

# Análise de Técnicas de Extração de Características para o Reconhecimento de Dígitos Manuscritos

Rafael M. O. Cruz, George D. C. Cavalcanti e Tsang Ing Ren

*Centro de Informática*

*Universidade Federal de Pernambuco*

*Recife, PE, Brasil - [www.cin.ufpe.br/~viisar](http://www.cin.ufpe.br/~viisar)*

*Email: {rmoc,gdcc,tir}@cin.ufpe.br*

**Resumo**—O reconhecimento de dígitos manuscritos é um dos problemas fundamentais da área de visão computacional. A escolha da técnica para fazer a extração de características da imagem pode ser considerada o fator mais importante para conseguir altas taxas de reconhecimento. Este artigo apresenta uma análise detalhada de cinco diferentes técnicas de extração de características, três já consolidadas e duas modificações, propostas neste trabalho, para técnicas conhecidas. Os testes foram realizados utilizando a base de dados MNIST e redes neurais do *Multi-Layer Perceptron* como classificadores. A partir dos resultados são propostas três soluções para melhorar o reconhecimento do sistema: (1) Extração de características com maior poder discriminativo; (2) Combinação de diferentes técnicas de extração de características; (3) Utilização de um módulo de verificação.

**Palavras-chave**—Reconhecimento de dígitos; Extração de características; Redes Neurais; Reconhecimento de Padrões.

## I. INTRODUÇÃO

O reconhecimento de dígitos manuscritos é um dos problemas clássicos da visão computacional. O interesse nesta área existe pelo fato de que ela possui um grande potencial para aplicações, principalmente em domínios nos quais uma grande massa de documentos precisa ser analisada [1]. Por exemplo: reconhecimento de códigos postais, processamento de formulários e reconhecimento do valor em um cheque bancário. Porém, para ser utilizado em aplicações reais o sistema precisa atingir um alto grau de confiabilidade, pois para certas aplicações o custo de uma classificação errada é muito alto.

O grande problema encontrado no reconhecimento de caracteres manuscritos, em geral, deve-se ao fato da grande variedade de estilos de escrita. Estudo apresentado por Srihari et al. [2] discute a individualidade do estilo de escrita para numerais e mostra que uma mesma pessoa pode escrever um mesmo dígito usando mais de um estilo de escrita. Isto ocorre principalmente no caso dos dígitos mais complexos como nos casos dos dígitos quatro e oito. Esta variabilidade torna o problema muito mais complexo quando comparado ao de reconhecimento de dígito impresso, no qual já existem soluções eficientes disponíveis.

Um sistema para reconhecimento de caracteres manuscritos pode ser dividido basicamente em quatro etapas: Pré-

processamento, Extração de Características, Classificação e Pós-Processamento. Um fator importante para desenvolver um sistema de reconhecimento de caracteres é a escolha da técnica para fazer a extração das características da imagem. Existem varias técnicas disponíveis na literatura [3]. Porém a maioria dessas técnicas é testada em condições diferentes, ou seja, usando bases de dados, técnicas de normalização e classificadores diferentes. Portanto fica difícil fazer uma comparação dos seus resultados para então poder selecionar as mais promissoras.

A contribuição deste trabalho é estudar cinco diferentes técnicas de extração de características utilizando a mesma base de dados e o mesmo classificador. Destas cinco técnicas, três são bem conhecidas na literatura, *Structural Characteristics* [4], *Concavities Measurement* [5] e *Image Projections* [6]. As outras duas são modificações dos algoritmos *Zoning* [3] e *Edge Maps* [6], propostas neste artigo, para melhorar os seus desempenhos. Estas técnicas foram escolhidas por possuírem abordagens bem diferentes (projeções, concavidades, bordas) e por serem abordagens muito utilizadas para o reconhecimento de dígitos manuscritos. A partir dos resultados será feito um estudo comparativo entre estas técnicas, mostrando onde cada uma apresentou melhor desempenho e em quais partes cada algoritmo apresenta maiores problemas. Por último, a partir deste estudo, algumas melhorias para aumentar a capacidade de reconhecimento serão propostas.

Na Seção II serão explicadas as técnicas de extração de características utilizadas. Na Seção III é mostrada, para cada dígito manuscrito, quais características extraídas contribuíram mais para o seu reconhecimento. Na Seção IV será apresentada a conclusão e algumas sugestões para futuras melhorias.

## II. TÉCNICAS DE EXTRAÇÃO DE CARACTERÍSTICAS

Extração de características pode ser definida como sendo a captura das informações mais relevantes para fazer uma classificação de um dado fornecido como entrada [7]. Existem várias técnicas para extrair características de dígitos manuscritos e a sua escolha é considerada como fator mais

importante para o sistema alcançar altas taxas de confiabilidade [3]. As cinco técnicas utilizadas são descritas a seguir.

#### A. Características Estruturais (Structural Characteristics)

A técnica *Structural Characteristics* consiste em calcular algumas projeções e *profiles* da imagem de entrada, combinando-os para formar um único vetor de características. Primeiramente a imagem deve ser normalizada para uma matriz de tamanho 32x32 pixels. Em seguida são calculadas as projeções horizontais e verticais, a projeção radial e os *profiles In-Out* e *Out-In* também radiais [4]. Na Figura 1 é mostrado um exemplo das três projeções, para uma imagem do dígito 2.

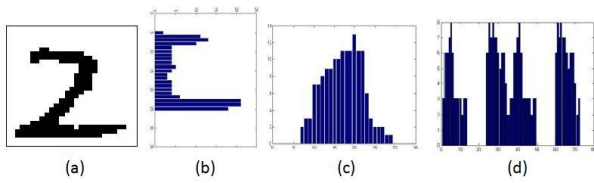


Figura 1. a) Exemplo do dígito dois, b) Projeção horizontal, c) Projeção vertical, d) Projeção radial

1) *Projeção Horizontal*: A projeção horizontal é calculada simplesmente pela soma dos pixels pretos em cada linha da imagem, totalizando 32 características.

2) *Projeção Vertical*: A projeção vertical é calculada pela soma dos pixels pretos de cada coluna da imagem, totalizando um vetor com 32 características.

3) *Projeção radial*: A projeção radial é calculando pela soma dos pixels pretos na imagem, partindo do centro geométrico da imagem e caminhando em direção a borda em ângulos pré-definidos. Os ângulos foram selecionados em intervalos de 5 graus, totalizando 72 direções (características).

4) *Radial In-Out Profile*: Este *profile* pode ser visto como a posição do primeiro pixel preto encontrado através de uma busca, que parte do centro geométrico da imagem e caminha em direção a borda nas direções de ângulos pré-definidos. Assim como na projeção radial, os ângulos são selecionados em intervalos de 5 graus, totalizando 72 direções diferentes. A busca começa do centro da imagem e caminha na direção do ângulo em questão em passos de um pixel. A cada passo é verificado se a cor do pixel é preta. Caso o pixel seja de cor preta, o número de passos realizados até o momento é retornado como resultado.

5) *Radial Out-In Profile*: Este *profile* tem a definição semelhante à do *Profile In-Out*. A diferença é que neste *profile* a posição do último pixel preto encontrado é utilizada como característica.

Após o cálculo das projeções e *profiles*, os valores de cada um são normalizados, para um valor no intervalo [0,1]. Esta normalização é feita simplesmente pela divisão de cada característica pelo maior valor que pode ser assumido para

cada projeção ou *profile*. Nos casos das projeções horizontais e verticais, é feita uma divisão por 32 que é o número de linhas e colunas da imagem. Para a projeção radial e *profiles*, é feita uma divisão por 16, pois esse é o número máximo de passos que podem ser realizados. Com os resultados normalizados as projeções e *profiles* formam um único vetor contendo 280 ( $32 + 32 + 72 + 72 + 72$ ) características.

#### B. Mapa de bordas (Edge Maps)

Esta técnica consiste em extrair segmentos de linhas que representem o caractere, para então fazer o reconhecimento. Uma justificativa para usar segmentos de linhas, deve-se ao fato de que caracteres manuscritos são, em sua essência, linhas em um espaço bi-dimensional [6]. Antes de extrair os segmentos de linhas, a imagem deve ser redimensionada para uma de tamanho 25x25 pixels e deve ser utilizada uma estratégia de *thinning*, pois este pré-processamento facilita a extração das bordas da imagem [6]. Para isto foi utilizado o algoritmo de Zhang-Suen [8].

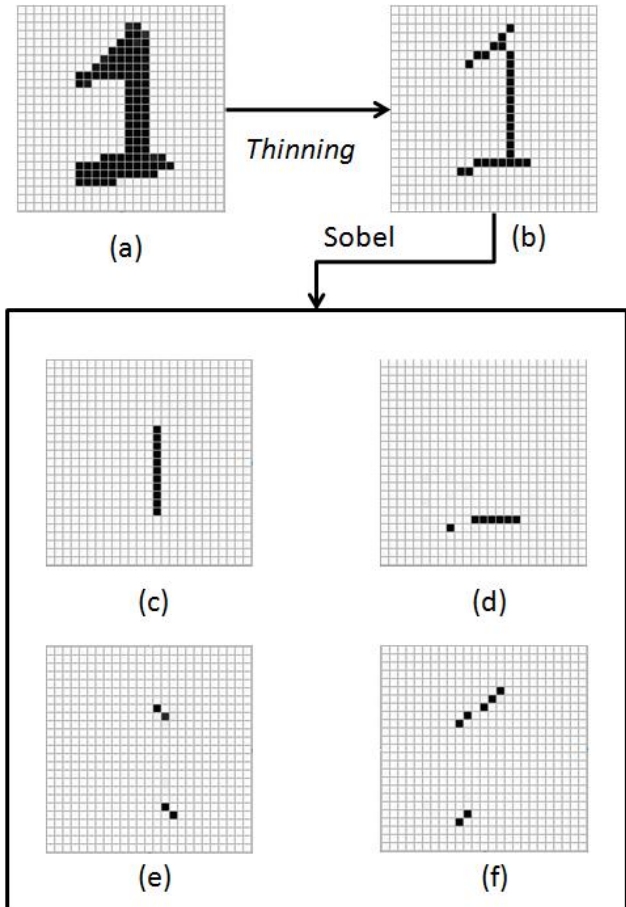


Figura 2. Imagens com os passos do algoritmo *edge maps*. a) Imagem original, b) Resultado do *thinning*, c) Mapa de bordas vertical, d) Mapa de bordas horizontal, e) Mapa de bordas diagonal  $-45^\circ$ , f) Mapa de bordas diagonal  $45^\circ$

Após o pré-processamento da imagem, o próximo passo

é extrair os segmentos horizontais, verticais e diagonais. Para isto foram utilizados os operadores de Sobel [9], para cada direção definida anteriormente, que atuam como detectores de bordas. Quatro novas imagens são obtidas com o resultado da convolução de cada um na imagem principal. Na Figura 2 são apresentados os resultados do *thinning* e da aplicação do operador de Sobel para um exemplo do dígito um (1). Essas imagens, juntamente com a imagem original, são divididas em 25 partes de tamanhos 5x5 pixels cada. A utilização da imagem original também é importante para conseguir capturar as características globais dos dígitos. Para cada parte, o percentual de pixels preto é calculado (número de pixels pretos dividido pela quantidade de pixels da parte) obtendo para cada imagem um vetor com 25 características. O resultado de cada imagem é combinado para formar um único vetor com 125 características (25 para cada imagem). Os seus valores não precisam ser normalizados, pois já pertencem ao intervalo [0,1].

A técnica descrita acima apresenta algumas diferenças em relação ao original Chim et al. [6]. No algoritmo original, após a extração das bordas é feita uma compressão de cada uma das cinco imagens para a dimensão de 5x5 pixels. O valor de cada pixel das novas imagens é então utilizado como características. As modificações no algoritmo original foram necessárias, pois os resultados obtidos, com a técnica original, não foram satisfatórios.

### C. Projeções de Imagem (Image Projections)

Extrações de características baseadas em projeções têm sido usadas com sucesso no reconhecimento de caracteres [6]. Para esta técnica foram utilizadas as projeções diagonais e radiais. A projeção radial é calculada agrupando os pixels pela sua distância ao centro da imagem. O problema deste tipo de projeção é ser totalmente invariante à rotação, o que é indesejável para fazer reconhecimento de dígitos, pois com invariância à rotação não é possível diferenciar o dígito seis do dígito nove. Para remover a invariância, a imagem deve ser dividida em quatro partes: superior, inferior, esquerda e direita. Na Figura 3 é mostrada a divisão da imagem em quadrantes e suas respectivas projeções. Para cada uma destas partes é calculado um histograma agrupando os pixels pretos que possuem a mesma distância radial, em pixels, em relação ao centro da imagem.

A projeção diagonal é obtida agrupando os pixels nas linhas diagonais. Mais detalhes sobre o algoritmo pode ser encontrado Chim et al. [6]. Na Figura 4 são apresentadas as projeções diagonais para um exemplo do dígito quatro.

Os valores de cada projeção são então normalizados através da divisão pelo maior valor possível e são combinados para formar um único vetor contendo 128 (16 para cada projeção radial, e 32 para cada projeção diagonal) características.

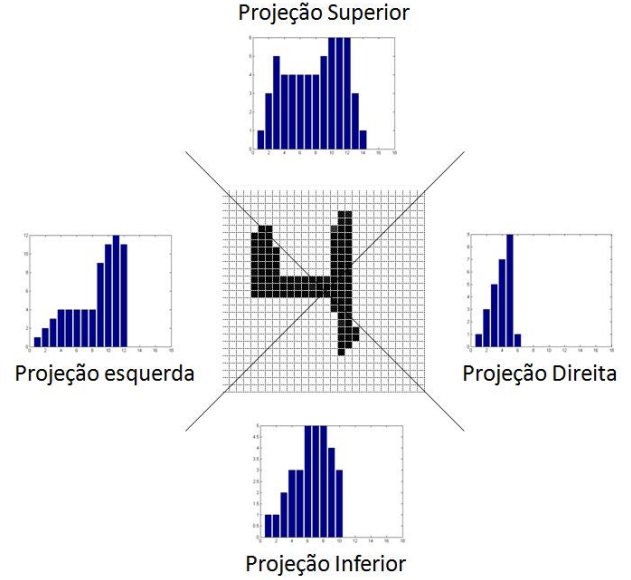


Figura 3. Projeções radiais nas quatro direções

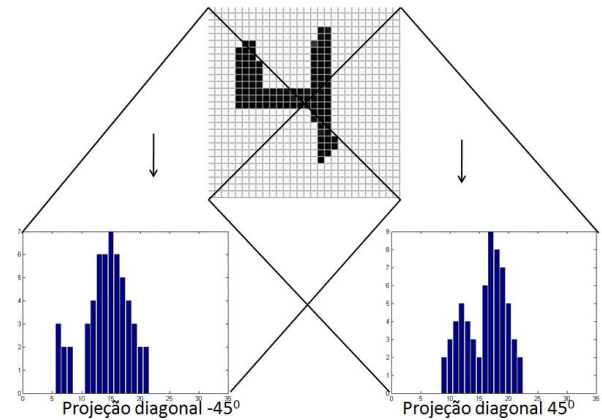


Figura 4. Projeções diagonais

### D. Zoning

O algoritmo de *Zoning* consiste em dividir uma imagem em várias regiões e calcular o percentual de pontos pretos para cada região. Na Figura 5 é ilustrada a divisão em zonas para uma imagem do dígito três. Durante a implementação da técnica foi identificado, a partir de testes preliminares, que utilizar apenas uma configuração de divisão para a imagem não era suficiente para obter bons resultados. Por este motivo foi decidido utilizar várias configurações de divisão, modificando o tamanho das zonas, e combinando cada resultado para formar o vetor de características. Foram escolhidas 13 configurações diferentes (3x1, 1x3, 2x3, 3x2, 3x3, 1x4, 4x1, 4x4, 6x1, 1x6, 6x2, 2x6 e 6x6) resultando em um vetor com 123 características (3 + 3 + 6 + 6 + 9 + 4 + 4 + 16 + 6 + 6 + 18 + 18 + 36). Na Figura 5 é apresentada a divisão feita por algumas destas configurações. Os seus

valores não precisam ser normalizados, pois como eles são calculados através do percentual eles já se encontram no intervalo [0,1].

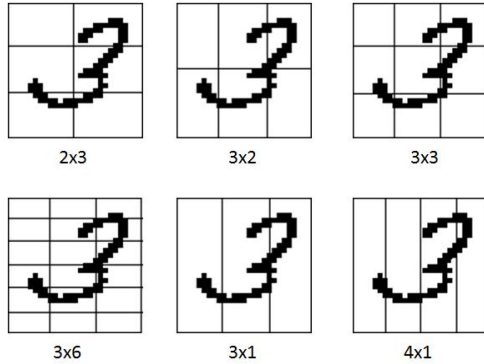


Figura 5. Exemplos de divisões utilizadas na técnica *zoning*

### E. Medida de Concavidades (Concavities Measurement)

A idéia desta técnica é fazer uma busca, para cada pixel branco (*background*) na imagem, considerando quatro direções, com o objetivo de saber para quais direções a busca não encontrou nenhum pixel preto [5]. Na Figura 6d) são especificadas as direções da busca. Nos casos em que um pixel preto é alcançado nas quatro direções, uma nova busca é lançada utilizando quatro direções auxiliares para garantir que o pixel esta dentro de uma área fechada [5]. Na Figura 6c) são ilustradas as quatro direções auxiliares.

Para a implementação da técnica, primeiramente, é preciso definir um vetor contendo treze elementos. Cada posição deste vetor equivale a um tipo de configuração diferente que representa a maneira na qual um pixel branco alcança um pixel preto. Na Figura 6b) é mostrado o exemplo de um vetor e as configurações que cada elemento deste vetor representa. As configurações de cada posição do vetor são especificadas por dois valores na figura. O valor de cima representa a quantidade de pixels pretos atingidos. O valor de baixo especifica as direções nas quais nenhum pixel preto é alcançado. Por exemplo, no caso do ponto  $X_2$ , a direção 3 não atinge um ponto preto, portanto, a configuração correspondente a este pixel é a da posição de número sete no vetor. Para o ponto  $X_1$ , as quatro direções chegam a um pixel preto, precisando realizar uma nova busca utilizando as direções auxiliares. Para fazer a busca auxiliar, o valor adotado para o salto foi de três pixels. O resultado da busca auxiliar mostra que na direção  $S_1$  nenhum pixel preto é encontrado. Portanto a posição de número nove do vetor é incrementada.

Para iniciar a execução do algoritmo, a imagem foi dividida em seis partes, duas colunas e três linhas, cada uma possuindo seu próprio vetor de características. A busca é então realizada para cada parte separadamente. Com o resultado da busca, a posição do vetor que se enquadra no

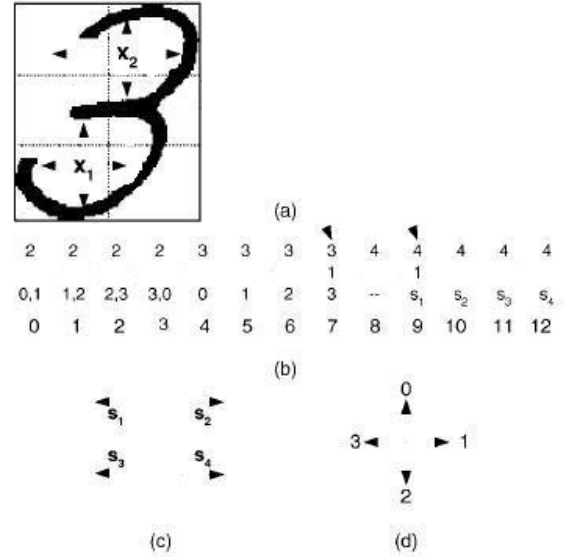


Figura 6. a) Cálculo da concavidade, b) vetor de características, c) direções auxiliares, d) quatro direções principais [Adaptado de [5]].

resultado é incrementada. No final, os resultados de cada parte são concatenados para formar o vetor final de 78 (13x6) características. Para esta técnica não foi realizada normalização nos valores.

## III. EXPERIMENTOS E RESULTADOS

Foram realizados experimentos para cada técnica de extração de características descrita no capítulo anterior. A base de dados utilizada nos experimentos foi a MNIST [10] que consiste em 60 mil dígitos para o conjunto de treinamento e 10 mil para testes. Esta base foi formada a partir de um subconjunto de imagens das bases NIST *Special Database* um e três. O conjunto de treinamento conta com, em média, seis mil imagens para cada classe, totalizando 60 mil. O conjunto de teste conta com, em média, mil exemplos para cada dígito, totalizando 10 mil. Todas as imagens do conjunto já se encontram normalizadas, binarizadas e possuem tamanho 28x28 pixels. A grande vantagem de se utilizar esta base de dados é não se preocupar com a etapa de pré-processamento da imagem, podendo focar apenas nas etapas de extração de características e classificação. Outra vantagem de utilizar esta base de dados ocorre pelo fato de que o conjunto de testes já vem separado do treinamento, o que facilita a comparação dos resultados obtidos por outros trabalhos. Todos os experimentos foram feitos utilizando redes neurais do tipo *Multilayer Perceptron* (MLP) [11] como classificador.

O treinamento das redes neurais foi realizado separando 25% do conjunto de treinamento para formar o conjunto de validação, totalizando 15 mil (1500 em média para cada dígito) padrões para validação, sobrando 45 mil para treinamento (4500 imagens em média para cada dígito).

As imagens para formar o conjunto de validação foram selecionadas aleatoriamente, porém mantendo a proporção de 1500 imagens para cada classe. A rede foi treinada utilizando o algoritmo *Resilient Backpropagation* [12] por apresentar uma convergência mais rápida em comparação ao *Backpropagation* tradicional. A utilização de uma técnica de treinamento mais rápida é importante, pois, devido ao tamanho da base de dados, o custo de treinamento é alto.

#### A. Resultados

Para cada técnica de extração foram realizados vários testes, com intuito de encontrar a melhor configuração da rede para cada técnica. Após testes preliminares, as configurações que apresentaram os melhores resultados, para cada técnica, foram selecionadas. Para as técnicas *Image Projections* e *Edge Maps*, foi selecionada uma configuração com 300 neurônios na camada escondida. Para as técnicas *Zoning*, *Structural Characteristics* e *Concavities Measurement*, foram escolhidas configurações com 150, 290 e 175 neurônios na camada intermediária, respectivamente. Para estas configurações foram realizadas dez iterações para obter a média e o desvio padrão. Na Tabela I é apresentada as taxas de erros do melhor desempenho obtido. Na Figura 7 é mostrado um gráfico com a média das 10 iterações e o desvio padrão para cada técnica.

Tabela I  
MENORES TAXAS DE ERROS POR TÉCNICA

Dígito	Structural	Edge	Projections	Zoning	Concavities
0	1,12%	2,14%	1,83%	1,12%	3,87%
1	0,88%	1,85%	1,58%	1,05%	1,67%
2	3,97%	4,92%	4,74%	3,77%	4,34%
3	3,86%	5,24%	5,24%	3,16%	8,31%
4	2,75%	7,84%	3,67%	2,95%	7,02%
5	4,37%	5,33%	6,39%	3,03%	4,44%
6	2,19%	3,34%	2,82%	2,92%	3,65%
7	3,11%	6,49%	4,57%	4,38%	5,62%
8	4,00%	6,46%	6,26%	4,10%	10,36%
9	4,40%	9,77%	6,15%	4,86%	7,89%
<b>Média</b>	<b>3,05%</b>	<b>5,32%</b>	<b>4,28%</b>	<b>3,12%</b>	<b>5,69%</b>

A partir dos resultados, é possível observar que as técnicas *Structural Characteristics* e *Zoning* alcançaram os melhores resultados, obtendo taxas de acerto semelhantes. Também pode ser visto que a maior parte dos erros ocorreu em dígitos que apresentam estruturas semelhantes a outros, como nos casos de (4-9-8) e (2-3-5). Estes numerais apresentaram um alto índice de classificações erradas entre si como pode ser visto nas Tabelas II e III, onde são apresentadas as matrizes de confusão para as técnicas *Structural Characteristics* e *Concavities Measurement*. Estas foram as técnicas que alcançaram o melhor e o pior resultado, respectivamente. Uma abordagem para resolver este tipo de problema seria, primeiramente, fazer uma consulta para saber se o dígito pertence a um desses grupos (4-9-8), (2-3-5), (1-7), para depois utilizar uma técnica que consiga melhor separar estes numerais. Em Ping et al. [13] pode ser encontrado

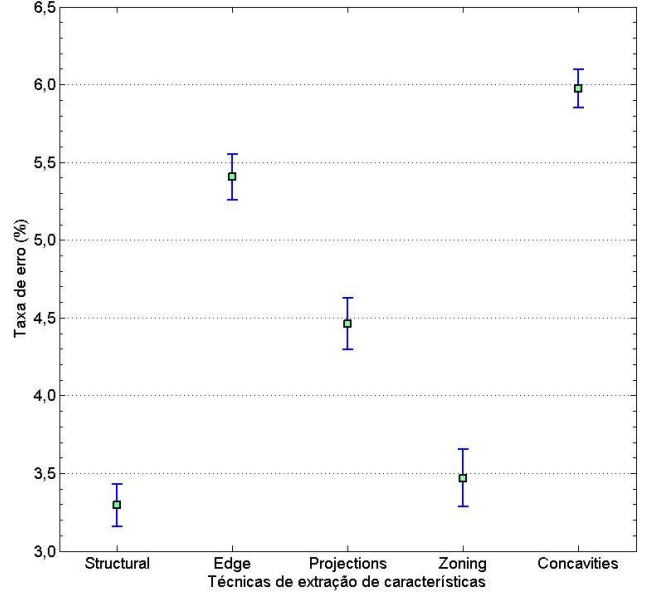


Figura 7. Gráfico com a média do erro e o desvio padrão para cada técnica

um trabalho que utiliza este tipo de abordagem. Já para o caso de dígitos mais simples como o dígito zero e o dígito um, as taxas de erros foram baixas em todas as técnicas. Isto pode ser explicado pelo fato de que por serem mais simples, estes numerais não apresentam muitas variações nos seus estilos de escrita. A única exceção foi a técnica de medidas de concavidades que obteve uma taxa de erro para o numeral zero muito superior as demais, pois ela apresentou um alto índice de zeros que foram classificados erradamente como oito como pode ser visto na Tabela III. Isto pode ser explicado pelo fato de que para ambos os caracteres, ou os pixels pertencem a uma região totalmente fechada ou a uma região que possui a mesma direção fechada.

Tabela II  
MATRIZ DE CONFUSÃO PARA TÉCNICA STRUCTURAL CHARACTERISTICS

Real	Predita									
	0	1	2	3	4	5	6	7	8	9
0	969	1	1	0	3	2	1	1	0	2
1	0	1125	3	1	0	1	2	0	3	0
2	3	1	991	11	3	2	4	9	6	2
3	1	0	9	959	0	14	1	8	17	1
4	1	0	3	0	955	0	5	1	4	13
5	5	1	2	14	4	853	9	1	10	2
6	4	2	1	1	4	6	938	0	2	1
7	0	3	13	3	4	1	0	994	2	8
8	3	1	4	8	4	7	3	3	935	6
9	5	3	0	6	10	3	2	7	9	964

Também pode ser observado que as técnicas apresentaram resultados bem diferentes para cada dígito, principalmente para os mais complexos. Para o caso dos dígitos zero, um, quatro e sete, a técnica de características estruturais obteve melhores resultados enquanto para os dígitos dois, três, cinco, oito e nove a técnica de zoneamento apresentou



Tabela III  
MATRIZ DE CONFUSÃO PARA TÉCNICA CONCAVITIES MEASUREMENT

Real	Predita									
	0	1	2	3	4	5	6	7	8	9
0	942	0	1	4	0	5	9	0	17	2
1	0	1116	8	2	0	0	4	1	3	1
2	5	1	978	20	2	1	3	7	6	0
3	0	5	24	926	1	20	0	23	4	7
4	3	6	8	0	913	1	7	3	17	24
5	3	1	2	17	0	852	3	4	7	3
6	9	6	2	0	4	7	923	0	7	0
7	0	10	9	12	6	1	1	971	2	16
8	16	2	7	6	11	14	12	6	873	27
9	5	5	1	8	13	3	0	20	22	977

menor taxa de erro. No caso do dígito seis a técnica de projeções obteve o melhor resultado. Isto mostra que diferentes técnicas possuem um melhor poder para reconhecer determinados grupos de dígitos. Outro fator que deve ser observado é que as demais técnicas apresentaram erros em partes diferentes. Alguns dígitos apresentaram erro por apenas uma das técnicas, outros por mais de uma e outros por todas elas. Tomando como exemplo as técnicas *Zoning*, *Structural Characteristics* e *Concavities Measurements*, apenas nove dígitos foram classificados erradamente pelas três. Na Figura 8 é apresentado um diagrama que mostra as interseções das classificações erradas por estas três técnicas.

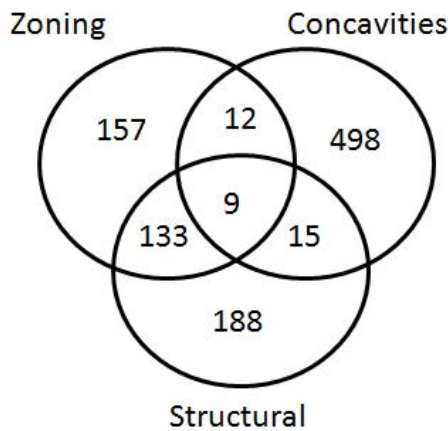


Figura 8. Diagrama da interseção dos erros entre as técnicas *Structural Characteristics*, *Zoning* e *Concavities Measurement*

A partir desta figura pode ser observado que embora a técnica *Concavities Measurement* tenha apresentado uma taxa de erro bem maior em relação às outras, a quantidade de erros comuns entre ela e as outras duas é bem pequena. Isto mostra que os erros cometidos por ela são em padrões de estilos bem diferentes das outras. Portanto, mesmo que o desempenho dela tenha sido inferior, ainda é interessante utilizar este algoritmo combinado com os outros, pois eles são complementares. Esta grande diminuição dos erros na interseção entre as técnicas *Zoning* e *Structural Characteristics* com a *Concavities Measurement* pode ser explicado pelo fato de esta última apresentar uma abordagem bem

diferente das outras duas. As duas primeiras são baseadas nas quantidades de pixels pretos por região da imagem enquanto a última é baseada na quantidade de direções que um pixel branco (*background*) atinge um pixel preto.

Considerando as cinco técnicas, 1447 dígitos diferentes apresentaram erros (erro que ocorreu em mais de uma técnica é contado como apenas um erro). Destes 1447 erros, a grande maioria (1010) foi gerado por apenas uma técnica, enquanto apenas oito erros foram gerados por todas as cinco. Na Tabela IV é apresentada a quantidade de exemplos que foram classificados errados por número de técnicas. Na primeira coluna aparece a quantidade de erros cometidos por apenas uma técnica (qualquer técnica), na segunda coluna aparece a quantidade de erros cometidos por apenas duas técnicas (quaisquer duas) e assim sucessivamente. As imagens que obtiveram erros em todas elas são apresentadas na Figura 9. Estes números indicam que uma combinação entre estas técnicas é uma abordagem interessante para alcançar melhor desempenho. Eles também podem indicar que o limite de uma combinação ideal destas técnicas seria de oito padrões errados, gerando um erro de 0.08%.

Tabela IV  
DISTRIBUIÇÃO DOS ERROS COMUNS ENTRE AS TÉCNICAS DE EXTRAÇÃO DE CARACTERÍSTICAS

No. de técnicas que cometeram o mesmo erro	1	2	3	4	5	Total
No. de erros	1010	247	115	67	8	1447

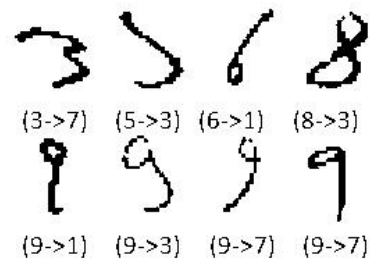







Figura 9. Imagens com classificações erradas por todas as técnicas

Uma dificuldade encontrada durante a classificação foi em relação aos dígitos que apresentam alguns tipos de problemas como ruídos, degradação, tipo de escrita, distorção e erros durante o processo de segmentação. Para estas imagens, o que normalmente é feito na prática é colocar um limiar na saída do classificador para rejeitá-las. Desta forma, a confiabilidade do sistema é aumentada. Na Tabela V são mostrados exemplos desses dígitos e as técnicas que erraram cada um. A inclusão de um módulo de verificação não mudaria a taxa de reconhecimento do sistema, mas sim sua confiabilidade. A quantidade de dígitos reconhecidos seria a mesma ou até mesmo menor, porém a quantidade de dígitos classificados erradamente diminuiria. A utilização de

um módulo de verificação é importante, pois, para algumas aplicações, um erro é muito menos tolerável do que uma rejeição [14].

Tabela V  
ERROS COMUNS ENTRE AS TÉCNICAS

					
Dígito	1	6	5	0	4
Structural	X	X			X
Edge			X	X	
Projections	X	X	X	X	X
Zoning	X	X		X	X
Concavities		X			X

#### IV. CONCLUSÃO

O reconhecimento de dígitos manuscritos *off-line* é um dos problemas clássicos da visão computacional, possuindo várias aplicações em potencial. Várias pesquisas têm sido feitas nesta área, porém muito ainda precisa ser feito para conseguir resultados mais confiáveis. Dentre as etapas de um sistema de reconhecimento de caracteres manuscritos *off-line*, a etapa de extração de características pode ser considerada a mais importante para obter boas taxas de reconhecimento.

Várias técnicas de extração de características foram estudadas. Através dos resultados foi comprovado que a escolha da técnica para fazer a extração de características da imagem é um fator crucial para o desempenho no reconhecimento, pois, com a utilização do mesmo classificador, as técnicas estudadas apresentaram taxas de reconhecimento bem diferentes. Também pode ser concluído, com base nos resultados, que técnicas diferentes podem apresentar um melhor poder de classificação para determinados dígitos e que as técnicas apresentam erros em diferentes imagens, indicando que uma possível combinação entre elas é uma boa alternativa para aumentar a capacidade de reconhecimento.

Três pontos podem ser citados como melhorias futuras: estudar uma maneira para combinar as demais técnicas de extração de características de forma que maximize a capacidade de reconhecimento; inclusão de novas formas de extração de características que possuam resultados bons, principalmente para os dígitos que apresentaram desempenhos mais fracos e, a utilização de uma estratégia de rejeição para que o sistema rejeite imagens que possam ser classificadas erradamente por apresentarem problemas como degradação, ruídos ou tipo de escrita.

#### REFERÊNCIAS

- [1] C. O. A. Freitas, L. S. Oliveira, F. Bortolozzi, and S. B. K. Aires, "Handwritten character recognition using nonsymmetrical perceptual zoning," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, no. 1, pp. 135–155, 2007.
- [2] S. N. Srihari, C. I. Tomai, B. Zhang, and S. Lee, "Individuality of numerals," *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pp. 1086–1091, 2003.
- [3] O. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition: A survey," *Pattern Recognition*, no. 4, pp. 641–662, 1995.
- [4] E. Kavallieratou, K. Sgarbas, N. Fakotakis, and G. Kokkinakis, "Handwritten word recognition based on structural characteristics and lexical support," *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pp. 562–567, 2003.
- [5] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Automatic recognition of handwritten numerical strings: A recognition and verification strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1438–1454, 2002.
- [6] Y. Chim, A. A. Kassim, and Y. Ibrahim, "Dual classifier system for handprinted alphanumeric character recognition," *Pattern Analysis and Application*, no. 1, pp. 155–162, 1998.
- [7] P. Devijver and K. J., *Pattern Recognition: A Statistical Approach*. Londres: Prentice-Hall, 1982.
- [8] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Addison-Wesley Longman Publishing Co., Inc., 2008.
- [10] Y. Lecun and C. Cortes, "The mnist database of handwritten digits." [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [11] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, 2008.
- [12] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," *Proceedings of the IEEE International Conference on Neural Networks*, pp. 586–591, 1993.
- [13] Z. Ping and C. Lihui, "A novel feature extraction method and hybrid tree classification for handwritten numeral recognition," *Pattern Recognition Letters*, vol. 23, no. 1-3, pp. 45–56, 2002.
- [14] C. Y. Suen and J. Tan, "Analysis of errors of handwritten digits made by a multitude of classifiers," *Pattern Recognition Letters*, vol. 26, no. 3, pp. 369–379, 2005.