

# Um Modelo Híbrido MLP-SVM para Reconhecimento de Caracteres Manuscritos Cursivos

Washington W. Azevedo<sup>1</sup>, Cleber Zanchettin<sup>1</sup>

<sup>1</sup>Centro de Informática – Universidade Federal de Pernambuco (UFPE)  
50.732-970 – Cidade Universitária – Recife – PE – Brasil

{wwas, cz}@cin.ufpe.br

**Abstract.** *This paper presents a hybrid MLP-SVM method for cursive characters recognition. Specialized Support Vector Machines (SVMs) are introduced to significantly improve the performance of Multilayer Perceptron (MLP) in the local areas around the surfaces of separation between each pair of characters in the space of input patterns. This hybrid architecture is based on the observation that when using MLPs in the task of handwritten characters recognition, the correct class is almost always one of the two maximum outputs of the MLP. The second observation is that most of the errors consist of pairs of classes in which the characters have similarities (e.g. (U, V), (m, n), (O, Q), among others). Specialized local SVMs are introduced to detect the correct class among these two classification hypotheses. The hybrid MLP-SVM recognizer showed improvement, significant, in performance in terms of recognition rate compared with an MLP for a task of character recognition.*

**Resumo.** *Este artigo apresenta um método híbrido MLP-SVM para o reconhecimento de caracteres cursivos. Support Vector Machines (SVMs) especializados são introduzidos para melhorar significativamente o desempenho da Multilayer Perceptron (MLP) nas áreas locais em torno das superfícies de separação entre cada par de caracteres, no espaço de padrões de entrada. Essa arquitetura híbrida é baseada na observação de que ao utilizar MLPs na tarefa de reconhecimento de caracteres manuscritos, a classe correta é quase sempre uma das duas saídas máximas da MLP. A segunda observação é que a maioria dos erros consiste em pares de classes nos quais os caracteres possuem similaridades (ex. (U, V), (m, n), (O, Q), entre outros). SVMs locais especializados são introduzidos para detectar a classe correta entre estas duas hipóteses de classificação. O reconhecedor híbrido MLP-SVM apresentou melhoria, significativa, no desempenho em termos de taxa de reconhecimento em comparação com uma MLP para uma tarefa de reconhecimento de caracteres.*

## 1. Introdução

A escrita manual é uma das formas mais naturais de comunicação entre as pessoas, constatando-se a geração de uma grande quantidade de dados manuscritos em papel. Muitas vezes é necessário processar os dados contidos nestes documentos de forma automática, sendo extremamente desejável, que computadores tenham capacidade de

‘ler’ e interpretar documentos manuscritos. O reconhecimento de caracteres manuscritos tem sido uma preocupação da comunidade científica [Plamondon et al. 2000]. As aplicações para um sistema que faça tal tipo de reconhecimento são muitas, podendo citar leitoras automáticas de cheques bancários, máquinas automáticas de processamento de códigos postais, máquinas automáticas para processar qualquer tipo de formulário preenchido manualmente, entre outros [Aires 2005]. O fato dos caracteres em um documento estarem na forma manuscrita torna a tarefa de reconhecimento automática complexa. Isto ocorre devido às variações de estilos existentes na escrita manuscrita, ou seja, o estilo pessoal de cada escritor, onde para cada escritor a forma de um caractere pode ser feita de diferentes maneiras e até o mesmo escritor pode representar o caractere de maneiras diferentes [Aires 2005]. Outro problema no reconhecimento de caracteres manuscritos é a similaridades entre caracteres distintos, por exemplo, U e V, Q e O, entre outros. Neste artigo, demonstramos a vantagem de utilizar Support Vector Machines (SVMs) [Vapnik 1999] para melhorar o desempenho de um sistema ICR (Intelligent Character Recognition ) baseado em redes neurais MLP [Rumelhard et al. 1990]. Na seção 2, é feita uma breve descrição sobre reconhecimento de caracteres e apontamos os principais problemas da tarefa de reconhecimento de manuscritos. A seção 3, descreve a base de dados C-Cube. Na seção 4, a extração de características e alguns resultados são brevemente introduzidos. Na Seção 5, explicamos a arquitetura híbrida MLP-SVM. Na seção 6 os experimentos e os resultados obtidos são analisados. Uma discussão sobre análise de erros é feita na Seção 7. A conclusão é dada na seção final.

## **2. Reconhecimento de Caracteres**

O reconhecimento de caracteres é algo que vem sendo estudado amplamente pela comunidade científica desde a invenção do computador [Veloso 1998] e consiste em, a partir de características extraídas de um conjunto de caracteres, separá-los em 10 classes, no caso dos algarismos, ou 26 classes, no caso das letras do alfabeto ocidental.

### **2.1. Principais Problemas da Tarefa de Reconhecimento de Caracteres**

Existem alguns problemas que dificultam a realização do reconhecimento de caracteres. Pode ocorrer, por exemplo, de uma imagem digitalizada possuir baixa qualidade devido a algum inconveniente durante o processo de digitalização do documento, sendo necessário realizar um pré-processamento para a eliminação de ruídos na imagem. Outro problema que dificulta esta etapa é a existência de caracteres distorcidos, principalmente se tratando de documentos manuscritos, devido às características de caligrafia do escritor, que pode dificultar inclusive o reconhecimento dos caracteres por uma pessoa.

Pode-se destacar também como um possível empecilho o fato de um mesmo documento possuir vários tipos de caracteres diferentes como, por exemplo, letras maiúsculas e minúsculas, números, caracteres especiais, letras gregas utilizadas em fórmulas matemáticas, entre outros, o que implicaria no desenvolvimento de um sistema de reconhecimento de caracteres mais generalizado e complexo. Além disto, outra dificuldade a ser considerada é a semelhança entre alguns caracteres, como “I” e “J”, “Q” e “O”, “U” e “V”, entre outros, que pode dificultar a classificação dos caracteres reconhecidos.

O trabalho [Rodrigues et al. 2002] descreve um conjunto de testes através do uso de um modelo de rede neural MLP, para a avaliação de uma técnica de extração de características proposta em [Rodrigues et al. 2001] que utiliza como base de dados um conjunto de letras manuscritas. Em [Rodrigues et al. 2001] foi feita uma avaliação desta técnica, porém a base de dados era composta somente de dígitos. Esta técnica foi desenvolvida principalmente para tratar o problema de reconhecimento de caracteres manuscritos e é baseada na projeção do contorno da imagem sobre os lados de um polígono regular construído em torno de cada caractere.

Em [Vamvakas et al. 2010], é apresentada uma metodologia para o reconhecimento de caracteres manuscritos. A metodologia proposta baseia-se em uma nova técnica de extração de características com base na subdivisão recursiva da imagem do caractere para que o resultado de sub-imagens em cada iteração tenha um número equilibrado (aproximadamente igual) de pixels de primeiro plano (foreground), na medida em que isso é possível. Nos experimentos foram utilizadas duas bases de dados de caracteres manuscritos (CEDAR e CIL), bem como duas bases de dados de dígitos manuscritos (MNIST e CEDAR). A etapa de classificação foi realizada utilizando Support Vector Machine (SVM) com Radial Basis Function (RBF). Os resultados de reconhecimento para a base de dados CEDAR foi de 94.73% e para a base de dados MNIST (99.03%).

No artigo [Cruz et al. 2010] é apresentada uma nova abordagem para o reconhecimento de caracteres cursivos usando múltiplos algoritmos de extração de características e um classifier ensemble. Várias técnicas de extração de características, usando diferentes abordagens, são extraídas e avaliadas. Duas técnicas, Modified Edge Maps e Multi Zoning, são propostas. Com base nos resultados, uma combinação dos conjuntos de características é proposta, a fim de alcançar um alto desempenho de reconhecimento. Esta combinação é motivada pela observação de que os conjuntos de características são, independentes e complementares. O ensemble é realizado através da combinação das saídas geradas pelo classificador em cada conjunto de características separado. A base de dados utilizada para os experimentos foi a C-Cube e uma rede MLP de três camadas, treinada com o Resilient Backpropagation.

Bellili apresentou em seu trabalho [Bellili et al. 2001] um método híbrido MLP-SVM para o reconhecimento de dígitos manuscritos. Essa arquitetura híbrida é baseada na idéia de que a classe correta para um dígito quase sistematicamente pertence às duas saídas máximas da MLP e que alguns pares de classes de dígitos constituem a maioria das substituições da MP (erros). SVMs locais especializados são introduzidos para detectar a classe correta entre estas duas hipóteses de classificação. O reconhecedor híbrido MLP-SVM alcançou uma taxa de reconhecimento de 98.01%, para uma tarefa real de reconhecimento de códigos postais.

Já no trabalho [Camastra 2007] é apresentado um reconhecedor de caracteres cursivos que realiza a classificação de caracteres usando SVM e neural gas. O neural gas é usado para verificar se a versão maiúscula e minúscula de uma determinada letra pode ser unida em uma única classe ou não. Feito isso para cada letra, o reconhecimento de caracteres é feita usando SVMs. SVMs apresentam melhores resultados, em termos de taxas de reconhecimento, comparado com outros classificadores neurais, tais como aprendizagem de quantização vetorial e MLPs. A taxa de reconhecimento das redes

SVMs são as mais elevadas apresentadas na literatura para o reconhecimento de caracteres cursivos.

Nos trabalhos [Rodrigues et al. 2002] e [Vamvakas et al. 2010], foi observado uma tendência de erro entre letras que possuem similaridades (ex. (B, D), (H, N) e (O, Q)). Em [Cruz et al. 2010] foi detectado uma elevada taxa de erro em caracteres que possuem duas formas completamente diferentes de escrita (ex. (a, A) e (f, F) ver figura 1).



**Figura 1. Formas diferentes para a versão maiúscula das letras A e F.**

Sistemas Inteligentes Híbridos (SIH) tem sido amplamente estudado nos últimos anos como uma alternativa para aumentar a eficiência e precisão de classificadores [Goonatilake et al. 1995]. A principal motivação para a utilização de SIHs é que uma única técnica, por causa de suas limitações e/ou deficiências, pode não ser capaz, por si só, resolver um determinado problema. Para isso a combinação de várias técnicas pode levar a uma solução mais robusta e eficiente. Neste trabalho nós mostramos que através da combinação de classificadores simples (por exemplo, MLP-SVM) para a tarefa de reconhecimento de caracteres, podemos alcançar um desempenho melhor em termos de taxa de reconhecimento, do que técnicas e métodos complexos de difícil treinamento apresentados na literatura, além de tratar o problema de semelhança entre letras (por exemplo, (A, N), (J, S) (M, N) (O, Q), entre outros) apontados em [Rodrigues et al. 2002] e [Vamvakas et al. 2010].

### **3. Base de Dados C-CUBE**

A base de dados C-Cube é uma base pública disponível para download no site Cursive Character Challenge (<http://ccc.idiap.ch>). A base de dados consiste de 57.293 imagens, incluindo letras maiúsculas e minúsculas, manualmente extraída do CEDAR e United States Post Service (USPS) databases. Como reportado por Camastra [Camastra et al. 2006], esta base de dados apresenta três vantagens:

1. A base de dados já está dividida em conjuntos de treinamento e teste, assim os resultados de diferentes pesquisadores podem ser comparados rigorosamente;
2. A base de dados não contém apenas imagens, mas também seus vetores de características extraídos utilizando o algoritmo proposto por Camastra [Camastra 2007];
3. Os resultados obtidos usando os métodos do estado da arte ainda deixam espaço significativo para melhorias significativas.

A base de dados é dividida em 38.160 (22.274 letras minúsculas e 15.886 letras maiúsculas) imagens para treinamento e 19.133 (11.161 letras minúsculas e 7.972 letras maiúsculas) imagens para teste. Todas as imagens binárias e com tamanho variável. Para cada imagem, informações são fornecidas como a distância a base e a linha superior, a distância do extremo superior da linha base e a distância do extremo inferior

da linha base. O número de exemplos para cada classe é variável e foram selecionados de acordo com sua frequência em documentos extraídos do conjunto de dados do CEDAR e USPS. Nas figuras 2 e 3 são apresentadas a distribuição das letras nas versões minúsculas e maiúsculas, respectivamente. Pode-se observar que há uma grande diferença no número de padrões entre as letras.

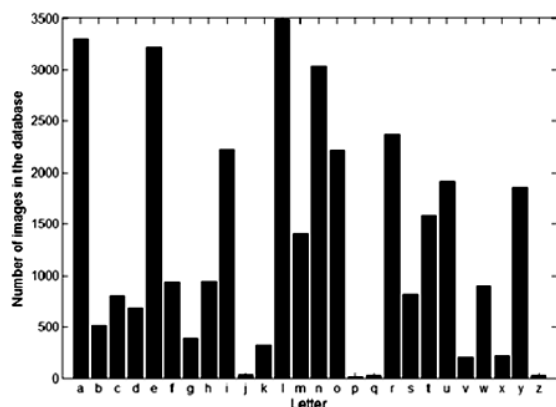


Figura 2. Distribuição de letras minúsculas na Base de Dados C-Cube.

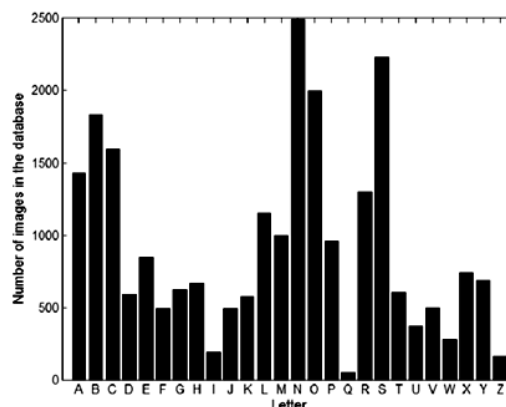


Figura 3. Distribuição de letras maiúsculas na Base de Dados C-Cube.

Thornton [Thornton et al. 2009] observou, através de engenharia reversa, que os arquivos de imagem (test.chr e training.chr) não correspondem aos vetores de características (test.vec e training.vec) disponível no site da C-Cube. Os vetores de características são de uma diferente parte da base de dados (ou seja, algumas imagens estão no conjunto de treinamento em uma parte e no conjunto de teste em outra). Por esta razão Thornton [Thornton et al. 2009] denominou os arquivos dos vetores de características da base de dados (training.vec and test.vec) como Split A e os arquivos de imagens da base de dados (test.chr and training.chr) como Split B. Foi também provado que a Split B consiste em uma divisão mais difícil da base de dados, com resultados em torno de 3 pontos percentuais menor quando comparado com a Split A.

Neste trabalho apenas o Split B foi utilizado para os experimentos, porque os arquivos de imagem do Split A não estão disponíveis para download.

#### 4. Extração de Características

A fase de extração de características pode ser definida como a extração das informações mais importantes e discriminantes para realizar a classificação para uma determinada tarefa [Devijver et al. 1982]. Existem várias técnicas de extração de características e autores sugerem que sua escolha pode ser considerada o fator mais importante para alcançar taxas de alta precisão [Trier et al. 1995].

Em [Cruz et al. 2010] dois experimentos diferentes foram feitos (todos os experimentos foram realizados utilizando uma MLP de três camadas treinada com o algoritmo Resilient Backpropagation [Riedmiller et al. 1993]): Primeiro, dividindo o conjunto de dados com letras maiúsculas e minúsculas e outro experimento com ambos os casos.

Os resultados de classificação para os casos separados e juntos são apresentados nas tabelas 1 e 2, respectivamente. Os resultados são ordenados pelas taxas de reconhecimento. A proposta de modificação do algoritmo Edge Maps apresentou o

melhor resultado global. A maioria dos conjuntos de características apresentou melhor precisão para as letras maiúsculas, com exceção do método proposto por Camastra que apresentou melhor desempenho para letras minúsculas. Este conjunto de características também apresentou a melhor precisão (84,37%) para as letras minúsculas. Pode ser visto que os métodos baseados em gradientes e o modified edge maps apresentaram os melhores resultados. Esses métodos têm em comum o uso da informação direcional. Camastra 34D feature set também usa recursos de características direcionais.

**Tabela 1. Taxa de reconhecimento pelo conjunto de características para letras maiúsculas e minúsculas separadas [Cruz et al. 2010].**

Método	No. Nodos	Maiúsculo (%)	Minúsculo (%)
Edge	490	<b>86.52</b>	81.13
Binary Grad.	490	86.35	79.89
MAT Grad.	300	85.77	79.22
Median Grad.	360	85.10	79.48
CAMASTR A 34D	400	79.63	<b>84.37</b>
Zoning	450	84.46	78.07
Structural	320	81.94	77.70
Concavities	530	73.35	81.89
Projections	500	71.73	79.90

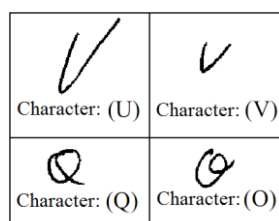
**Tabela 2. Taxa de reconhecimento pelo conjunto de características para ambos os casos [Cruz et al. 2010].**

Método	No. Nodos	Taxa de Reconhecimento (%)
Edge	490	<b>82.49</b>
Binary Grad.	490	81.46
MAT Grad.	300	80.83
Median Grad.	360	79.96
CAMASTRA 34D	400	79.97
Zoning	450	78.60
Structural	320	77.07
Concavities	530	74.90
Projections	500	73.85

## 5. Uma Arquitetura Híbrida Combinando MLP-SVM

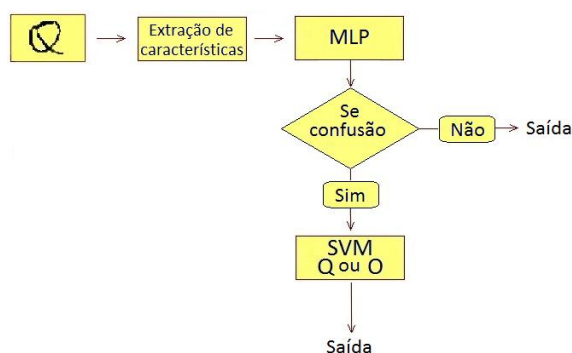
Um dos problemas que dificultam a tarefa de reconhecimento de caracteres manuscritos, é a similaridade entre caracteres distintos, (por exemplo, U e V, Q e O, entre outros) causando confusões no momento de classificar tais padrões. A figura 4 apresenta algumas similaridades entre caracteres. Esta observação motiva a busca por um método adequado que pode detectar a classificação correta entre padrões de maior confusão tendo um nível de confiança máxima na decisão de classificação. Uma vez que a decisão da MLP é feita, o problema é escolher a classe certa entre duas hipóteses de classificação. Desta forma, a escolha resulta em um problema binário, considerando que estudos realizados por [Bellili et al. 2001] identificaram que em problemas de reconhecimento de caracteres a resposta certa, na maioria dos casos, está entre as duas primeiras saídas da MLP.

Um dos métodos mais eficazes para resolver um problema de classificação binária, com a máxima confiança na decisão, é a introdução de SVMs. A eficácia desses métodos é devido à sua capacidade em construir um hiperplano de separação ótimo entre os exemplos de duas classes diferentes. A margem de separação entre estes exemplos é máxima.



**Figura 4. Similaridade entre caracteres distintos.**

Esta combinação de métodos resulta na especialização de SVMs nas áreas locais ao redor da superfície de separação entre cada par de caracteres que constituem a maioria dos erros da MLP (confusões). Assim, serão introduzidos SVMs apenas para os pares de classes que constituem as maiores confusões da MLP. O modelo híbrido MLP-SVM é mostrado na figura 5.



**Figura 5. Modelo híbrido ML-SVM: Um padrão é apresentado a MLP. Se a saída classificada pela rede MLP faz parte das classes de confusão, esta saída é apresentada ao SVM especialista para detectar a classe correta para a classificação destas duas hipóteses.**

Em [Bellili et al. 2001] a arquitetura MLP+SVM foi usada para classificar dígitos (0...9). Neste trabalho avaliamos essa arquitetura para uma tarefa de classificação de maior complexidade, que foi a classificação de caracteres cursivos com 26 padrões para cada caso (maiúsculo/minúsculo) e 39 padrões para o caso maiúsculo+minúsculo e foram obtidos resultados satisfatórios. A preocupação principal de [Bellili et al. 2001] foi em observar as duas saídas da rede com maior ativação, para analisar se a resposta estava correta. No trabalho proposto, nos preocupamos em avaliar a matriz de confusão e obter quais os pares de caracteres com maior índice de confusão e treinarmos SVMs especialistas nestas classes (conforme Figura 6).

## 6. Experimentos e Resultados

### 6.1. Processo de treinamento MLP

Todos os experimentos foram realizados utilizando uma MLP com três camadas (uma camada de entrada, uma camada escondida e uma camada de saída) e treinada com o algoritmo Resilient Backpropagation (RProp) [Riedmiller et al. 1993]. O algoritmo RProp foi escolhido porque apresentou convergência mais rápida e melhores resultados para este problema, quando comparado ao Backpropagation convencional. Os valores dos pesos foram ajustados baseando-se no erro do resultado obtido com o resultado esperado da rede. A medida de erro utilizada foi o SSE (soma dos erros quadráticos).

Com isso espera-se que a cada iteração o erro decresça. O treinamento é finalizado quando o erro do conjunto de validação crescer por cinco iterações seguidas ou até a milésima iteração, sendo estes os critérios de parada utilizados.

A arquitetura da rede MLP usada para todos os casos (maiúsculo, minúsculo e maiúsculo+minúsculo) foi a seguinte:

- Uma camada de entrada contendo 34 neurônios, referente ao número de características extraídas de cada caractere usando o algoritmo proposto por Camastra [Camastra 2007];
- Uma camada escondida contendo a seguinte quantidade de neurônios para cada caso: 530 nodos para o caso maiúsculo, 450 nodos para o caso minúsculo e 400 nodos para o caso maiúsculo+minúsculo. A quantidade de nodos na camada escondida para cada caso, foi baseada em testes preliminares para verificar a melhor configuração para a resolução do problema;
- Uma camada de saída contendo 26 neurônios para os casos maiúsculo e minúsculo (“A” a “Z” e “a” a “z” respectivamente) e 39 neurônios para o caso maiúsculo + minúsculo ((A, B, C+c, D, E, F+f, G, H, I, J+j, K+k, L, M+m, N+n, O+o, P, Q, R, S, T, U+u, V+v, W+w, X+x, Y+y, Z+z) + (a, b, d, e, g, h, i, l, p, q, r, s, t)).

Dois diferentes experimentos foram realizados: Primeiro dividindo os casos maiúsculos e minúsculos, e outro experimento com ambos os casos. Para este último, como algumas letras apresentam a mesma forma em ambas as versões maiúsculas e minúsculas, elas foram agrupadas em uma única classe (c, x, o, w, y, z, m, k, u, j, n, f, v). Camastra [Camastra 2007] utilizou uma análise de cluster para verificar se as versões maiúsculas e minúsculas das mesmas letras têm formas similares. As letras (c, x, o, w, y, z, m, k, u, j, n, f, v) apresentaram maior similaridade entre as duas versões e foram agrupadas em uma única classe. Os resultados para os casos separados e juntos são apresentados nas tabelas 3 e 4.

**Tabela 3. Taxa de reconhecimento para letras maiúsculas e minúsculas separadas.**

Classificador	Nº de nodos	Caso	Taxa de reconhecimento (%)
<i>MLP</i>	530	Maiúsculo	86.83
<i>MLP</i>	450	Minúsculo	82.12

**Tabela 4. Taxa de reconhecimento para ambos os casos.**

Classificador	Nº de nodos	Caso	Taxa de reconhecimento (%)
<i>MLP</i>	400	Maiúsculo + Minúsculo	82.18

## 6.2. Análise de Confusão

Com as redes MLP treinadas para cada caso (maiúsculo, minúsculo e maiúsculo+minúsculo) foram geradas as suas matrizes de confusão, (A figura 6 ilustra a parte referente às maiores confusões ocorridas nos experimentos com letras maiúsculas. Devido a grande quantidade de classes, 26 classes para o caso maiúsculo/minúsculo e 39 classes para o caso maiúsculo + minúsculo, é inviável exibir as matrizes de confusão completa para todos os casos) para que se possam analisar os pares de caracteres os quais a rede obteve maior número de confusão.



	A	B	C	H	K	L	N	O	R	S	U
A	308	8	4	0	1	3	8	4	5	4	0
C	1	5	370	0	0	9	1	5	1	0	0
D	1	8	2	2	1	1	3	17	0	3	0
E	1	2	9	0	2	0	0	0	9	1	0
G	0	11	5	1	0	1	1	3	1	4	0
H	4	5	0	132	3	0	11	0	0	4	0
J	1	3	0	0	0	0	6	0	0	15	1
L	0	0	4	3	0	249	1	0	4	17	0
M	1	1	0	8	1	0	25	0	2	3	0
N	6	7	1	4	4	1	545	2	3	3	0
R	11	5	5	1	1	1	4	0	277	0	1
S	1	11	4	0	0	2	2	4	2	517	1
V	0	0	0	0	1	1	4	0	0	0	14
W	0	0	1	0	0	0	10	0	0	1	1
X	2	1	0	0	6	2	4	0	7	1	0

Figura 6. Parte da matriz de confusão referente às maiores confusões ocorridas nos experimentos com letras maiúsculas.

### 6.3. Processo de treinamento SVM

Nos diferentes SVMs derivados dos pares de classes (EX. (U, V), (m, n), (N, n), entre outras) que constituem a maioria das confusões da MLP, como mostra a figura 6, diferentes funções de kernel (linear, polynomial and RBF) foram testadas e os melhores desempenhos foram obtidos pelos SVMs treinados com a função de kernel RBF.

### 6.4. Arquitetura Híbrida MLP-SVM

Baseado nas matrizes de confusão analisadas, a arquitetura híbrida para cada caso foi construída da seguinte forma:

- Caso maiúsculo: 1 MLP + 21 SVMs;
- Caso minúsculo: 1 MLP + 32 SVMs;
- Caso maiúsculo+minúsculo: 1 MLP + 33 SVMs.

A escolha dos pares de classes foi baseada na quantidade de erros tomando como valor mínimo 10% do tamanho do conjunto de teste.

A tabela 5 apresenta os resultados obtidos pelo nosso reconhecedor híbrido MLP-SVM.

Tabela 5. Resultados obtidos pelo reconhecedor MLP-SVM.

Caso	MLP (%)	SVM+MLP (%)	Melhoria (%)
Maiúsculo	86.83	90.48	3.65
Minúsculo	82.12	88.84	6.72
Maiúsculo +	82.18	87.12	4.94
Minúsculo			

Pode-se observar que todos os resultados obtidos pelo nosso método apresentou um ganho significativo quando comparado com qualquer par (extração de características, classificador) expostos nas tabelas 1 e 2 (as comparações realizadas com outros trabalhos tem que levar em consideração que os experimentos, apesar de serem

realizados com as mesmas bases de dados podem não ter sido realizados sobre as mesmas condições, visto que foram conduzidos em momentos diferentes e por pesquisadores diferentes). Vale ressaltar a simplicidade, a velocidade de treinamento e o baixo custo computacional do método, comparado com técnicas de extração de características mais complexas [Trier et al. 1995] que exigem, na maioria dos casos, custo computacional elevado e maior tempo de treinamento.

Os melhores resultados obtidos nos últimos anos, para a base de dados C-Cube, são apresentados na Tabela 6. O reconhecedor híbrido MLP-SVM proposto superou todos os resultados anteriores sobre o *Split B* da base de dados por uma boa margem. Os resultados usando o *Split A* não foram comparados, pois as imagens dos caracteres desta versão da base de dados não estão disponíveis no site. Também é importante observar que todos os melhores resultados do passado são baseados em modelos mais complexos, que levam mais tempo para serem treinados e têm alto custo computacional. Assim, provamos que usando uma técnica de extração de características simples combinada com um método híbrido formado por dois classificadores simples (MLP-SVM), temos as principais vantagens:

1. Processo de treinamento rápido: Em métodos de alta complexidade, que apresentam uma convergência lenta, o processo de treinamento pode parar em regiões de mínimo local onde os gradientes são nulos;
2. Baixo custo computacional: Métodos complexos, por sua vez, apresentam cálculos de alta complexidade, que podem causar perda de eficiência do sistema;
3. Melhores resultados do que os obtidos por modelos de alta complexidade apresentados na tabela 6.

**Tabela 6. Taxas de Reconhecimento para a base de dados C-Cube. RBF = Radial Basis Network, HVQ = Hierarchical Vector Quantization, MDF = Modified Directional Features, SVM = SVM with Radial Basis Kernel.**

Algoritmo	Taxa de Reconhecimento (%)
HVQ-32 [Thornton et al. 2008]	84.72
HVQ-16 [Thornton et al. 2008]	85.58
MDF-RBF [Thornton et al. 2009]	80.92
34D-RBF [Thornton et al. 2009]	84.27
MDF-SVM [Thornton et al. 2009]	83.60
34D-SVM + Neural GAS [Camastra 2007]	86.20
34D-MLP [Camastra 2007]	71.42
<b>Método proposto</b>	<b>87.12</b>

## 7. Análise de Erros

Alguns dos caracteres com as maiores taxas de erro são apresentados na tabela 7. Esses caracteres (g, p, V, Z e Q), são os que apresentam poucos padrões em toda a base de dados como é mostrado nas figuras 2 e 3. A rede pode não aprender a classificar esses padrões por causa do número de padrões desequilibrado entre as diferentes classes. Isso pode explicar a alta taxa de erros para essas letras. Para o caractere p, que é o caractere com o menor número de imagens na base de dados, existem apenas 9 imagens no conjunto de treinamento e 4 imagens no conjunto de teste. Todas as 4 imagens de teste

deste caractere foram classificadas erroneamente devido ao baixo número de amostras na base de dados. Para resolver este problema, uma possível solução é criar novos padrões, para as classes desequilibrada, com algumas abordagens que são amplamente utilizadas no reconhecimento de dígitos manuscritos como elastic distortions [Simard et al. 2003], [Ranzato et al. 2008] e affine transformations [Lauer et al. 2007].

**Tabela 7. Caracteres com as maiores taxas de erro.**

Letra	No. Erros	No. de padrões no conjunto de teste	(%)
<b>g</b>	39	130	30.00
<b>p</b>	4	4	100.00
<b>v</b>	35	67	52.23
<b>z</b>	5	9	64.43
<b>Q</b>	7	18	38.88

## 8. Conclusões

Neste artigo, um método para aumentar as taxas de reconhecimento de caracteres manuscritos combinando MLP+SVM é proposto. Os experimentos demonstraram que a combinação das redes MLPs com SVMs especialistas em pares de classes que constituem as maiores confusões da MLP, apresentam melhoria no desempenho em termos de taxa de reconhecimento. Os resultados mostram que houve melhoria, de 3,65% a 6,72%, na taxa de reconhecimento para todos os casos testados (maiúsculo, minúsculo e maiúsculo+minúsculo).

Como proposta para trabalho futuro, pretende-se utilizar diferentes técnicas de extração de características combinados com classifier ensemble em nosso sistema híbrido MLP-SVM, tendo como motivação conseguir melhores taxas de reconhecimento para o reconhecedor.

## Agradecimento

Este trabalho foi apoiado pela FACEPE - Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco.

## Referências

- Aires, S. B. K., Reconhecimento de caracteres manuscritos baseados em regiões perceptivas. Dissertação de Mestrado. Pontifícia Universidade Católica do Paraná. Curitiba. 2005.
- Bellili, A., Gilloux, M., and Gallinari, P., "An Hybrid MLP-SVM Handwritten Digit Recognizer," *icdar*, pp.0028, Sixth International Conference on Document Analysis and Recognition (ICDAR'01), 2001.
- Camastra, F., "A svm-based cursive character recognizer," *Pattern Recognition*, vol. 40, no. 12, pp. 3721–3727, 2007.
- Camastra, F., Spinetti, M., and Vinciarelli, A., "Offline cursive character challenge: a new benchmark for machine learning and pattern recognition algorithms.," *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 913–916, 2006.
- Cruz, R. M. O., Cavalcanti, G. D. C. and Ren, T. I., "An Ensemble Classifier For Offline Cursive Character Recognition Using Multiple Feature Extraction Techniques"; *IEEE International Joint Conference on Neural Networks (IJCNN)*, Barcelona, p. 744-751, 2010.

- Goonatilake, S. and Khebabal, S. (Eds.). *Intelligent Hybrid Systems*, vol. 1. John Wiley and Sons, 1995.
- Haykin, S. *Neural Networks - A Comprehensive Foundation*. [S.I.]: Prentice Hall, 1996.
- Ho, T. K., Hull, J. J., Srihari, S.N. (1994) Decision Combination in Multiple Classifier Systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(1):66–75.
- Lauer, F., Suen, C. Y., and Bloch, G., “A trainable feature extractor for handwritten digit recognition,” *Pattern Recognition*, vol. 40, no. 6, pp. 1816–1824, 2007.
- Nadal, C., Legault, R., Suen, C. Y. (1990) Complementary Algorithms for the Recognition of Totally Unconstrained Handwritten Numerals. In: *Proceedings of the tenth International Conference on Pattern Recognition 1990*. Atlantic City, NJ, USA, pp 443–449.
- Plamondon, R. and Srihari, S. N., "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, p. p. 63-84, 2000.
- Ranzato, M., Boureau, Y., and LeCun, Y., “Sparse feature learning for deep belief networks,” *Advances in Neural Information Processing Systems*, pp. 1185–1192, 2008.
- Riedmiller, M. and Braun, H., “A direct adaptive method for faster backpropagation learning: The rprop algorithm,” *Proceedings of the IEEE International Conference on Neural Networks*, pp. 586–591, 1993.
- Rodrigues, R. J., Kupac, G. V., Thomé, A. C. G.; “Character Feature Extraction Using Polygonal Projection Sweep (Contour Detection)”, *IWANN2001*, Granada - Spain, June 13 a 15, LINCOS, 2001, pp 687-695, 2001.
- Rodrigues, R. J., Silva, E., Thomé, A. C. G., “Extração de Características para o Reconhecimento de Letras Manuscritas”, *V Simpósio Brasileiro de Automação Inteligente - V SBAI*, Canela, novembro de 2002.
- Rodrigues, R. J., Silva, E., Thomé, A. C. G., “Feature Extraction Using Contour Projection”; *5th World Multiconference on Systemics, Cybernetics and Informatics SCI2001 – Orlando – USA*, July 22 – 25.
- Rumelhard, D.E. and Weigend, S.A., "Predicting the Future: A Connectionist Approach", *International Journal of Neural Systems*, pp. 193-209, 1990.
- Simard, P. Y., Steinkraus, D., and Platt, J. C., “Best practices for convolutional neural networks applied to visual document analysis,” *International Conference on Document Analysis and Recognition*, vol. 2, pp. 958–963, 2003.
- Thornton, J., Blumenstein, M., Nguyen, V., and Hine, T., “Offline cursive character recognition: A state-of-the-art comparison,” *14th Conference of the International Graphonomics Society*, 2009.
- Thornton, J., Blumenstein, M., Nguyen, V., and Hine, T., “Offline cursive character recognition: A state-of-the-art comparison,” *14th Conference of the International Graphonomics Society*, 2009.
- Thornton, J., Faichney, J., Blumenstein, M., and Hine, T., “Character recognition using hierarchical vector quantization and temporal pooling,” *Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence*, pp. 562–572, 2008.
- Trier, O. D., Jain, A. K., and Taxt, T., “Feature extraction methods for character recognition: A survey,” *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, 1995.
- Vamvakas, G., Gatos, B., Perantonis, S. J., "Handwritten character recognition through two-stage foreground sub-sampling"; *Pattern Recognition* 43 (2010) 2807–2816.
- Vapnik, V. N., “An overview of statistical learning theory”; *IEEE Trans. on Neural Networks*, 10(5):988–999, 1999.
- Veloso, L. R., “Reconhecimento de Caracteres Numéricos Manuscritos.” *Dissertação de Mestrado*, Universidade Federal da Paraíba, Campina Grande, 1998.
- Woods, K., Jr, W. P. K., and Bowyer, K.. Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. on PAMI*, 19(4):405–410, 1997.