b$ hay:

$$predict\left(x\right)= sign\left(w^{T}x^{\left(i\right)}+b\right)$$

Trường hợp, dữ liệu không khả tách tuyến tính (ta không thể nào tách rời dữ liệu bằng một siêu phẳng mà không có phần tử nằm sai phía), mô hình SVM có thể mở rộng bằng cách thêm vào mô hình biến lỗi (ta xem khoảng cách từ các phần tử nằm sai phía so với siêu phẳng hỗ trợ của chúng như là lỗi). Bài toán tối ưu đối với SVM bây giờ trở thành bài toán tối ưu 2 mục tiêu: lề lớn nhất và lỗi nhỏ nhất. Ta có thể kết hợp 2 mục tiêu lại thành một tiêu duy nhất nhờ vào tham số điều chỉnh sự kết hợp này. Bài toán tối ưu của SVM trong trường hợp này sẽ là:

$$\min\_{w}\frac{1}{2}\left‖w\right‖^{2}+c\sum\_{i=1}^{m}z^{\left(i\right)}$$

với ràng buộc:

$$y^{\left(i\right)}\left(w^{T}x^{\left(i\right)}+b\right)+z^{\left(i\right)}\geq 1$$

$$z^{\left(i\right)}\geq 0$$

trong đó $z^{\left(i\right)}$ là biến lỗi được định nghĩa như là khoảng cách từ phần tử nằm sai phía đến siêu phẳng hỗ trợ của nó và c là hằng số điều chỉnh độ rộng của lề và lỗi. Bài toán đối ngẫu của nó:

$$\min\_{α}\frac{1}{2}\sum\_{i=1}^{m}\sum\_{j=1}^{m}α\_{i}α\_{j}y^{\left(i\right)}y^{\left(j\right)}x^{\left(i\right)}x^{\left(j\right)}-\sum\_{i=1}^{m}α\_{i}$$

với ràng buộc:

$$\sum\_{i=1}^{m}α\_{i}y^{\left(i\right)}=0$$

$$0\leq α\_{i}\leq c$$

Mô hình SVM cũng có thể được mở rộng để xử lý bài toán phân lớp phi tuyến sử dụng kỹ thuật hàm nhân (kernel function) trên bài toán đối ngẫu. Ta thay tích vô hướng của hai phần tử $x^{\left(i\right)}x^{\left(j\right)}$ bằng hàm nhân $K\left(x^{\left(i\right)}, x^{\left(j\right)}\right)$. Một số hàm nhân thông dụng được cho trong bảng 1.

**Bảng 1. Một số hàm nhân thường dùng**

|  |  |
| --- | --- |
| **Kiểu hàm nhân** | **Công thức** |
| Linear kernel | K(x,y) = x.y |
| Polynomial | K(x,y) = (x.y + 1)d |
| Radial basis function (Gaussian) kernel | K(x,y) = $e^{\frac{-\left|x-y\right|^{2}}{2σ^{2}}}$ |
| Hyperbolic tangent kernel | K(x,y) = tanh(a.x.y - b) |

Mô hình SVM được cho là một phương pháp tổng quát cho các bài toán của học máy bao gồm: phân lớp, hồi quy và ước lượng mật độ xác suất. Nếu xét riêng về khả năng giải bài toán phân lớp, SVM có tính tổng quát hoá cao (nhờ vào lề lớn) vì thế hiệu quả phân lớp luôn bằng hoặc cao hơn các phương pháp phân lớp khác.

# KẾT QUẢ THỰC NGHIỆM

Như đã giới thiệu, chúng tôi thực nghiệm trên hai tập dữ liệu mẫu là MNIST và USPS. MNIST [14] là tập dữ liệu do Yann Lecun đề xuất vào năm 1998. Tập dữ liệu này là chữ số viết tay gồm: 60.000 mẫu dùng để huấn luyện và 10.000 mẫu dùng để kiểm tra. Mỗi mẫu là một ảnh đa cấp xám kích thước 28×28. USPS [15] cũng là tập dữ liệu chữ số viết tay gồm: 7291 mẫu dùng để huấn luyện và 2007 mẫu dùng để kiểm tra. Mỗi mẫu là một ảnh đa cấp xám kích thước 16×16.

**Bảng 2: Sự phân bố các số của tập MNIST và USPS**

|  |  |  |
| --- | --- | --- |
| **Nhãn** | **MNIST** | **USPS** |
| Số mẫu huấn luyện | Số mẫu kiểm tra | Số mẫu huấn luyện | Số mẫu kiểm tra |
| **0** | 5293 | 980 | 1194 | 359 |
| **1** | 6742 | 1135 | 1005 | 264 |
| **2** | 5958 | 1032 | 731 | 198 |
| **3** | 6131 | 1010 | 658 | 166 |
| **4** | 5842 | 982 | 652 | 200 |
| **5** | 5421 | 892 | 556 | 160 |
| **6** | 5918 | 958 | 664 | 170 |
| **7** | 6265 | 1028 | 645 | 147 |
| **8** | 5851 | 974 | 542 | 166 |
| **9** | 5949 | 1009 | 644 | 177 |
| **Tổng số** | **60000** | **10000** | **7291** | **2007** |

  ****

MNIST USPS

**Hình 15: Các mẫu trích từ tập dữ liệu MNIST và USPS**

Chúng tôi thực hiện trích chọn đặc trưng GIST trên hai tập dữ liệu MNIST và USPS, sau đó sử dụng lần lượt các máy học: SVM, Adaboost, rừng ngẫu nhiên và mạng nơ-ron để tiến hành phân lớp và ghi nhận kết quả.

**Bảng 3. Kết quả phân lớp của các máy học trên tập dữ liệu MNIST và USPS**

|  |  |  |
| --- | --- | --- |
| **Máy học** | **Tập dữ liệu MNIST** | **Tập dữ liệu USPS** |
| **SVM (nhân Poly)** | **98.99 %** | **96.26 %** |
| **SVM (nhân RBF)** | **99.24 %** | **96.11 %** |
| **Adaboost + J48** | 98.37 % | 94.52% |
| **Rừng ngẫu nhiên** | 97.97 % | 94.72 % |
| **Mạng nơ-ron** | 97.59 % | 96.11 % |

Tham số các máy học trong quá trình thực nghiệm được thiết lập như sau:

* Tập dữ liệu MNIST:
* SVM tuyến tính (nhân Poly): tham số d = 1.
* SVM phi tuyến (nhân RBF): tham số Gamma γ = 0.05.
* Adaboost + J48: Số bước lặp I = 50.
* Rừng ngẫu nhiên: Số cây được xây dựng T = 100, số thuộc tính để tính toán phân hoạch K = 100.
* Mạng nơ-ron: Hệ số học L = 0.05, số tầng ẩn H = 10.
* Tập dữ liệu USPS:
* SVM tuyến tính (nhân Poly): tham số d = 1.
* SVM phi tuyến (nhân RBF): tham số Gamma γ = 0.05.
* Adaboost + J48: Số bước lặp I = 30.
* Rừng ngẫu nhiên: Số cây được xây dựng T = 100, số thuộc tính để tính toán phân hoạch K = 15.
* Mạng nơ-ron: Hệ số học L = 0.3, số tầng ẩn H = 485.

Chúng tôi cũng thực hiện so sánh kết quả phân lớp của đặc trưng GIST với các đặc trưng: Ma trận nhị phân, Zone 2×2, Projection, Contour Profile, Haar wavelet mà tác giả Phạm Anh Phương [8] đã thực hiện trên cùng tập dữ liệu MNIST sử dụng mô hình máy học SVM đa lớp theo chiến lược OVO với hàm nhân RBF (σ = 0.05). Kết quả cho thấy đặc trưng GIST mà chúng tôi đề xuất cho kết quả tốt hơn.

**Bảng 4. Kết quả phân lớp của các đặc trưng trên tập dữ liệu MNIST sử dụng máy học SVM**

|  |  |  |
| --- | --- | --- |
| **Đặc trưng** | **Số chiều** | **Độ chính xác** |
| **Ma trận nhị phân** | 256 | 97.2% |
| **Zone 2×2** | 64 | 97.9% |
| **Projection** | 94 | 97.0% |
| **Contour Profile** | 64 | 95.5% |
| **Haar wavelet** | 256 | 97.8% |
| **GIST** | **960** | **99.24 %** |

Các kết quả thực nghiệm ở Bảng 3 cho thấy mô hình nhận dạng ký tự viết tay dựa trên phương pháp trích chọn đặc trưng GIST kết hợp với máy học SVM cho độ chính xác cao và đáng tin cậy.

**5. KẾT LUẬN**

Bài báo đã đề xuất một mô hình nhận dạng ký tự số viết tay dựa trên phương pháp trích chọn đặc trưng GIST kết hợp với máy học SVM để nhận dạng. Chúng tôi tiến hành cài đặt phương pháp trích chọn đặc trưng GIST, sau đó áp dụng phương pháp trích chọn đặc trưng này để trích đặc trưng ảnh của hai tập dữ liệu MNIST và USPS. Chúng tôi sử dụng SVM để nhận dạng. Các kết quả thực nghiệm cho thấy, khi kết hợp đặc trưng GIST với các máy học khác nhau đều cho độ chính xác cao. Đặc biệt, khi kết hợp đặc trưng GIST với máy học SVM thì cho độ chính xác tốt nhất. Như vậy, GIST là một đặc trưng tốt và hoàn toàn có thể áp dụng được vào nhận dạng chữ viết tay.

**TÀI LIỆU THAM KHẢO**

1. A.Oliva and A.Torraalba (2001). *Modeling the Shape of the Scene: A Holistic Representationof the Spatial Envelope*. International Journal of Computer Vision 42(3), 145–175.
2. CHRISTOPHER M. BISHOP (1995). *Neural Networks for Pattern Recognition*. CLARENDON PRESS • OXFORD, UK.
3. Cubic interpolation. [*http://www.paulinternet.nl/?page=bicubic*](http://www.paulinternet.nl/?page=bicubic). *(ngày truy cập 25/06/2013)*
4. David G. Lowe (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. Computer Science Department University of British ColumbiaVancouver, B.C., Canad.
5. Freund Y, Schapire R (1995). A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory. pp. 23-37.
6. Joni-Kristian Kamarainen. *Gabor Features in Image Analysis.*, Machine Vision and Pattern Recognition Laboratory, Lappeenranta University of Technology (LUT Kouvola).
7. L. Breiman. Random forests (2001). *Machine Learning* 45(1):5-32.
8. Lương Mạnh Bá, Nguyễn Thanh Thuỷ (1999). *Nhập môn xử lý ảnh số - ĐHBK Hà Nội*. NXB Khoa học kỹ thuật Hà Nội.
9. Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia (2009). *Evaluation of GIST descriptors for web-scale image search*. CIVR 09, Santorini, GR.
10. Nguyễn Đăng Bình. *Giáo trình xử lý ảnh số.* Trường ĐH Huế, Khoa CNTT, Huế 9/2011.
11. Phạm Anh Phương (2009). Một số phương pháp trích chọn đặc trưng hiệu quả cho bài toán nhận dạng chữ viết tay rời rạc. Tạp chí khoa học, Đại học Huế, Số 53.
12. Prakhar Jain, Ejaz Ahmed (2009). Project Report GPU Implementation of GIST. International Institute Of Information Technology, Hyderabad.
13. Renting Liu, Jiaya Jia (2008). Reducing boundary artifacts in image deconvolution. The Chinese University of Hong Kong.
14. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>. *(ngày truy cập 16/03/2013)*
15. The USPS database of handwritten digits. [*http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html*](http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html). *(ngày truy cập 16/03/2013)*
16. Tien C. Bau. *Using Two-Dimensional Gabor Filters for Handwritten Digit Recognition*. Department of Electrical Engineering and Computer Science University of California, Irvine, CA 92697.
17. Vapnik, V (1995). The Nature of Statistical Learning Theory. Springer-Verlag.