

**Combinação de Classificadores:
Aplicação em projeto multi-níveis de
operadores morfológicos binários**

Nina S. T. Hirata
nina@ime.usp.br

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

Relatório Técnico, Junho de 2007

1 Introdução

Nos dias atuais, a tecnologia disponível permite que uma enorme quantidade de dados, nos mais variados formatos e tipos, seja gerada e armazenada em mídias eletrônicas. Surge então, além da necessidade de organizar, armazenar e manipular eficientemente, a necessidade de também interpretar esses dados.

Diversas abordagens podem ser utilizadas para auxiliar a interpretação de um conjunto de dados, começando por tabelas e gráficos que sumarizam os mesmos de forma a condensá-los em uma quantidade menor de dados. Tais abordagens são comuns na área de estatística e de sistemas gerenciadores de bases de dados.

Outras abordagens mais sofisticadas visam encontrar informações “escondidas” nos dados. Por exemplo, profissionais do mercado preocupam-se em traçar perfis de clientes ou padrões de consumo e explorar esse tipo de informação em campanhas de marketing e para o desenvolvimento de novos tipos de produtos ou serviços. Esse é um exemplo no qual as informações obtidas dos dados dão suporte à tomada de decisões e trata-se de um tópico bastante investigado numa área mais recente conhecida como *Data Mining* (usualmente traduzida como Mineração de Dados).

Em contraposição, abordagens mais clássicas tratam problemas de reconhecimento ou classificação, nos quais os dados são coletados de forma mais controlada. Em sistemas de processamento de documentos, um problema clássico é o reconhecimento de caracteres: imagens de documentos são gerados a partir de dispositivos de imageamento como *scanners* e em seguida os símbolos (ou caracteres) são isolados por técnicas de processamento de imagens; uma vez isolados, os mesmos são classificados (reconhecidos) por um classificador que atribui um rótulo de classe ao qual o símbolo pertence. Diagnóstico médico é um outro problema clássico, no qual a partir de um conjunto de sintomas e dados clínicos tenta-se diagnosticar o estado de saúde do paciente.

Em muitos desses problemas, enquanto os “melhores classificadores” são os especialistas da área na qual o problema está inserido, muitas vezes é possível automatizar parte ou o processo de reconhecimento/classificação. Isto é particularmente desejável quando o processo envolve grande quantidade de dados resultando em um processo repetitivo e cansativo ou quando o processo é realizado em ambientes insalubres ou de difícil acesso.

Seguindo esta breve contextualização, na seção 2 apresentamos definições e conceitos básicos relacionados à classificação de dados, ao mesmo tempo em que introduzimos algumas notações. Em seguida, apresentamos brevemente a noção de combinação de classificadores, com ênfase em uma técnica de combinação conhecida por *stacked generalization*. Na segunda parte do documento descrevemos, na seção 3, operadores morfológicos binários e sua relação com os classificadores. Na terceira parte do texto apresentamos, na seção 4, uma proposta que integra a abordagem *stacked generalization* para projeto de classificadores à abordagem de projeto multi-níveis de operadores morfológicos, seguida pelos resultados experimentais na seção 5. Na parte final do texto, na seção 6, apresentamos as conclusões deste estudo e as perspectivas futuras para este trabalho.

2 Classificadores e Combinação de Classificadores

Nesta seção apresentamos definições e conceitos básicos relacionados à classificação de dados e introduzimos algumas notações necessárias ao longo do texto. Apresentamos também brevemente a noção de combinação de classificadores, com ênfase em uma técnica de combinação conhecida por *stacked generalization*, que serviu de inspiração para o trabalho descrito neste texto.

Nos problemas de classificação têm-se tipicamente um espaço de objetos a serem classificados. No caso de **classificação supervisionada**, sabe-se qual é a classe à qual cada um desses objetos pertence. Para efeitos de formalização do problema de classificação, consideramos que os objetos a serem classificados são representados por um conjunto de n características. Por exemplo, num problema de diagnóstico médico, essas características podem ser dados clínicos como temperatura, pressão arterial, peso, altura, nível de colesterol no sangue, presença/ausência de dores, etc. Os valores dessas características podem ser vistos como um vetor n -dimensional $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$. Deste ponto em diante iremos denominar de objeto o próprio vetor de características \mathbf{x} que representa um objeto.

Seja $X \subseteq \mathbb{R}^n$ o subespaço de todos os possíveis objetos de um dado contexto, a ser denominado de **espaço de características** ou **espaço de exemplos**. Consideramos também que cada objeto pertence a uma classe de um conjunto de classes $Y = \{1, 2, \dots, k\}$. Então, dados um espaço de características X e as classes em Y , um **classificador** é uma função

$$f : X \rightarrow Y \tag{1}$$

que a cada exemplo $\mathbf{x} \in X$ associa uma classe $y = f(\mathbf{x}) \in Y$.

2.1 Treinamento de classificadores

No projeto de classificadores, ou treinamento supervisionado, é dada uma **amostra de treinamento**, ou seja, um conjunto $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ de objetos de X e as respectivas classes. O termo supervisionado refere-se ao fato dos exemplos estarem rotulados, com classificação conhecida, em contraposição aos problemas de classificação não-supervisionada nos quais não se sabe a priori a classificação de nenhum dos objetos. A partir dos dados de treinamento, deseja-se encontrar um classificador f de forma que $f(\mathbf{x}_i) = y_i$ para todos os exemplos do conjunto T e tal que f seja também capaz de “acertar” as classes dos exemplos de X que não aparecem em T .

Na prática, quando adotamos a representação de objetos por vetores de características, diferentes objetos que pertencem a classes distintas podem ser mapeados para um mesmo vetor de características. Assim, na amostra de treinamento T é possível termos $\mathbf{x}_i = \mathbf{x}_j$ com $y_i \neq y_j$, $i \neq j$. Uma vez que um classificador é uma função determinística, nesses casos alguns erros de classificação serão inevitáveis.

Para minimizar o número de erros devido à limitação de representação dos exemplos, deve-se escolher um conjunto de características que constituem uma bijeção entre os objetos a serem classificados e os vetores de características. Tal escolha, no entanto, nem sempre é

possível ou conveniente. Portanto, deste ponto em diante, supomos que o espaço $X \times Y$ é caracterizado por uma distribuição conjunta denotado por $P(X, Y)$.

Definimos o erro de classificação de f relativo a um objeto $\mathbf{x} \in X$ da classe $y \in Y$ através da função

$$e_f(\mathbf{x}, y) = \begin{cases} 1, & \text{se } f(\mathbf{x}) \neq y, \\ 0, & \text{caso contrário.} \end{cases} \quad (2)$$

A **taxa de erro** de um classificador f é dada pela esperança

$$Err[f] = E[e_f(\mathbf{x}, y)] \quad (3)$$

calculada sobre o espaço $X \times Y$. Quando $Y = \{0, 1\}$ a taxa de erro pode ser escrita como

$$Err[f] = E[|f(\mathbf{x}) - y|], \quad (4)$$

a esperança do erro absoluto, comumente denominada de erro absoluto médio (*mean absolute error* ou MAE). Outras medidas de desempenho, como as que associam diferentes pesos aos diferentes tipos de erros (por exemplo, falso positivo e falso negativo em caso de diagnósticos) podem ser elaboradas para os classificadores.

2.1.1 Classificador de Bayes

Suponha que as probabilidades $P(y)$ de cada classe $y \in Y$ são conhecidas. Na ausência de qualquer outra informação, uma escolha natural para a classificação seria atribuir todos os objetos à classe com maior probabilidade. No entanto, se supormos que \mathbf{x} é conhecido (ou seja, foi observado) e que as funções de densidade a priori $p(\mathbf{x}|y)$ dos objetos \mathbf{x} condicionadas a cada classe y são conhecidas, então podemos calcular a probabilidade a posteriori da classe de \mathbf{x} ser y usando a fórmula de Bayes:

$$P(y|\mathbf{x}) = \frac{P(y)p(\mathbf{x}|y)}{p(\mathbf{x})}. \quad (5)$$

Pode-se mostrar que a escolha que minimiza a taxa de erro é atribuir \mathbf{x} à classe com máxima probabilidade a posteriori, ou seja,

$$f(\mathbf{x}) = i \quad \text{onde } i = \arg \max_j \{P(j|\mathbf{x})\}. \quad (6)$$

Esta regra é conhecida por **regra de classificação de Bayes** (veja, por exemplo, [DHS01]).

Na prática essas probabilidades dificilmente são conhecidas. As probabilidades a priori de cada classe y podem ser estimadas com considerável precisão, porém as funções de densidade condicionais $p(\mathbf{x}|y)$ são muito mais difíceis de serem estimadas. Duas abordagens comuns para estimar as funções de densidade condicionais são a paramétrica e a não paramétrica. Na abordagem paramétrica, supõe-se que o tipo de distribuição é conhecida e então os dados de treinamento são utilizados para estimar os parâmetros dessa distribuição (tipicamente a média e o desvio padrão). Na abordagem não-paramétrica, a função densidade é aproximada por densidades calculadas localmente usando funções locais que consideram as amostras observadas numa pequena vizinhança.

2.1.2 Funções discriminantes, regiões e fronteiras de classificação

O classificador de Bayes insere o problema de classificação num contexto probabilístico-estatístico. Podemos mudar o ponto de vista e obter uma visão funcional e geométrica do problema de classificação conforme descrito a seguir.

Consideremos k funções dados por

$$g_i(\mathbf{x}) = P(i|\mathbf{x}), \quad i = 1, 2, \dots, k \quad (7)$$

Usando essas funções, a regra de classificação de Bayes pode ser reescrita como

$$f(\mathbf{x}) = i \iff g_i(x) = \max_j \{g_j(\mathbf{x})\}. \quad (8)$$

De forma semelhante, poderíamos considerar diferentes funções g_i , não necessariamente os dados acima. Essas funções são denominadas **funções discriminantes**, no sentido de que aquele que “responde mais forte” a um objeto \mathbf{x} determina a sua classificação. Note, por exemplo, que qualquer transformação monotônica (como $\log g_i$) das funções g_i dadas acima mantém a relação de ordem entre eles para um determinado objeto \mathbf{x} .

Se considerarmos R_i como sendo o conjunto de objetos de X que são classificados como da classe i , podemos ver que o espaço X é particionado em regiões R_1, R_2, \dots, R_k de forma que $R_i \subseteq X$, $R_i \cap R_j = \emptyset$ se $i \neq j$ e $\cup_{i=1}^k R_i = X$. Note que cada região R_i não necessariamente é conexo. Em algumas situações, pode-se mostrar que as funções discriminantes dadas acima definem regiões convexas, duas a duas separadas por um hiperplano. Porém, em geral as fronteiras entre duas regiões, quando as mesmas são adjacentes, podem formar superfícies complexas. Esses segmentos que formam os pontos de adjacência entre duas regiões são denominadas de **fronteiras de decisão**.

Desta forma, o problema de projetar classificadores pode ser visto com um problema de

1. classificação Bayesiana, na qual deve-se estimar as probabilidades $P(y)$ e as funções de densidades $p(\mathbf{x}|y)$, $y \in Y$; ou
2. encontrar as funções discriminantes adequadas. Note que se soubermos as probabilidades $P(y)$ e as funções de densidades $p(\mathbf{x}|y)$, $y \in Y$, então automaticamente temos essas funções; ou
3. determinar as regiões associadas a cada classe $y \in Y$. Note que se tivermos (1) ou (2), então temos também essas regiões; ou
4. determinar as fronteiras de decisão. Note que, se tivermos (3) então podemos calcular essas fronteiras e, por outro lado, se tivermos as fronteiras, podemos calcular as regiões em (3).

Essas modelagens estão interrelacionadas, no sentido de que uma pode ser obtida da outra ou vice-versa (embora tal derivação possa ser não-trivial). Dependendo dos tipos de dados disponíveis, pode ser mais conveniente olhar o problema de projetar classificadores de um

certo ponto de vista ou outro. Para cada ponto de vista ou combinação de pontos de vista, diferentes questões aparecem. Por exemplo, no caso de determinar as fronteiras de decisão, devemos decidir se as fronteiras serão representadas explicitamente ou implicitamente, ou se serão lineares, quadráticas ou mais complexas, quais algoritmos serão utilizados e assim por diante.

Existem vários algoritmos de classificação que, de alguma forma, representam o classificador final $f : X \rightarrow Y$. Muitas vezes tal representação é implícita como no caso de redes neurais, classificação pelos vizinhos mais próximos, entre outros. Não cabe apresentarmos o processo de treinamento desses algoritmos neste texto. Para maiores informações, o leitor deverá consultar as inúmeras referências disponíveis para o assunto, dentre os quais citamos [DHS01, TK06, JDM00, Mit97, HTF01, Kun04]. A capacidade de um classificador atribuir a classificação correta aos exemplos não observados no conjunto de treinamento T é denominado de **generalização**.

2.2 Combinação de classificadores

Combinação de classificadores é uma prática relativamente antiga mas que passou a receber atenção como um tópico de estudo na área de reconhecimento de padrões e afins na década de 1990. A idéia central em combinação de classificadores é a geração de classificadores baseados em um conjunto de classificadores base e diferentes formas de combinar a classificação de cada um desses classificadores base. Um exemplo muito simples e intuitivo consiste em projetar um conjunto de classificadores a partir de um mesmo conjunto de treinamento, eventualmente variando-se a quantidade de dados de treinamento utilizado ou algum parâmetro do algoritmo de treinamento ou ainda o próprio algoritmo de classificação e associar a cada exemplo a classe mais votada, considerando-se que esses classificadores base são os votantes. A estratégia de combinação pode ser modificada e inúmeras estratégias tem sido propostas.

Neste texto não iremos descrever esse tópico em profundidade. Iremos apenas descrever uma abordagem conhecida por *stacked generalization* que serviu de inspiração para o desenvolvimento deste trabalho. Uma referência bibliográfica atual que apresenta o estado-da-arte sobre combinação de classificadores é o livro de Ludmila Kuncheva [Kun04]. Outras referências importantes são [KR00, KR01, RK02, GHvL03, RKW04, OPKR05, HKR05, Alp98, BK99, Bre96, Die00, AGK98, Gho02, GMS02, HHS94, KHDm98, KA03, Kun02, RF99, TvBDK00].

2.2.1 Stacked generalization

Em uma situação em que múltiplos classificadores são projetados para um mesmo problema de classificação, podemos simplesmente escolher o melhor deles, ou então, combinar as classificações dadas por cada um deles de forma a se determinar a classificação final. Uma forma de combinação bastante natural e bastante conhecida é a escolha majoritária, ou seja, escolher como a classe final aquela que “recebeu mais votos”. *Stacked generalization* é uma técnica de combinação de classificadores baseada em múltiplos níveis de

treinamento, proposto por Wolpert [Wol92b, Wol92a]. A forma de combinação não é uma escolha do projetista, porém é determinada por um processo de treinamento. Cada nível de treinamento considera as classificações fornecidas pelos classificadores de níveis anteriores como dados de treinamento.

Considere uma situação com um conjunto de treinamento T . Suponha que dois classificadores, f_1 e f_2 , são treinados usando o conjunto $T - \{(\mathbf{x}, y)\}$. Ao classificarmos \mathbf{x} usando f_1 e f_2 , teremos $f_1(\mathbf{x}) = y_1$ e $f_2(\mathbf{x}) = y_2$. Como nenhum deles foi treinado utilizando o par (\mathbf{x}, y) , tanto y_1 como y_2 podem ser diferentes de y . No entanto, pelo menos sabemos que quando f_1 classifica o objeto como da classe y_1 e quando f_2 classifica o objeto como da classe y_2 , a classe correta do objeto é y . Essa informação, ou seja, o par $((y_1, y_2), y)$, pode ser visto como um exemplo de treinamento em um novo espaço de exemplos. Considerando-se diferentes partições de T para treinar f_1 e f_2 , pode-se construir esse novo espaço com um maior número de exemplos de treinamento, e esse novo espaço pode ser utilizado para treinar um outro classificador (o classificador de segundo nível). De uma forma geral, o número de classificadores treinados inicialmente pode ser arbitrário e o processo pode ser repetido seqüencialmente, resultando em um classificador de múltiplos níveis.

No artigo original, Wolpert sugere a partição do conjunto T de forma similar ao realizado pelas técnicas de validação cruzada. Porém, as possibilidades de variação em torno dessa abordagem multi-níveis são grandes. Destacamos:

- usar diferentes subconjuntos de T para cada classificador de primeiro nível,
- usar diferentes subconjuntos de características dos exemplos para cada classificador,
- usar diferentes tipos de classificadores,
- usar o mesmo tipo de classificador, porém variar alguns de seus parâmetros.

3 Operadores morfológicos

Denominamos operadores morfológicos uma classe de mapeamentos entre imagens cujo formalismo é baseado na teoria dos reticulados completos. A morfologia matemática, disciplina que estuda esses operadores, foi introduzida na década de 1960 por Georges Matheron e Jean Serra [Mat75, Ser82, Ser88]. Atualmente é uma abordagem bastante utilizada em processamento e análise de imagens [Ser82, Soi03]. Uma característica interessante dos operadores morfológicos é que eles podem, muitas vezes, ser associados a interpretações geométrico-topológicas, que podem ser explorados no projeto desses operadores. Essas interpretações geométrico-topológicas são possíveis pois os operadores morfológicos podem ser obtidos através da combinação de operadores básicos conhecidos por erosão e dilatação. De fato, do ponto de vista formal pode-se mostrar que todos os operadores morfológicos (mapeamentos entre reticulados completos) que sejam invariantes à translação podem ser expressos como o supremo de operadores per-

tencentes a quatro classes de operadores elementares (erosão, dilatação, anti-erosão e anti-dilatação) [Mat75, Mar85, Mar89, BB91, BB93].

Nesta seção descrevemos os operadores morfológicos binários invariantes a translação e localmente definidos e relembramos de que forma os mesmos podem ser vistos como operadores equivalentes à classe de funções booleanas. Mais ainda, veremos que essa equivalência permite encarar o problema de projeto de operadores morfológicos binários como um problema de classificação binária. Em seguida descrevemos um procedimento utilizado para o projeto desses operadores e apresentamos uma limitação desse procedimento que motivou o presente trabalho que será descrito na próxima seção.

3.1 Operadores morfológicos binários

Imagens binárias definidas em $\mathbb{E} = \mathbb{Z}^2$ podem ser representadas por funções binárias do tipo $f : \mathbb{E} \rightarrow \{0, 1\}$. Essas funções podem ser interpretadas como funções indicadoras de subconjuntos de \mathbb{E} . Portanto, o conjunto de todas as imagens binárias definidas no domínio \mathbb{E} pode ser representado pelo conjunto das partes de \mathbb{E} , ou seja, por $\mathcal{P}(\mathbb{E}) = \{S : S \subseteq \mathbb{E}\}$.

Operadores de imagens binárias são mapeamentos de $\mathcal{P}(\mathbb{E})$ em $\mathcal{P}(\mathbb{E})$. A translação de uma imagem S por um vetor z em \mathbb{E} é denotado S_z e definido por $S_z = \{x + z : x \in S\}$, na qual $+$ denota a operação usual de adição de vetores em \mathbb{E} . Dizemos que um operador $\Psi : \mathcal{P}(\mathbb{E}) \rightarrow \mathcal{P}(\mathbb{E})$ é **invariante a translação** se, para todo $S \in \mathcal{P}(\mathbb{E})$ e $z \in \mathbb{E}$, $[\Psi(S)]_z = \Psi(S_z)$, ou seja, o resultado ao se transladar uma imagem e aplicar o operador é o mesmo obtido ao se aplicar o operador à imagem e em seguida transladar o resultado.

O conjunto $\mathcal{P}(\mathbb{E})$ munido com a relação de ordem parcial \subseteq (a relação usual de inclusão de conjuntos) forma um reticulado completo. Logo, todos os resultados conhecidos sobre mapeamentos entre reticulados completos são válidos para os operadores de imagens binárias. Em particular, um resultado importante para os operadores binários invariantes a translação é o teorema da decomposição canônica. De acordo com esse teorema, qualquer operador $\Psi : \mathcal{P}(\mathbb{E}) \rightarrow \mathcal{P}(\mathbb{E})$, invariante à translação, pode ser expresso em termos de operadores básicos conhecidos como operadores sup-geradores (ou operadores intervalo).

O operador sup-gerador é caracterizado por dois subconjuntos A e B , $A \subseteq B \subseteq \mathbb{E}$, sendo definido por

$$\Lambda_{(A,B)}(S) = \{x \in \mathbb{E} : A_x \subseteq S \subseteq B_x\}. \quad (9)$$

Uma vez que os parâmetros A e B definem um intervalo ($[A, B] = \{X \subseteq \mathbb{E} : A \subseteq X \subseteq B\}$), esse operador é conhecido também por operador intervalo. Em outras palavras, ele pode ser usado para detectar todas as imagens binárias que contêm A e estão contidas em B . Pode-se ainda mostrar que os operadores sup-geradores podem ser expressos em termos de erosões e dilatações (dois operadores elementares da morfologia matemática) [BB91].

O núcleo de um operador $\Psi : \mathcal{P}(\mathbb{E}) \rightarrow \mathcal{P}(\mathbb{E})$ é definido por

$$\mathcal{K}(\Psi) = \{S \subseteq \mathcal{P}(\mathbb{E}) : o \in \Psi(S)\} \quad (10)$$

na qual o denota a origem de \mathbb{E} . A decomposição canônica de Ψ é dada por

$$\Psi = \bigvee \{ \Lambda_{(A,B)} : [A, B] \subseteq \mathcal{K}(\Psi) \} \quad (11)$$

Esta decomposição é redundante pois se $[A_1, B_1] \subseteq [A_2, B_2]$ então teremos $\Lambda_{(A_1, B_1)} \subseteq \Lambda_{(A_2, B_2)}$ e consequentemente $\Lambda_{(A_1, B_1)} \vee \Lambda_{(A_2, B_2)} = \Lambda_{(A_2, B_2)}$. A decomposição acima pode ser simplificada para

$$\Psi = \bigvee \{ \Lambda_{(A,A)} : A \in \mathcal{K}(\Psi) \} \quad (12)$$

Uma decomposição canônica minimal é obtida se consideramos apenas os intervalos maximais contidos em $\mathcal{K}(\Psi)$. O conjunto desses intervalos maximais é denominado base de Ψ e denotado $\mathcal{B}(\Psi)$. A decomposição de Ψ em termos de sua base é dada por:

$$\Psi = \bigvee \{ \Lambda_{(A,B)} : [A, B] \in \mathcal{B}(\Psi) \} \quad (13)$$

Essa decomposição pode também ser redundante no sentido de a união de dois ou mais intervalos conter algum outro, fazendo deste último redundante no cálculo do supremo. A decomposição em termos de núcleo e em termos de base foram introduzidas inicialmente para a classe dos operadores crescentes respectivamente por Matheron [Mat75] e Maragos [Mar85, Mar89], e foram generalizadas para operadores não necessariamente crescentes por Banon e Barrera [BB91, BB93].

3.2 Relação com classificadores

Além da invariância à translação, outra propriedade sobre os operadores que pode ser considerada é a definição local [Hei94]. Seja $W \subseteq \mathbb{E}$ um conjunto não vazio a ser denominado **janela**. Dizemos que um operador $\Psi : \mathcal{P}(\mathbb{E}) \rightarrow \mathcal{P}(\mathbb{E})$ é **localmente definido (l.d.) por W** se, para todo $z \in \mathbb{E}$ e $S \subseteq \mathbb{E}$,

$$z \in \Psi(S) \iff z \in \Psi(S \cap W_z). \quad (14)$$

Um operador l.d. Ψ pode ser localmente caracterizado por uma função $\psi : \mathcal{P}(W) \rightarrow \{0, 1\}$, da seguinte forma:

$$z \in \Psi(S) \iff \psi(S_{-z} \cap W) = 1, \quad (15)$$

na qual a translação por $-z$ é apenas para garantir que o argumento da função ψ é um subconjunto de W . Mais ainda, a representação via núcleo pode ser simplificada para

$$\Psi = \bigvee \{ \Lambda_{(A,B)} : [A, B] \subseteq \mathcal{K}_W(\Psi) \} \quad (16)$$

na qual $\mathcal{K}_W(\Psi) = \{X \subseteq \mathcal{P}(W) : o \in \Psi(X)\}$. Da mesma forma, os elementos da base podem ser restritos aos intervalos em $\mathcal{P}(W)$. Operadores i.t. e l.d. são denominados **W -operadores**.

Se associarmos uma variável binária x_i para cada elemento $w_i \in W$ e considerarmos $x_i = 1 \iff w_i \in S_{-z} \cap W$, então ψ pode ser encarado como uma função booleana em $n = |W|$ variáveis.

Note que ao fixarmos um determinado W , restringimos o espaço de operadores para o conjunto daqueles que são localmente definidos por W . Porém, a modelagem por definição local não impõe restrição nenhuma sobre o espaço de operadores i.t., uma vez que se considerarmos $W = \mathbb{E}$, o espaço de W -operadores é exatamente igual ao espaço dos operadores i.t. A vantagem da formulação de definição local é que, em se tratando de operadores i.t. (localmente definidos ou não), podemos sempre caracterizá-lo via uma função binária.

Desta forma, o projeto de operadores de imagens binárias pode ser reduzido ao problema de projeto dessas funções características. Mais precisamente, o projeto de W -operadores pode ser visto como um problema de projeto de funções do tipo $\psi : \{0, 1\}^{|W|} \rightarrow \{0, 1\}$. Os elementos $\mathbf{x} \subseteq \mathcal{P}(W)$ tal que $\psi(\mathbf{x}) = 1$ são os elementos de $\mathcal{K}_W(\Psi)$ enquanto aqueles tais que $\psi(\mathbf{x}) = 0$ não fazem parte de $\mathcal{K}_W(\Psi)$. Em outras palavras, projetar W -operadores pode ser visto como um problema de classificar elementos de $\mathcal{P}(W)$ como sendo ou não elementos do núcleo do operador. A função que realiza esta classificação corresponde justamente à função característica do operador.

3.3 Treinamento de operadores morfológicos binários

Conforme visto acima, o problema de projetar W -operadores pode ser visto como um problema de determinar a função booleana que caracteriza um operador. Para colocar o problema de projeto de operadores no contexto de classificação supervisionada, consideramos pares de imagens (S_i, I_i) , realizações de processos aleatórios conjuntamente estacionários (\mathbf{S}, \mathbf{I}) , no qual \mathbf{S} corresponde ao processo que gera as imagens de entrada (imagens a serem processadas) e \mathbf{I} ao correspondente processo gerador de imagens ideais (imagens desejadas como resultados do processamento). O objetivo é encontrarmos um W -operador Ψ tal que $\Psi(\mathbf{S})$ seja estatisticamente o mais próximo de \mathbf{I} . Essa proximidade pode ser caracterizada por alguma medida de erro. Neste trabalho consideramos o erro absoluto médio (MAE de *mean absolute error*), definido como segue:

$$MAE\langle\Psi\rangle = E[|\Psi(\mathbf{S})(z) - \mathbf{I}(z)|] \quad (17)$$

na qual $E[\cdot]$ denota a esperança e $\Psi(\mathbf{S})(z)$ e $\mathbf{I}(z)$ denotam as variáveis aleatórias binárias que correspondem aos valores observados no ponto z nas imagens de entrada e saída, respectivamente.

As observações do tipo $S_{-z} \cap W$ em um ponto arbitrário z podem ser vistos como realizações de um vetor aleatório $\mathbf{X}_z = (x_1, x_2, \dots, x_n)$, com $x_i = 1 \iff w_i \in S_{-z} \cap W$. Similarmente, os valores de \mathbf{I} em um ponto arbitrário z podem ser encarados como realizações de uma variável aleatória binária \mathbf{y}_z . Devido à estacionaridade, podemos desconsiderar a localização z , e considerar um vetor aleatório $\mathbf{X} = (x_1, x_2, \dots, x_n)$ cujas realizações são observações do tipo $S_{-z} \cap W$ em pontos arbitrários z , e uma variável aleatória binária \mathbf{y} . Em resumo, a distribuição conjunta de (\mathbf{S}, \mathbf{I}) induz uma distribuição conjunta $P(\mathbf{X}, \mathbf{y})$ de (\mathbf{X}, \mathbf{y}) .

A partir dessas considerações definimos o W -operador ótimo com respeito ao processo

(\mathbf{S}, \mathbf{I}) como sendo aquele que minimiza

$$MAE\langle\Psi\rangle = E[|\psi(\mathbf{X}) - \mathbf{y}|]. \quad (18)$$

Expandindo a expressão acima temos

$$\begin{aligned} MAE\langle\Psi\rangle &= \sum_{(\mathbf{x}, y)} |\psi(\mathbf{x}) - y| \\ &= \sum_{\mathbf{x}} \psi(\mathbf{x})P(\mathbf{x}, 0) + |\psi(\mathbf{x}) - 1|P(\mathbf{x}, 1) \\ &= \sum_{\{\mathbf{x}:\psi(\mathbf{x})=0\}} P(\mathbf{x})P(\mathbf{x}, 1) + \sum_{\{\mathbf{x}:\psi(\mathbf{x})=1\}} P(\mathbf{x})P(\mathbf{x}, 0) \end{aligned}$$

Logo, para minimizar o MAE, devemos escolher

$$\psi(\mathbf{x}) = \begin{cases} 1, & \text{se } P(\mathbf{x}, 1) > P(\mathbf{x}, 0), \\ 0, & \text{se } P(\mathbf{x}, 0) > P(\mathbf{x}, 1), \\ 0 \text{ or } 1 & \text{se } P(\mathbf{x}, 0) = P(\mathbf{x}, 1). \end{cases} \quad (19)$$

Isto quer dizer que se a probabilidade conjunta de (\mathbf{X}, \mathbf{y}) for conhecida, podemos determinar o W -operador que minimiza o MAE. No entanto, na prática, essa probabilidade é raramente conhecida. Portanto, os dados de treinamento podem ser utilizados para estimar essa probabilidade. Mesmo em situações em que a quantidade de dados de treinamento é relativamente grande, para janelas de tamanho moderadamente grande (por exemplo 5×5), dificilmente todos os padrões $\mathbf{x} \subseteq W$ serão observados nos dados de treinamento. Para definir a classificação desses padrões não observados, uma possibilidade é utilizar algoritmos de classificação.

A seguir descrevemos o procedimento de treinamento utilizado neste trabalho. Dados um conjunto de imagens de treinamento $\{(S_i, I_i) : i = 1, 2, \dots, m\}$ e uma janela W , o procedimento de treinamento do operador consiste dos seguintes passos:

1. Estimar as probabilidades conjuntas $P(\mathbf{x}, y)$, para $\mathbf{x} \subseteq W$ e $y \in \{0, 1\}$: “percorrer” com a janela W sobre todos os pontos de cada imagem de entrada S_i e, desde que a janela esteja inteiramente sobre a imagem, realizar a contagem dos padrões $\mathbf{x} \subseteq W$ observados na imagem S_i e dos respectivos valores nas correspondentes imagens de saída I_i . Esta contagem gera uma estimativa $\hat{P}(\mathbf{x}, y)$.
2. Decidir a classificação de cada padrão observado: fazer $\hat{\psi}(\mathbf{x}) = 1$ se $\hat{P}(\mathbf{x}, 1) > \hat{P}(\mathbf{x}, 0)$ e $\hat{\psi}(\mathbf{x}) = 0$ em caso contrário. Observe que, do ponto de vista estatístico, para os padrões em que $P(\mathbf{x}, 1) = P(\mathbf{x}, 0)$ o valor de $\psi(\mathbf{x})$ poderia ser tanto 0 como 1.
3. Generalização: após o passo 2, padrões que não foram observados em (1) não têm classificação definida. Para atribuir uma classificação aos padrões não observados, utiliza-se algum algoritmo de generalização. Em particular, utilizamos minimização

de funções booleanas incompletamente especificadas [HP93] como algoritmo de generalização. O algoritmo de minimização de funções booleanas incompletamente especificadas mantém o valor para os padrões observados conforme estabelecido em (2) e atribui aos padrões não-observados (*don't cares* na terminologia de teoria de chaveamentos) ou valor 0 ou valor 1. Neste trabalho em específico, utilizamos o algoritmo descrito em [HBTD02].

O passo (2) pode ser dispensado desde que o algoritmo de generalização do passo (3) seja capaz de lidar com exemplos com rótulos conflitantes. No passo (3), em princípio, qualquer algoritmo de generalização poderia ser utilizado. A escolha pelo algoritmo de minimização de funções booleanas justifica-se pelo fato de que o resultado da minimização pode ser mapeado diretamente ao operador correspondente numa expressão na forma supremo de sup-geradores (correspondendo a uma representação próxima àquela baseada na base, descrita acima).

Exemplo: A figura 1(a) mostra um par de exemplos de treinamento. Essas imagens são utilizadas para se estimar as probabilidades conjuntas $P(\mathbf{x}, y)$. Após a estimação (passo

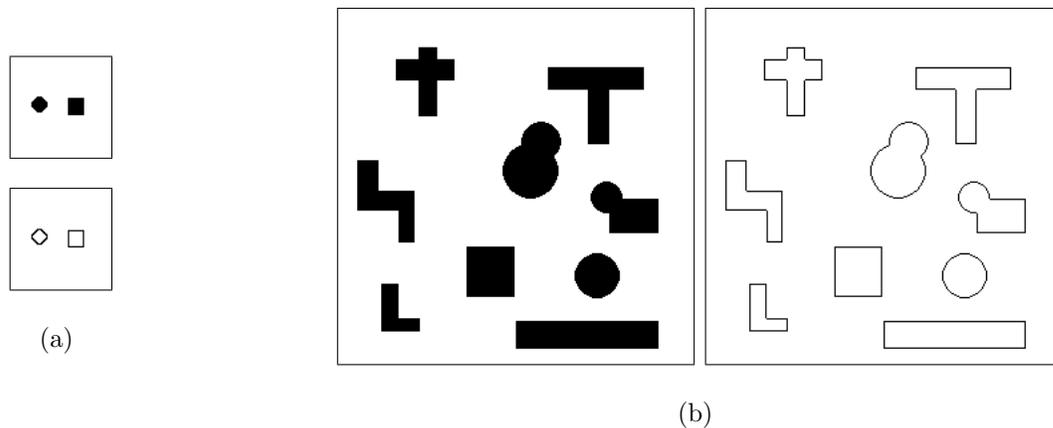


Figura 1: Learning binary image operators: (a) training images and (b) test images.

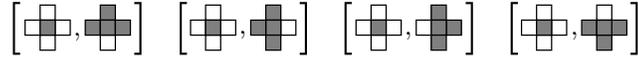
1 acima) seguida da decisão (passo 2), os elementos para os quais $\hat{\psi}(\mathbf{x}) = 1$ (elementos do núcleo) são



enquanto aqueles para os quais $\hat{\psi}(\mathbf{x}) = 0$ são



Após a aplicação do passo 3 (minimização de função booleana incompletamente especificada), o conjunto de intervalos resultantes é



A figura 1(b) mostra uma imagem de teste e o respectivo resultado obtido ao se aplicar o operador representado pelo conjunto de intervalos acima. Os intervalos obtidos correspondem à base do operador para extração de borda interna, 8-conectado. Outros exemplos de aplicação podem ser vistos por exemplo em [Hir07].

3.4 O problema de overfitting

Com relação ao projeto de operadores a partir de exemplos, resultados experimentais mostram que soluções satisfatórias podem ser obtidas para diferentes problemas de processamento de imagens binárias. Porém, há problemas nos quais a utilização de janelas maiores é condição necessária para alguma melhora significativa nos resultados. No entanto, na prática, como o número de dados de treinamento é limitado, ao se aumentar indiscriminadamente o tamanho da janela deparamos com o fenômeno conhecido por *overfitting*. Este fenômeno é entendido como o ajuste demasiado do classificador aos dados de treinamento. Uma vez que, quanto maior o espaço de exemplos, menos representativo torna-se uma amostra de tamanho fixo, pior tende a ser o desempenho do classificador para os dados de teste. De fato, no caso de operadores de imagens, para uma quantidade fixa de dados de treinamento observa-se que ao se aumentar gradativamente o tamanho da janela, começando-se por uma janela pequena (por exemplo 3×3), o erro MAE estimado nos dados de teste decresce rapidamente no início, atinge um ponto de mínimo e em seguida começa a crescer gradualmente. Veja a figura 2.

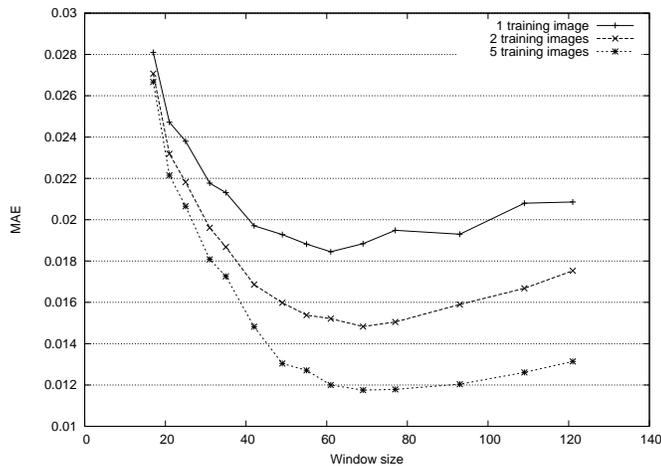


Figura 2: O fenômeno *overfitting* no projeto de operadores binários.

Para janelas pequenas, o erro MAE elevado é devido a inabilidade da janela distinguir padrões maiores, resultando em padrões com classificações conflitantes. Já para as janelas maiores, o erro MAE elevado é devido a esse fenômeno de overfitting. Esse fenômeno relaciona-se também com o fenômeno conhecido por maldição da dimensionalidade (*curse of dimensionality*).

Do ponto de vista teórico pode-se facilmente mostrar que o MAE do W -operador ótimo é menor ou igual ao MAE de um W' -operador ótimo, para qualquer $W' \subset W$. Na prática, porém, como a quantidade de dados de treinamento é limitada, ocorre o fenômeno *overfitting*, o que implica que há uma janela ótima – não necessariamente a maior de todas – que resulta no operador com menor MAE.

Portanto, a questão a que chegamos é se já atingimos o final de uma linha (i.e., não há mais o que se fazer de significativo), ou se é possível, a partir do mesmo conjunto de dados de treinamento, projetar operadores com desempenho melhor do que aquele que corresponde ao ponto mínimo na curva mostrada acima. Na próxima seção será apresentada a abordagem que propomos para isso.

4 Treinamento multi-níveis de operadores morfológicos

Se diversos classificadores forem projetados para um mesmo problema de classificação, é razoável esperarmos que eles difiram entre si na classificação de alguns padrões. Portanto, a idéia de combinar o resultado de diversos classificadores, de forma a obter uma classificação que seja uma espécie de consenso, é uma abordagem natural.

No caso de operadores de imagens vale também o mesmo raciocínio. A abordagem proposta aqui foi inspirada no *stacked generalization*, descrito anteriormente. Em essência, a idéia consiste em projetarmos diferentes operadores baseados em diferentes janelas. Em seguida, a forma de combinar os resultados desses operadores é determinada utilizando-se novamente um procedimento de treinamento.

Para formalizar esse modelo, apresentamos inicialmente um exemplo com dois níveis de treinamento. Os operadores de nível 1 são denotados por $\Psi_i^{(1)}$, $i = 1, 2, \dots, n_1$, enquanto o de nível dois é denotado $\Psi_1^{(2)}$. O operador resultante é uma composição dos operadores de nível 1 com o operador de nível 2, ou seja, $\Psi = \Psi_1^{(2)}(\Psi_1^{(1)}, \Psi_2^{(1)}, \dots, \Psi_{n_1}^{(1)})$. Uma vez que o operador de nível 2 toma como entrada um pixel da saída de cada um dos operadores de nível 1, o operador final, indiretamente, tem como entrada a união das entradas de cada um dos operadores de nível 1. Desta forma, a janela associada ao operador final é maior ou igual às janelas associadas a cada um dos operadores de nível 1. Esse esquema é ilustrado na figura 3.

O exemplo apresentado acima para um operador de dois níveis pode ser facilmente generalizado para um número arbitrário de níveis. Além disso, um operador não precisa necessariamente tomar como entrada apenas um pixel das imagens resultantes de operadores de níveis inferiores. Mais do que isso, as imagens iniciais, entradas para os operadores de nível 1, não precisam provir de um único processo aleatório.

Um esquema mais geral, porém para três níveis, é ilustrado na figura 4. Como pode ser visto, um operador pode ter como parte de sua entrada observações provindas de diferentes imagens, iniciais ou resultantes de operadores de níveis inferiores. O esquema da figura considera apenas um processo inicial, porém como já mencionado, poderiam ser considerados mais processos.

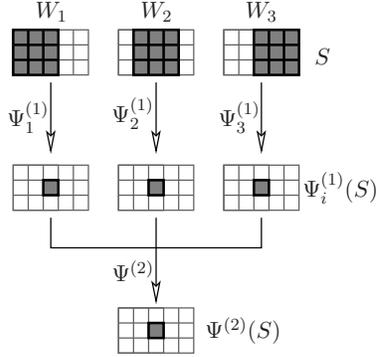


Figura 3: Um operador de dois níveis. Os operadores de nível 1, $\Psi_1^{(1)}$, $\Psi_2^{(1)}$ e $\Psi_3^{(1)}$ são baseados nas janelas W_1 , W_2 e W_3 , respectivamente. Embora ilustrados separadamente, a imagem de entrada para os três operadores de nível 1 é a mesma; o que difere é a vizinhança considerada (realçada em cinza-escuro) por cada um deles. O operador de nível 2, $\Psi^{(2)}$, toma como entrada o padrão formado por três pixels, um de cada imagem de saída $\Psi_i^{(1)}(S)$, $i = 1, 2, 3$ (realçado em cinza-escuro).

4.1 Relação com *stacked generalization* e projeto iterativo

A idéia central do modelo proposto acima é basicamente a mesma do *stacked generalization* descrito acima. Apesar dessa semelhança, há uma diferença relevante. Classificadores em geral geram apenas uma saída que é a classe a ser associada ao padrão. Desta forma, quando combinamos as saídas de classificadores de níveis anteriores obtemos padrões em um novo espaço de exemplos com dimensão exatamente igual ao do número de classificadores considerados na combinação. Com as imagens, apesar da função característica (função booleana no caso de imagens binárias) também mapear um padrão para um único pixel de saída, uma vez que a saída de um operador é uma imagem (e não um valor), informações de pixels vizinhos nessa imagem podem ser consideradas para a formação do padrão no novo espaço de exemplos.

Isto possibilita que padrões mais complexos possam ser considerados nos operadores de segundo nível ou superiores.

O treinamento iterativo [HDB00a], que considera treinamentos sequenciais de operadores que visam refinar sucessivamente o resultado do operador anterior na seqüência pode ser visto como um caso particular. No treinamento iterativo, dados pares de imagens de treinamento na forma (S_i, I_i) , no primeiro nível de treinamento um $W^{(1)}$ -operador $\Psi^{(1)}$ é obtido de forma a minimizar $E[\Psi^{(1)}(\mathbf{S}), \mathbf{I}]$, o MAE entre a imagem transformada e a correspondente imagem ideal. No segundo nível de treinamento, pares da forma $(\Psi^{(1)}(S_j), I_j)$ são considerados para treinamento. Após k níveis de treinamento, o operador final consistirá da composição $\Psi(S) = \Psi^{(k)}(\Psi^{(k-1)}(\dots(\Psi^{(2)}(\Psi^{(1)}(S)))\dots))$.

No modelo proposto, o treinamento iterativo corresponde ao caso no qual há exatamente um operador por nível de treinamento, porém a janela associada a esses operadores (mesmo em níveis superiores ao primeiro) consiste de mais de um ponto. Para três níveis de treinamento iterativo, o esquema da figura 4 reduz-se ao mostrado na figura 5.

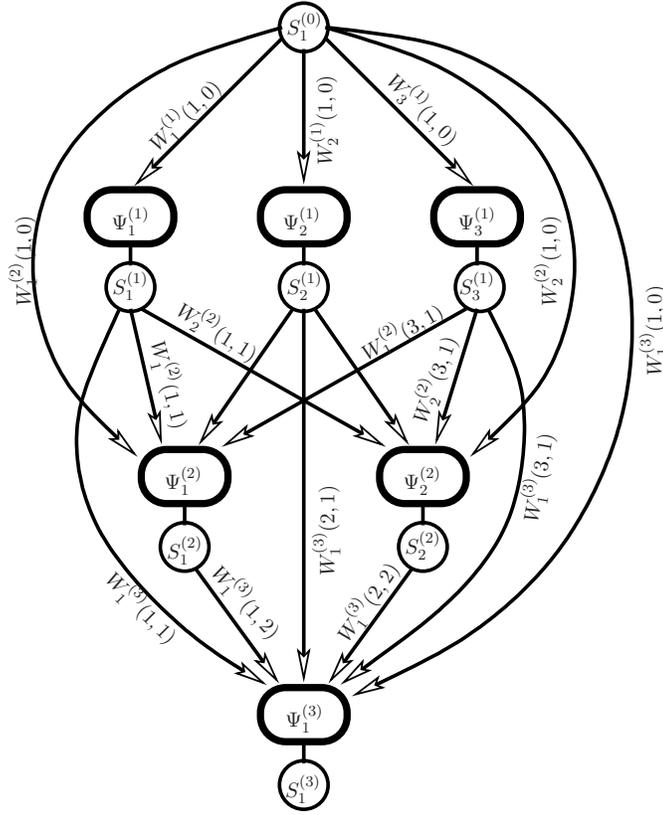


Figura 4: Uma arquitetura de um operador de três níveis: um processo de entrada, 3 operadores de nível 1, 2 operadores de nível 2 e um operador de nível 3. A saída do operador multi-níveis é representado pelo processo $S_1^{(3)}$. Neste modelo, todos os operadores podem receber parte da entrada original e das saídas dos operadores de níveis inferiores como entrada.

5 Implementações e resultados experimentais

5.1 Implementações

O software para treinamento de operadores morfológicos está implementado como uma toolbox no ambiente Khoros [KR94]. O Khoros é um sistema para desenvolvimento de aplicativos para processamento e análise de imagens. Em particular, possui um programa chamado *craftsman*, usado para gerenciar a criação de toolboxes, e uma interface de programação gráfica chamada *cantata*, para compor um fluxo de processamento utilizando

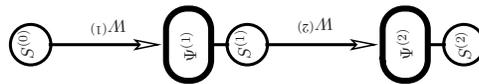


Figura 5: Projeto iterativo de operadores morfológicos, visto como um caso particular da abordagem multi-níveis proposta.

as funções disponíveis nas toolboxes. Existem várias toolboxes disponíveis, principalmente relacionadas à manipulação de dados como visualização, entrada/saída, transformações simples (rotação, mudança de escala, negação, etc). O ambiente permite, de uma forma simples, que uma toolbox seja criada e agregada ao sistema.

A toolbox de projeto de operadores é denominado PAC (de *Probably Approximately Correct*). Trata-se de um trabalho que teve início no ano de 1994, numa versão anterior do sistema Khoros e inicialmente era restrito às imagens binárias [Tom96]. Porém, ao longo dos anos, além de ser adaptado para a versão 2.2 do Khoros¹, ajustes foram realizados em sua estrutura como um todo para conferir-lhe modularidade e possibilidade de expansão para tratar também imagens em tons de cinza e imagens multi-bandas [HDB00b, Hir01, HBFL99], assim como tratamento multi-resolução de imagens [HBBD02].

Neste documento apresentamos uma visão geral deste software no que refere a sua parte relacionada ao tratamento de imagens binárias. Em seguida apresentamos a expansão realizada para conferir-lhe capacidade para o projeto multi-níveis.

A parte correspondente ao projeto de operadores morfológicos binários é composto pelos seguintes módulos:

- **kcollecB**: responsável pela estimação das probabilidades conjuntas $P(\mathbf{x}, y)$. As entradas deste módulo são um arquivo com a relação de imagens de treinamento, um arquivo com a descrição da janela, e alguns parâmetros opcionais. Uma árvore balanceada, na qual cada nó armazena um padrão coletado com as frequências das respectivas saídas, é utilizada como estrutura de dados. A saída é um arquivo texto que contém o número total de padrões observados, o número total de padrões distintos observados, a janela considerada, e para cada padrão observado, a frequência de vezes em que o mesmo apareceu associado à classe 0 e à classe 1.
- **kdecisionB**: responsável pela decisão de qual classe será associada a cada padrão observado. Além da decisão para minimizar MAE, existem outras opções como ignorar padrões com pouca frequência ou associar classe 0 aos padrões com classificação conflitante. A entrada deste módulo é um arquivo resultante do módulo anterior mais a especificação das opções descritas acima. A saída é um arquivo contendo o número de padrões distintos, a janela considerada, e para cada padrão observado a classe escolhida pela decisão realizada.
- **kisi**: responsável pela generalização da classificação. Trata-se da implementação do algoritmo ISI, conforme descrito em [Hir00, HBTD02]. A entrada deste módulo é um arquivo no formato de saída do módulo anterior e a saída é um arquivo contendo os intervalos (termos produtos) que compõem a forma minimizada calculada pelo algoritmo ISI (denominado operador).

¹O sistema Khoros foi adquirido pela empresa Accusoft e atualmente é comercializada pelo nome de VisiQuest (<http://www.accusoft.com/products/visiquest/overview.asp>). A versão 2.2 do Khoros que utilizamos em nosso desenvolvimento não encontra-se mais disponível na web. Por essa razão, atualmente estamos migrando a toolbox PAC para o ambiente Phyton.

- **kapplicB**: responsável pela aplicação de uma função booleana (operador), dado em um arquivo no formato da saída do módulo anterior, sobre uma imagem. As entradas deste módulo são um arquivo de imagem, o arquivo com a descrição de um operador (em termos de intervalos), e alguns parâmetros opcionais (como uma segunda imagem que indica sobre quais pontos da imagem de entrada o operador deve ser aplicado).
- **kanalysis**: responsável por calcular a diferença entre duas imagens. É utilizado para estimar o MAE empírico entre as imagens de teste processadas pelo operador projetado e as respectivas imagens ideais. A saída deste módulo é um arquivo que contém o número de pontos na imagem, o número de pontos que foram efetivamente considerados para o cálculo da diferença (pois pontos próximos à borda da imagem são ignorados), número total de pontos com valores distintos entre as duas imagens em números absolutos e relativos.

Os cinco módulos acima formam o núcleo do processo de treinamento e avaliação de operadores morfológicos binários. Além desses, existem ainda outros módulos auxiliares que não serão descritos aqui.

5.2 Extensão do software

Para realizar a parte experimental do projeto multi-níveis de operadores morfológicos binários, foram criados dois módulos adicionais.

- **kcollec2**: responsável pela coleta de dados dos operadores de nível 2 ou superiores. A diferença deste em relação ao **kcollecB** é que neste último, para coleta de padrões dos operadores de nível 1, os padrões provêm de uma única imagem, enquanto que no primeiro os padrões correspondem à composição de partes que provêm de diferentes imagens. Desta forma, a entrada deste módulo consiste de um arquivo que descreve n -uplas de imagens (imagens das quais partes serão coletadas para formar um padrão) e as respectivas janelas (que definem exatamente quais partes de cada imagem serão consideradas). A saída deste módulo tem o mesmo formato da saída do **kcollecB**.
- **kapplic2**: responsável pela aplicação de um operador de nível 2 ou superior. Esses operadores devem ser aplicados sobre as imagens resultantes dos operadores de níveis anteriores. Assim, a entrada deste módulo consiste de um arquivo similar ao do **kcollec2**, contendo n -uplas com o nome das imagens resultantes dos operadores de níveis anteriores, as janelas associadas a cada um deles, o arquivo com a base do operador e o nome do arquivo no qual deverá ser armazenado a imagem resultante.

Note que o treinamento (decisão e generalização) do operador de segundo nível é realizado utilizando-se **kdecisionB** e **kisi**, como nos operadores de primeiro nível.

Dado um conjunto de imagens de treinamento, os seguintes passos são executados para se projetar um operador multi-níveis:

- Decidir a arquitetura de treinamento (número de níveis, número de operadores em cada nível, janelas associadas a cada operador).
- Decidir quais imagens serão utilizadas em quais partes da arquitetura
- Criação dos arquivos de entrada dos módulos envolvidos (arquivo com a descrição da janelas e arquivos com a lista das imagens de treinamento)
- Criação de uma arquivo script com a seqüência de chamadas aos módulos. Para dois níveis de treinamento, esse script possui a seguinte estrutura:
 1. Para cada operador de nível 1, $\Psi_i^{(1)}$, e respectiva janela $W_i^{(1)}$, executar a seqüência
 - (a) **kcollecB**
 - (b) **kdecisionB**
 - (c) **kisi**
 2. Aplicar cada um desses operadores para cada imagem a ser usada no operador de segundo nível

Para $i=1$ até n_1

para cada imagem $S_j^{(1)}$ a ser usada no treinamento nível 2

calcular $\Psi_i(S_j^{(1)})$
 3. Treinar operador de nível 2, executando a seqüência
 - (a) **kcollec2**
 - (b) **kdecisionB**
 - (c) **kisi**

5.3 Resultados experimentais

O modelo descrito na seção anterior é bastante flexível e permite variações na arquitetura do treinamento. As possíveis escolhas nessa arquitetura estão relacionadas com os seguintes parâmetros:

- Níveis de treinamento
- Número de operadores em cada nível
- As entradas de cada operador (formato e tamanho das janelas e as imagens de treinamento)

O número de possíveis variações é combinatório e portanto não é possível experimentar todas elas. Neste trabalho restringimos o número de níveis a dois. Além disso, as entradas para os operadores de primeiro nível são obtidas de um mesmo processo, variando apenas o tamanho e formato das janelas. Algumas variações no número de imagens de

treinamento foram realizadas, porém nenhum estudo sistemático foi realizado e, portanto, não apresentamos nenhum resultado referente a esse aspecto.

Vários conjuntos de imagens binárias, de diferentes características, foram utilizados nos experimentos. O principal objetivo destes experimentos é confrontar os operadores de um nível contra os de dois níveis. Para cada problema usamos uma janela W para o operador de um nível e um número k de subjanelas de W para os operadores de primeiro nível do operador de dois níveis. Na medida do possível, escolhemos uma janela W que sabemos, por experimentos prévios, ser adequada para o problema em questão.

Para cada tipo de problema, fixamos um número de imagens de treinamento. Os operadores de um nível foram obtidos usando-se todo o conjunto de imagens de treinamento, enquanto para os de dois níveis as imagens de treinamento foram divididas em dois sub-grupos, um para treinar os operadores de nível 1 e outro para os operadores de nível 2. Para ambos os casos, o mesmo conjunto de imagens de teste foram utilizados.

A tabela 1 descreve os conjuntos de dados usados na parte experimental deste trabalho. A primeira coluna descreve o tipo das imagens e o objetivo do processamento; a segunda coluna lista o número total de imagens de treinamento e a forma como foram distribuídas entre os treinamentos de nível 1 e 2 (por exemplo, na primeira linha da tabela, 8(5 : 3) significa que foram utilizados no total 8 pares de imagens de treinamento, dos quais 5 no primeiro nível e 3 no segundo nível); a terceira coluna indica quantas imagens foram utilizadas para testar os operadores projetados. As imagens utilizadas não são necessariamente de mesmo tamanho, porém as imagens de um grupo foram obtidas de um mesmo contexto, usando procedimento de aquisição comum (parâmetros de digitalização, parâmetros de binarização, etc). O número de dados utilizados em termos de pixels é descrito mais adiante.

Descrição	# Imagens de	
	treinamento	teste
A. Diagramas funcionais (segmentação de objetos circulares)	8 (5:3)	10
A'. Diagramas funcionais (segmentação de retângulos tracejados)	8 (5:3)	10
A''. Diagramas funcionais (segmentação de caracteres)	8 (5:3)	10
B. Segmentação de textura	3 (2:1)	2
C. Segmentação de caracter	10 (6:4)	10
D. Segmentação de texto (páginas de revista)	5 (3:2)	5
E. Segmentação de texto (páginas de livro)	5 (3:2)	5
F. Filtragem de ruído booleano	5 (3:2)	5
G. Filtragem de ruído de borda	5 (3:2)	5

Tabela 1: Conjunto de dados utilizados para treinamento e teste de operadores.

Um resumo dos resultados obtidos para diferentes conjuntos de imagens é apresentado na tabela 2. Cada experimento é identificado de acordo com o conjunto de dados utilizado. O experimento C_1 e C_2 (D_1 e D_2) referem-se ao conjunto C (D), e os índices 1 e 2 indicam experimentos com janelas distintas. As respectivas subjanelas usadas nos operadores de nível 1 são mostrados na figura 6. Colunas não preenchidas correspondem aos expe-

Exp.	W	# pixels		Um nível		Dois níveis	
		Treino	Teste	Tempo treino	MAE	Tempo treino	MAE
A	9×9	85967	102883	2700.87	0.015289	1288.16	0.007562
A'	25×25	85940	102883	-	-	419.6	0.025126
A''	9×9	85967	102883	12615.14	0.059747	6883.18	0.039773
B	9×9	230533	104399	677476.1	0.074004	169.28	0.043947
C ₁	9×7	193445	197458	206.8	0.009070	194.65	0.005520
C ₂	11×9	193319	197474	33190.19	0.101128	838.57	0.004254
D ₁	9×7	1049540	783834	45518.32	0.039959	20887.52	0.046352
D ₂	11×11	1047219	783834	-	-	62118.15	0.030531
E	11×11	176368	493755	-	-	48760.22	0.004462
F	9×9	1270080	1260020	9828.14	0.006271	662.17	0.003467
G	7×7	1030224	1030224	2631.65	0.004033	8547.80	0.003492

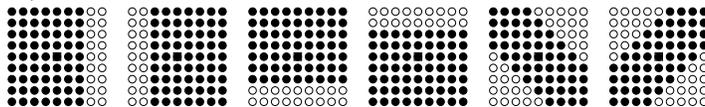
Tabela 2: Tempo de treinamento e MAE empírico para os diferentes experimentos (MAE médio sobre o total de imagens de teste).

rimentos cujo tempo de treinamento estavam excessivamente elevados e cuja execução foram abortadas. Tanto o tempo de treinamento como o MAE são melhores para os operadores de dois níveis. A única exceção é o tempo de treinamento do experimento com o conjunto de dados G. Nesse caso, as subjanelas utilizadas nos operadores de nível 1 são de tamanho 7×6 e sua união resulta na janela 11×9 , diferente da janela 7×7 usado no operador de um nível.

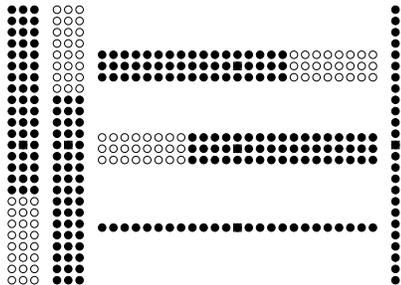
Apresentamos a seguir alguns resultados (imagens) obtidos pelos operadores projetados de acordo com a abordagem proposta. A figura 7 mostra a segmentação de estruturas específicas em imagens de diagramas funcionais, obtidos de páginas de um livro, digitalizados a 100dpi através de um *scanner* de mesa.

Experimentos similares para segmentar outras estruturas do mesmo conjunto de imagens foram também realizadas. A figura 8 mostra a segmentação de retângulos tracejados desses diagramas, enquanto a figura 9 mostra a segmentação de caracteres.

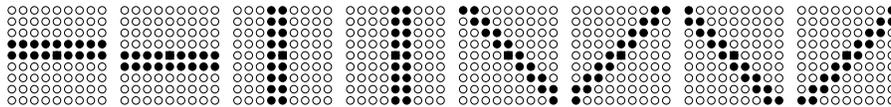
A, A''



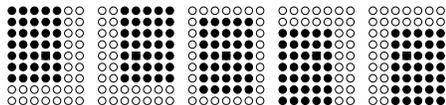
A'



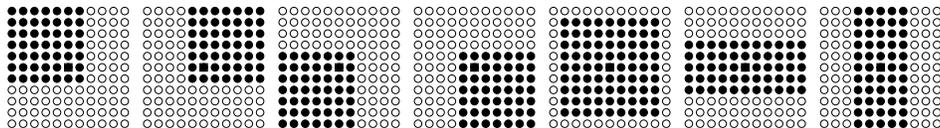
B



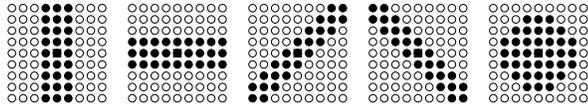
C, D₁



D₂, E



F



G

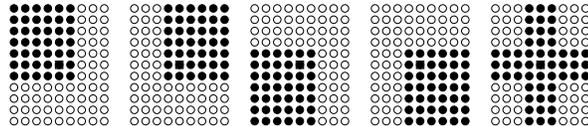


Figura 6: Janelas utilizadas nos operadores de nível 1.

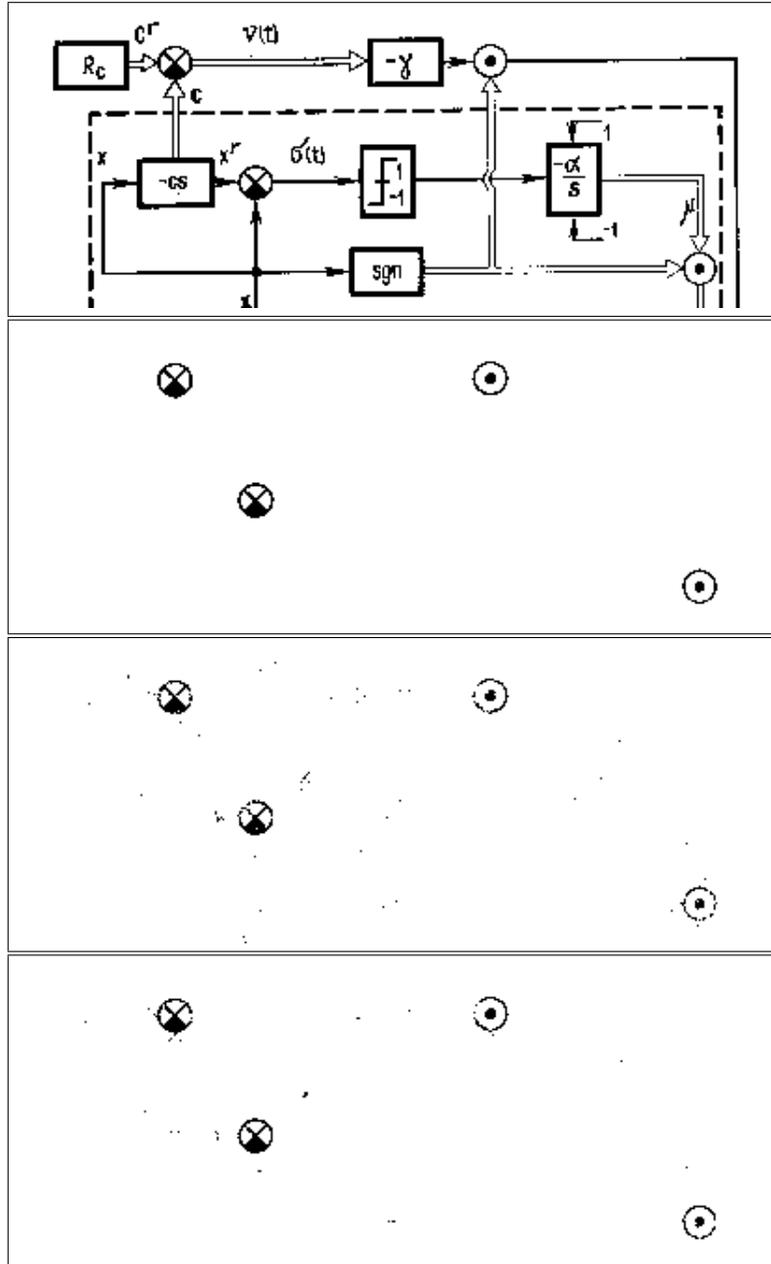


Figura 7: Experimento A (segmentação de objetos circulares): imagens de teste, ideal, resultado do operador de um de dois níveis, respectivamente.

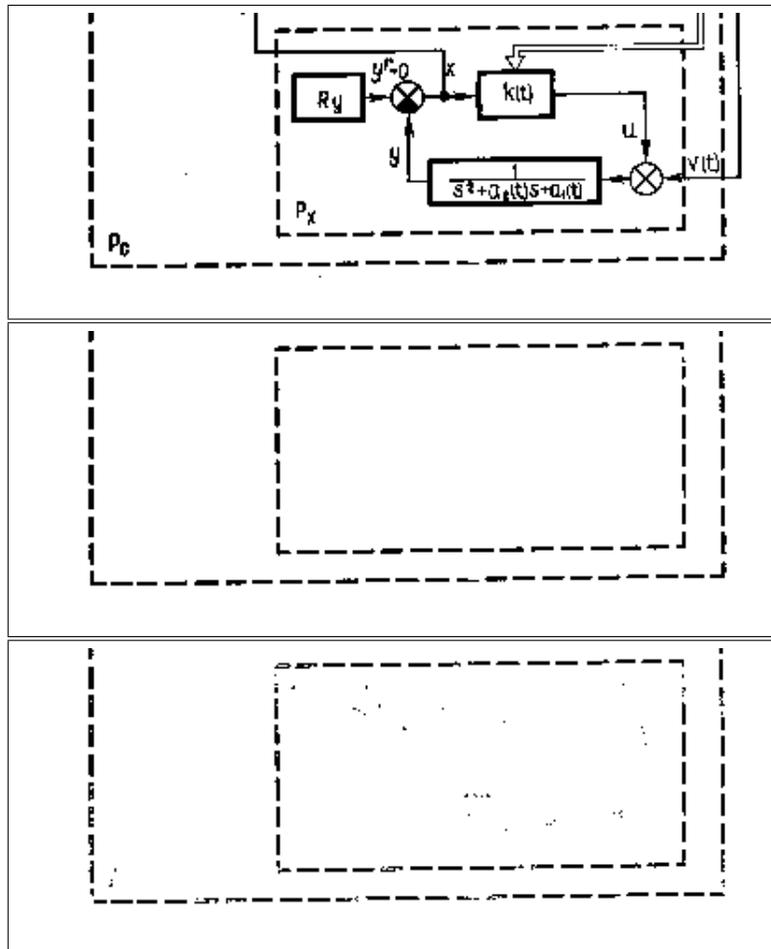


Figura 8: Experimento A' (segmentação de retângulos tracejados): imagens de teste, ideal, resultado do operador de dois níveis, respectivamente.

No segundo conjunto de imagens, imagens de mapa do tipo line-art foram processadas visando-se segmentar regiões do mapa preenchidas com um determinado padrão, no caso linhas horizontais. Essas imagens também foram digitalizadas de livros através de um scanner de mesa. As figuras 10 e 11 mostram alguns resultados.

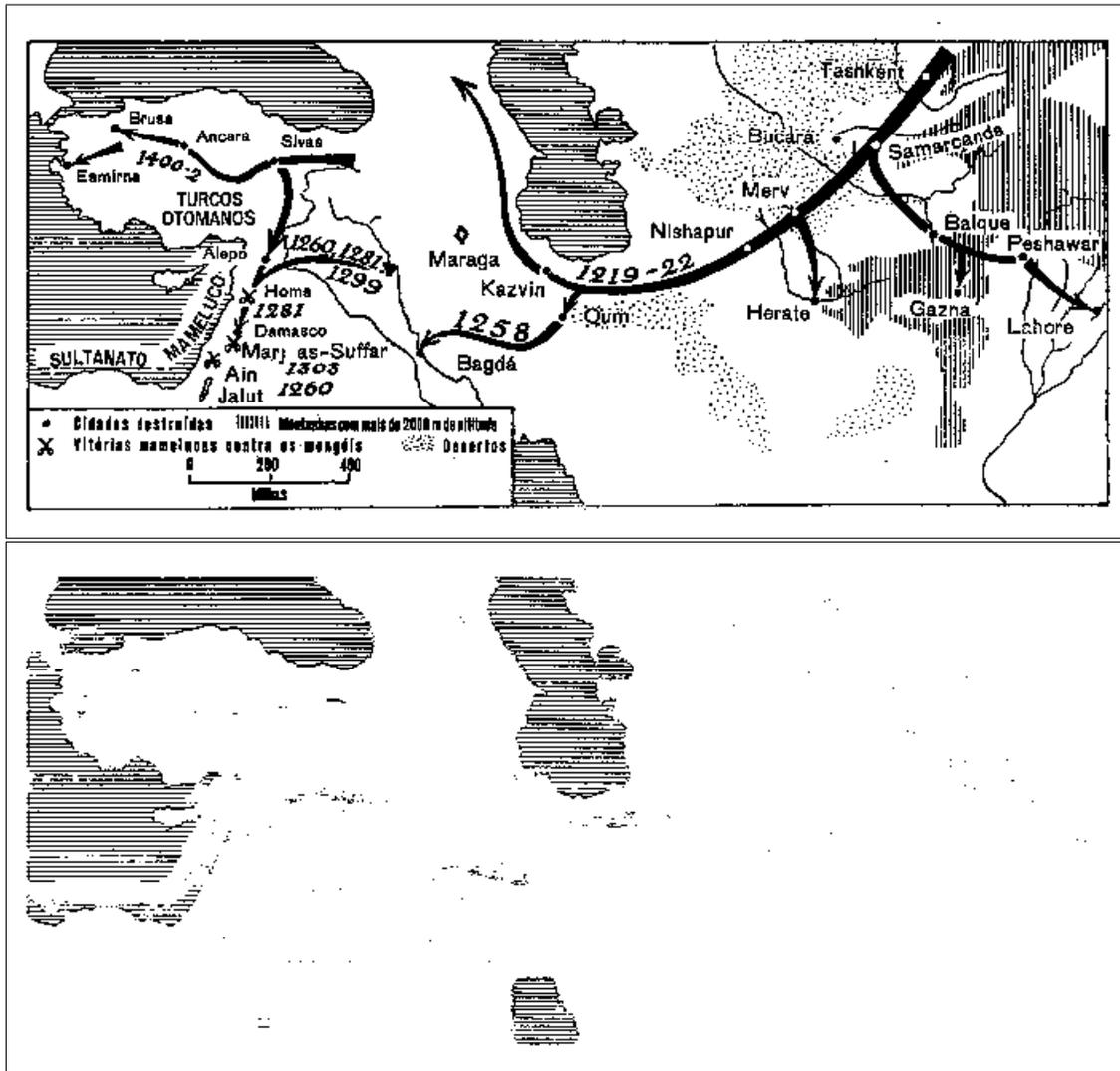


Figura 10: Experimento B (segmentação de regiões em mapas): imagem de teste e resultado do operador de dois-níveis.

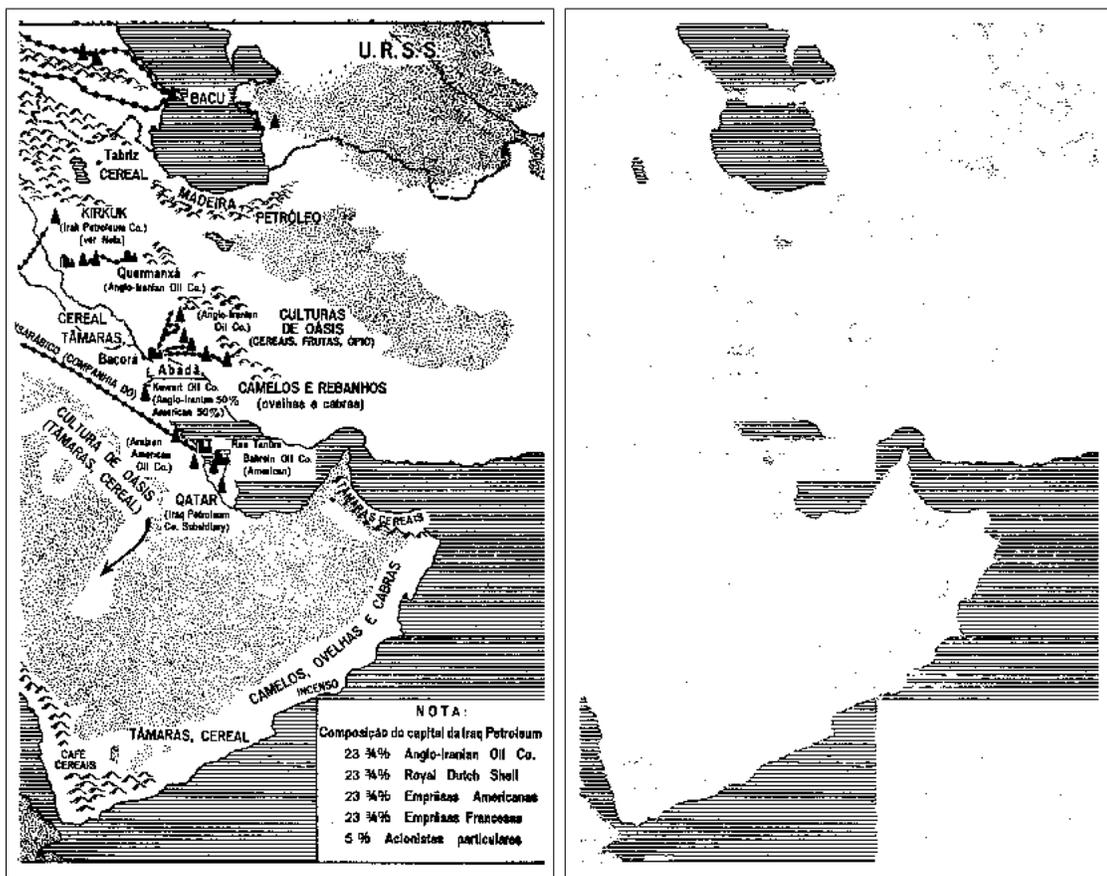


Figura 11: Experimento B (segmentação de regiões em mapas, cont.): imagem de teste e resultado do operador de dois-níveis.

No terceiro conjunto de imagens, ocorrências de caracteres “s” foram segmentadas de imagens de páginas de um livro. As figuras 12 e 13 mostra resultados da segmentação.

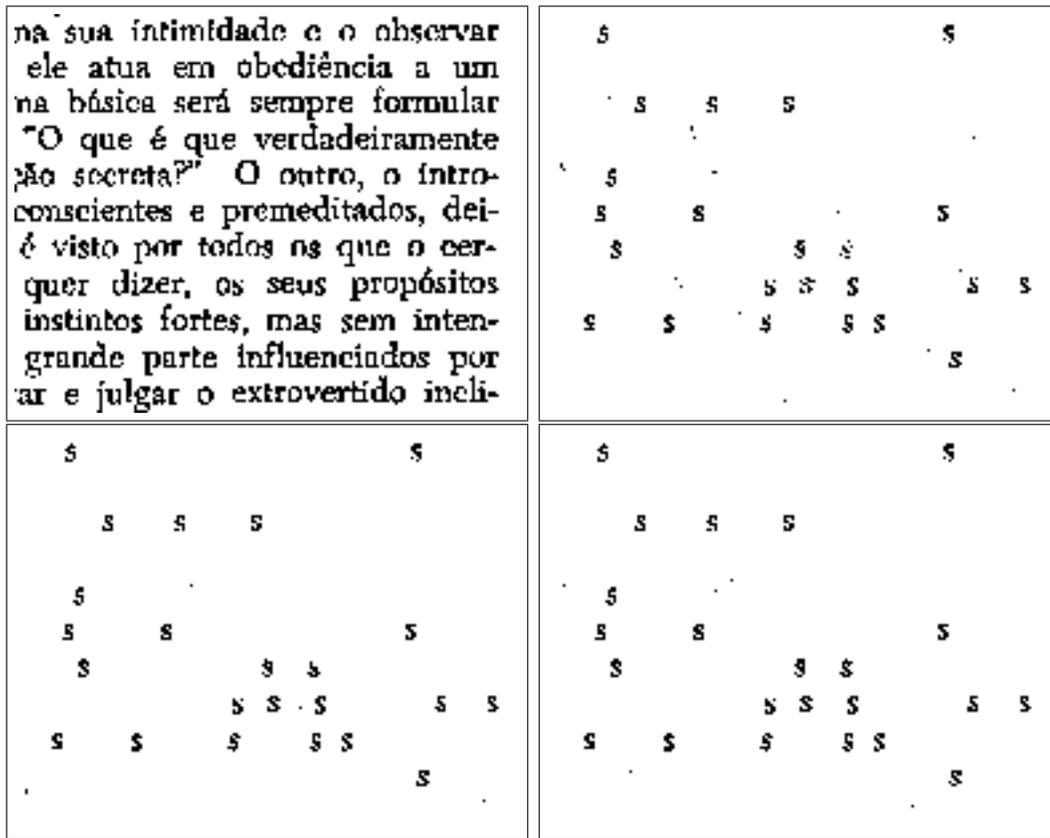


Figura 12: Experimento C (reconhecimento de caracteres): imagens de teste, resultados dos operadores de um nível, de dois níveis (exp. C₁) e de dois níveis (exp. C₂), respectivamente.

TIPOS PSICOLÓGICOS

identificação com o objeto amado. A psicologia dos primitivos oferece-nos abundantes exemplos da dissimilação em favor do objeto. É o caso da assimilação do animal totêmico ou do espírito dos ancestrais. No mesmo âmbito se situa ainda a estigmatização dos santos medievais e modernos. Na *imitatio Christi*, a dissimilação é, inclusive, elevada à categoria de princípio.

Em face dessa indubitável disposição da psique humana, no que se refere à dissimilação, é psicologicamente compreensível a entrega ao sujeito das conexões causais objetivas. Como dissemos, a psique é assim dominada pela impressão de validade única do princípio de causalidade e requer-se toda a bagagem da teoria do conhecimento para lutar contra a prepotência dessa impressão. Deve-se considerar também uma dificuldade o fato da disposição empírica, com toda a sua essência, nos impedir de crer na liberdade íntima. Pois faltam-nos completamente provas, inclusive de sua possibilidade. Que eloquência poderá ter esse pálido e vago sentimento de liberdade, ante a massa avassaladora de provas objetivas que o defrontam? Por isso é inevitável, por assim

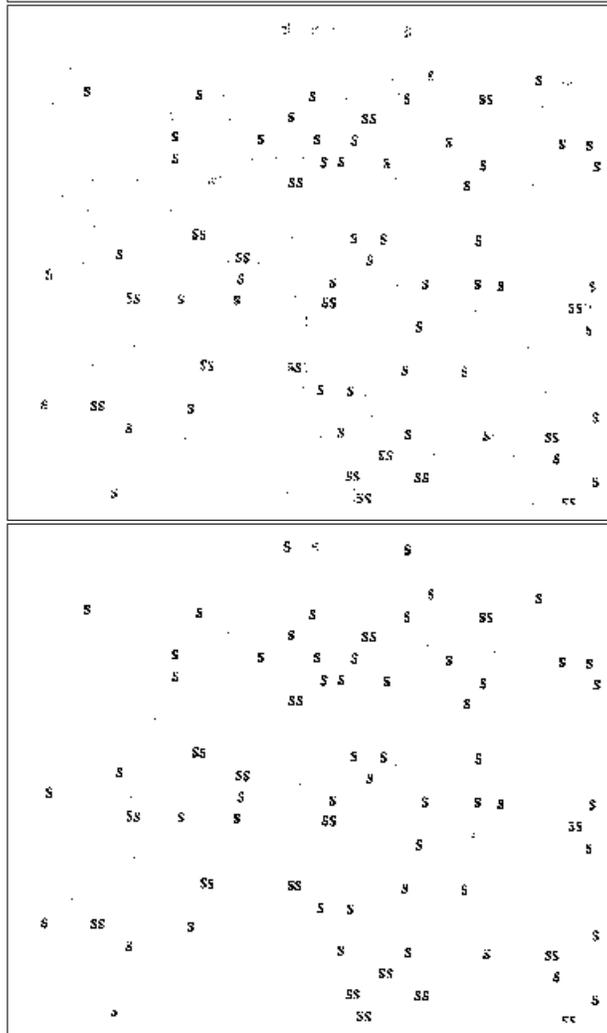


Figura 13: Experimento C (reconhecimento de caracteres): imagens de teste, resultados dos operadores de dois níveis (exp. C₁) e de dois níveis (exp. C₂), respectivamente.

No quarto conjunto de dados, consideramos o problema de segmentação de texto em imagens de páginas de revista. A figura 14 mostra a superposição das classificações efetuadas pelos operadores de nível 1 em uma pequena parte de uma imagem de teste, enquanto as figuras 15 e 16 mostram alguns resultados.

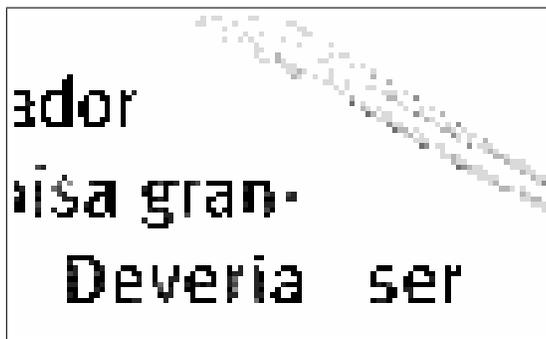


Figura 14: Experimento D': detalhe da superposição dos operadores de nível 1. Quanto mais escuro o tom do pixel, maior o número de operadores que associaram o valor 1 ao pixel.

Experimentos similares foram realizadas com páginas de livro também. As figuras 17 e 18 mostram alguns dos resultados obtidos.

Ansiada

frou confinado aos feudos das lojas especializadas. Depois pulou para lojas de eletrodomésticos, ganhou espaço nos grandes magazines. Agora conquistou o espaço nobre da popularização: apareceu em prateleiras de supermercado. Em breve, em se tratando de Brasil, deve aparecer computador em barraca de camelô.

Molecagens -- É hora do almoço num supermercado Carrefour da Zona Sul de São Paulo, conhecida como o "horário do patinho". Muita gente, pouco tempo, o maior índice de pequenos lurtos. Desde que a rede inaugurou o setor de informática, em 15 de agosto, é festa. Gravatas, tailleurs, officios e carrinhos de feira cheios estacionam na frente das máquinas em exposição. Em meio a melancias, bicicletas, refrigerantes, ninguém tem medo de se aproximar do bicho. Em supermercado, ninguém precisa pretender ser especialista. Uma senhora olha intrigada para uma mensagem em letras verdes sobre fundo azul que tem a em piscar no monitor de um micro Quest 100 (cinquenta unidades vendidas em uma semana). "Carrefour, tudo o que você NÃO quer?" A mensagem foi gravada clandestinamente, como senha, por um dos muitos microcos que sempre rondam o setor.

Essa vou tirar fácil da tela -- resmungu um dos vendedores, já treinado em apagar as molecagens.

Ele pega o mouse. Sacoleja a peça, e não. Descobre que roubaram a bolinha do mouse que faz funcionar o ratinho. Pega o do computador vizinho, mas também esse teve a bolinha surrupiada. É o Brasil entrando na era da informática.

Por que você não planeja comprar um computador?

Tem dinheiro	69%
Não sei	21%
Não tenho tempo	14%
Não sei onde comprar	9%
Não tenho espaço	7%

sa é que a máquina funciona, e isso eu garanto. Não que ela seja bonita. Um micro é sempre um micro. Tudo igual. Para conhecê-lo é só olhar o jornal". Uma tele-vendedora da Planus (grandes anúncios, pequenas instalações) até se esforça para romper a barreira. "Vou explicar: 1 byte tem 8 bits. Um caractere é o mesmo que 1 byte. Mega é 10". Portanto uma memória de 8 megas é mais do que suficiente". Dez à sexta? Suficiente para quê? Como diria o senhor da Paraíba: "Está certo, está certo".

Para o orçamento e a cabeça do comprador, sobretudo aquele que ainda não usa a máquina no trabalho, a aquisição de um com-

Em 1995 o computador chegou às prateleiras de supermercados no Brasil. De vez em quando, o micro proibido para a sua popularização foi um passo

frou confinado aos feudos das lojas especializadas. Depois pulou para lojas de eletrodomésticos, ganhou espaço nos grandes magazines. Agora conquistou o espaço nobre da popularização: apareceu em prateleiras de supermercado. Em breve, em se tratando de Brasil, deve aparecer computador em barraca de camelô.

Molecagens -- É hora do almoço num supermercado Carrefour da Zona Sul de São Paulo, conhecida como o "horário do patinho". Muita gente, pouco tempo, o maior índice de pequenos lurtos. Desde que a rede inaugurou o setor de informática, em 15 de agosto, é festa. Gravatas, tailleurs, officios e carrinhos de feira cheios estacionam na frente das máquinas em exposição. Em meio a melancias, bicicletas, refrigerantes, ninguém tem medo de se aproximar do bicho. Em supermercado, ninguém precisa pretender ser especialista. Uma senhora olha intrigada para uma mensagem em letras verdes sobre fundo azul que tem a em piscar no monitor de um micro Quest 100 (cinquenta unidades vendidas em uma semana). "Carrefour, tudo o que você NÃO quer?" A mensagem foi gravada clandestinamente, como senha, por um dos muitos microcos que sempre rondam o setor.

Essa vou tirar fácil da tela -- resmungu um dos vendedores, já treinado em apagar as molecagens.

Ele pega o mouse. Sacoleja a peça, e não. Descobre que roubaram a bolinha do mouse que faz funcionar o ratinho. Pega o do computador vizinho, mas também esse teve a bolinha surrupiada. É o Brasil entrando na era da informática.

Por que você não planeja comprar um computador?

Tem dinheiro	69%
Não sei	21%
Não tenho tempo	14%
Não sei onde comprar	9%
Não tenho espaço	7%

sa é que a máquina funciona, e isso eu garanto. Não que ela seja bonita. Um micro é sempre um micro. Tudo igual. Para conhecê-lo é só olhar o jornal". Uma tele-vendedora da Planus (grandes anúncios, pequenas instalações) até se esforça para romper a barreira. "Vou explicar: 1 byte tem 8 bits. Um caractere é o mesmo que 1 byte. Mega é 10". Portanto uma memória de 8 megas é mais do que suficiente". Dez à sexta? Suficiente para quê? Como diria o senhor da Paraíba: "Está certo, está certo".

Para o orçamento e a cabeça do comprador, sobretudo aquele que ainda não usa a máquina no trabalho, a aquisição de um com-

Em 1995 o computador chegou às prateleiras de supermercados no Brasil. De vez em quando, o micro proibido para a sua popularização foi um passo

frou confinado aos feudos das lojas especializadas. Depois pulou para lojas de eletrodomésticos, ganhou espaço nos grandes magazines. Agora conquistou o espaço nobre da popularização: apareceu em prateleiras de supermercado. Em breve, em se tratando de Brasil, deve aparecer computador em barraca de camelô.

Molecagens -- É hora do almoço num supermercado Carrefour da Zona Sul de São Paulo, conhecida como o "horário do patinho". Muita gente, pouco tempo, o maior índice de pequenos lurtos. Desde que a rede inaugurou o setor de informática, em 15 de agosto, é festa. Gravatas, tailleurs, officios e carrinhos de feira cheios estacionam na frente das máquinas em exposição. Em meio a melancias, bicicletas, refrigerantes, ninguém tem medo de se aproximar do bicho. Em supermercado, ninguém precisa pretender ser especialista. Uma senhora olha intrigada para uma mensagem em letras verdes sobre fundo azul que tem a em piscar no monitor de um micro Quest 100 (cinquenta unidades vendidas em uma semana). "Carrefour, tudo o que você NÃO quer?" A mensagem foi gravada clandestinamente, como senha, por um dos muitos microcos que sempre rondam o setor.

Essa vou tirar fácil da tela -- resmungu um dos vendedores, já treinado em apagar as molecagens.

Ele pega o mouse. Sacoleja a peça, e não. Descobre que roubaram a bolinha do mouse que faz funcionar o ratinho. Pega o do computador vizinho, mas também esse teve a bolinha surrupiada. É o Brasil entrando na era da informática.

Por que você não planeja comprar um computador?

Tem dinheiro	69%
Não sei	21%
Não tenho tempo	14%
Não sei onde comprar	9%
Não tenho espaço	7%

sa é que a máquina funciona, e isso eu garanto. Não que ela seja bonita. Um micro é sempre um micro. Tudo igual. Para conhecê-lo é só olhar o jornal". Uma tele-vendedora da Planus (grandes anúncios, pequenas instalações) até se esforça para romper a barreira. "Vou explicar: 1 byte tem 8 bits. Um caractere é o mesmo que 1 byte. Mega é 10". Portanto uma memória de 8 megas é mais do que suficiente". Dez à sexta? Suficiente para quê? Como diria o senhor da Paraíba: "Está certo, está certo".

Para o orçamento e a cabeça do comprador, sobretudo aquele que ainda não usa a máquina no trabalho, a aquisição de um com-

Em 1995 o computador chegou às prateleiras de supermercados no Brasil. De vez em quando, o micro proibido para a sua popularização foi um passo

frou confinado aos feudos das lojas especializadas. Depois pulou para lojas de eletrodomésticos, ganhou espaço nos grandes magazines. Agora conquistou o espaço nobre da popularização: apareceu em prateleiras de supermercado. Em breve, em se tratando de Brasil, deve aparecer computador em barraca de camelô.

Molecagens -- É hora do almoço num supermercado Carrefour da Zona Sul de São Paulo, conhecida como o "horário do patinho". Muita gente, pouco tempo, o maior índice de pequenos lurtos. Desde que a rede inaugurou o setor de informática, em 15 de agosto, é festa. Gravatas, tailleurs, officios e carrinhos de feira cheios estacionam na frente das máquinas em exposição. Em meio a melancias, bicicletas, refrigerantes, ninguém tem medo de se aproximar do bicho. Em supermercado, ninguém precisa pretender ser especialista. Uma senhora olha intrigada para uma mensagem em letras verdes sobre fundo azul que tem a em piscar no monitor de um micro Quest 100 (cinquenta unidades vendidas em uma semana). "Carrefour, tudo o que você NÃO quer?" A mensagem foi gravada clandestinamente, como senha, por um dos muitos microcos que sempre rondam o setor.

Essa vou tirar fácil da tela -- resmungu um dos vendedores, já treinado em apagar as molecagens.

Ele pega o mouse. Sacoleja a peça, e não. Descobre que roubaram a bolinha do mouse que faz funcionar o ratinho. Pega o do computador vizinho, mas também esse teve a bolinha surrupiada. É o Brasil entrando na era da informática.

Por que você não planeja comprar um computador?

Tem dinheiro	69%
Não sei	21%
Não tenho tempo	14%
Não sei onde comprar	9%
Não tenho espaço	7%

sa é que a máquina funciona, e isso eu garanto. Não que ela seja bonita. Um micro é sempre um micro. Tudo igual. Para conhecê-lo é só olhar o jornal". Uma tele-vendedora da Planus (grandes anúncios, pequenas instalações) até se esforça para romper a barreira. "Vou explicar: 1 byte tem 8 bits. Um caractere é o mesmo que 1 byte. Mega é 10". Portanto uma memória de 8 megas é mais do que suficiente". Dez à sexta? Suficiente para quê? Como diria o senhor da Paraíba: "Está certo, está certo".

Para o orçamento e a cabeça do comprador, sobretudo aquele que ainda não usa a máquina no trabalho, a aquisição de um com-

Em 1995 o computador chegou às prateleiras de supermercados no Brasil. De vez em quando, o micro proibido para a sua popularização foi um passo

Figura 15: Experimento D (segmentação de texto): imagens de teste, resultado do operador de um nível (exp. D₁) e de dois níveis (experimentos D₁ e D₂), respectivamente.

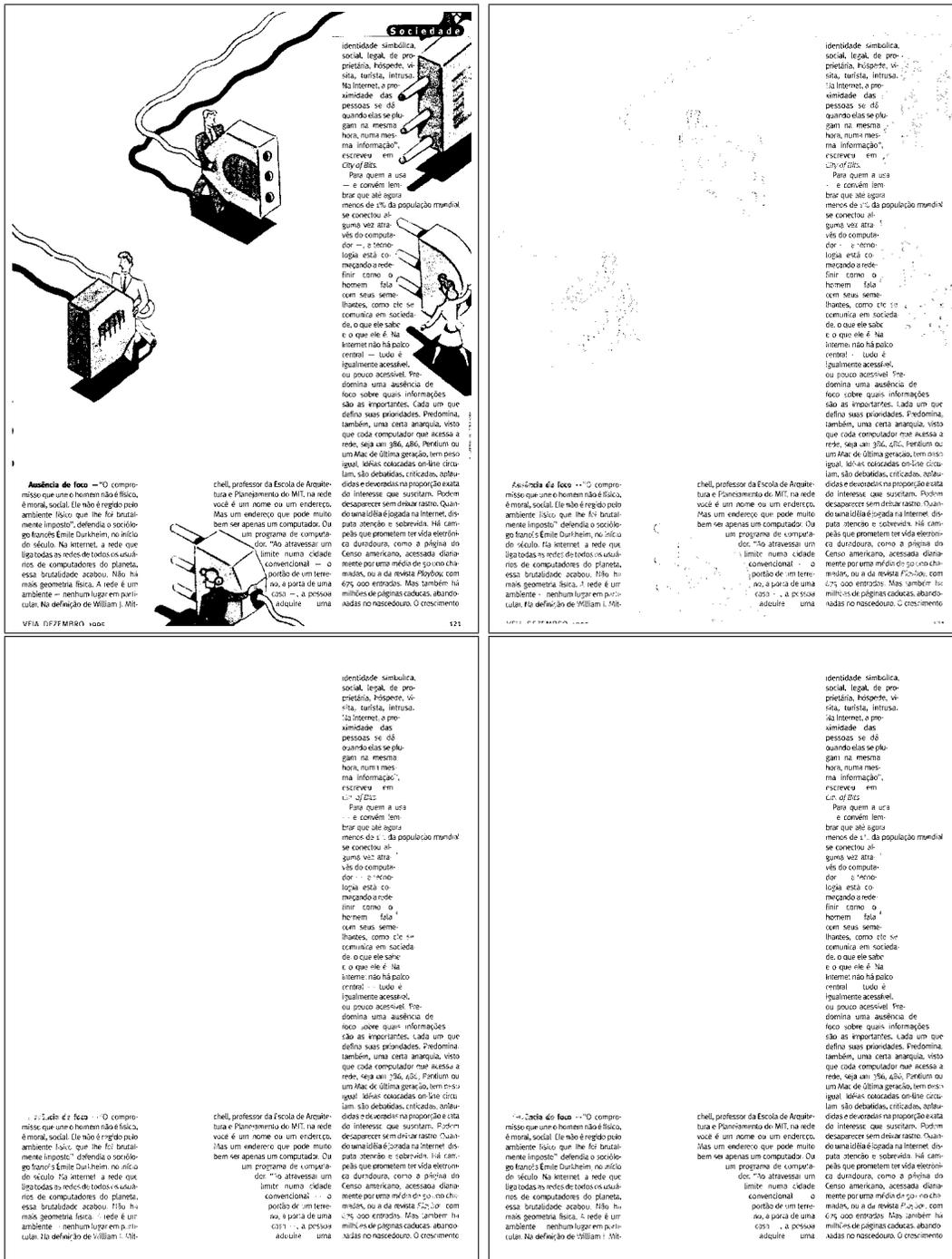


Figura 16: Experimento D (segmentação de texto): imagens de teste, resultado do operador de um nível (exp. D₁) e de dois níveis (experimentos D₁ e D₂), respectivamente.

§ 48 Die Resonanz 143

an den Ausschlagen des kleinen Pendelchens, daß jetzt die andere Stimmgabel schwingt. Es ist Resonanz eingetreten. Offenbar hat die Luft die Schwingungen der einen Gabel auf die andere übertragen. Daß so etwas möglich ist, soll uns ein weiterer Versuch mit ganz langsamen Schwingungen zeigen:

Ein 2 kg-Wägestock wird an einem etwa 2 m langen Faden aufgehängt. Dann blasen wir einmal kräftig dagegen. Der Erfolg ist eine kaum wahrnehmbare Pendelschwingung. Wir können sie aber leicht zu kräftigen Schwingungen aufschaukeln, wenn wir noch ein paar Mal gegen das Pendel blasen. Voraussetzung dafür ist, daß dies immer in der Eigenfrequenz der Pendelschwingung im richtigen Augenblick geschieht (Abb. 143.1).

Bei dem Versuch mit den beiden Resonanzstimmgabeln waren die kleineren Luftstöße, die die erste Gabel verursachte, imstande, die zweite zu kräftigen Schwingungen aufschaukeln. Beide Gabeln hatten die gleiche Eigenfrequenz, deshalb erfolgten die Luftstöße immer im richtigen Augenblick.

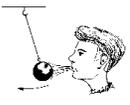
Ein weiterer Versuch zeigt, daß auch bei den Resonanzstimmgabeln die anregende und die angeregte Frequenz gleich sein müssen, wenn Resonanz eintreten soll:

Wir erniedrigen die Frequenz einer der Gabeln durch ein kleines Zusatzkörperchen, das wir an ihre Zinken klemmen. Wiederholen wir jetzt den Versuch 14, so finden wir keine Resonanz mehr.

Erfährt ein schwingungsfähiger Körper aufeinanderfolgend kleine Stöße in seiner Eigenfrequenz, so wird er zu kräftigen Schwingungen angeregt: es tritt Resonanz ein.

Eine praktische Anwendung findet die Resonanz beim Bau von Frequenzmessern. Solche Geräte besitzen viele Stabzungen verschiedener Länge. Die zu messende Frequenz regt die Zunge zum Schwingen an, deren Eigenfrequenz mit ihr übereinstimmt (s. Abb. 143.2).

Manche Teile einer Maschine oder eines Autos sind schwingungsfähige Gebilde. Fällt die Drehzahl des Motors mit der Eigenfrequenz eines solchen Teils zusammen, so erzeugt er Resonanzschwingungen. Derartige Schwingungen können so stark werden, daß sie zu Zerstörungen führen. Auf Drehzahlmassen werden die Bereiche solcher kritischen Drehzahlen durch rote Sektoren gekennzeichnet; auf ihnen soll der Zeiger des Instruments nicht lange verweilen.



143.1 Luftstöße bringen das schwere Pendel zum Schwingen



143.2 Frequenzmesser. Der halb durchsichtig erscheinende Deckel läßt die Stabzungen erkennen, die durch Beschießen mit Luft von oben erde auf die kleine Eigenfrequenz gebracht sind. Die Zunge mit 150 Hz schwingt in Resonanz.

§ 48 Die Resonanz 143

an den Ausschlagen des kleinen Pendelchens, daß jetzt die andere Stimmgabel schwingt. Es ist Resonanz eingetreten. Offenbar hat die Luft die Schwingungen der einen Gabel auf die andere übertragen. Daß so etwas möglich ist, soll uns ein weiterer Versuch mit ganz langsamen Schwingungen zeigen:

Ein 2 kg Wägestock wird an einem etwa 2 m langen Faden aufgehängt. Dann blasen wir einmal kräftig dagegen. Der Erfolg ist eine kaum wahrnehmbare Pendelschwingung. Wir können sie aber leicht zu kräftigen Schwingungen aufschaukeln, wenn wir noch ein paar Mal gegen das Pendel blasen. Voraussetzung dafür ist, daß dies immer in der Eigenfrequenz der Pendelschwingung im richtigen Augenblick geschieht (Abb. 143.1).

Bei dem Versuch mit den beiden Resonanzstimmgabeln waren die kleineren Luftstöße, die die erste Gabel verursachte, imstande, die zweite zu kräftigen Schwingungen aufschaukeln. Beide Gabeln hatten die gleiche Eigenfrequenz, deshalb erfolgten die Luftstöße immer im richtigen Augenblick.

Ein weiterer Versuch zeigt, daß auch bei den Resonanzstimmgabeln die anregende und die angeregte Frequenz gleich sein müssen, wenn Resonanz eintreten soll:

Wir erniedrigen die Frequenz einer der Gabeln durch ein kleines Zusatzkörperchen, das wir an ihre Zinken klemmen. Wiederholen wir jetzt den Versuch 14, so finden wir keine Resonanz mehr.

Erfährt ein schwingungsfähiger Körper aufeinanderfolgend kleine Stöße in seiner Eigenfrequenz, so wird er zu kräftigen Schwingungen angeregt: es tritt Resonanz ein.

Eine praktische Anwendung findet die Resonanz beim Bau von Frequenzmessern. Solche Geräte besitzen viele Stabzungen verschiedener Länge. Die zu messende Frequenz regt die Zunge zum Schwingen an, deren Eigenfrequenz mit ihr übereinstimmt (s. Abb. 143.2).

Manche Teile einer Maschine oder eines Autos sind schwingungsfähige Gebilde. Fällt die Drehzahl des Motors mit der Eigenfrequenz eines solchen Teils zusammen, so erzeugt er Resonanzschwingungen. Derartige Schwingungen können so stark werden, daß sie zu Zerstörungen führen. Auf Drehzahlmassen werden die Bereiche solcher kritischen Drehzahlen durch rote Sektoren gekennzeichnet; auf ihnen soll der Zeiger des Instruments nicht lange verweilen.



143.1 Luftstöße bringen das schwere Pendel zum Schwingen



143.2 Frequenzmesser. Der halb durchsichtig erscheinende Deckel läßt die Stabzungen erkennen, die durch Beschießen mit Luft von oben erde auf die kleine Eigenfrequenz gebracht sind. Die Zunge mit 100 Hz schwingt in Resonanz.

Figura 17: Experimento E (segmentação de texto): imagem de teste e resultado do operador de dois níveis, respectivamente.

288 IV. Elektrische Gesetze und Größen

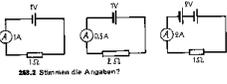
Wenn zwischen den Enden eines Leiters die Spannung U liegt und in ihm Strom der Stärke I fließt, berechnet man den Widerstand R des Leiters nach

$$R = \frac{U}{I} \text{ (Definition des Widerstands), (288.1)}$$

Die Einheit des Widerstands ist 1 Ohm (Ω) = 1 Volt/Ampere.

3. G. S. Ohne untersucht auch, wie sich der Widerstand eines Drahtes beim Erwärmen ändert:

Erhalte mit dem Boussolemeter eine Wendel aus einem Eisendraht, die nach Abb. 288.3 in einem Stromkreis liegt. Die Stromstärke I sinkt erheblich, sobald die Spannung U konstant bleibt. Nach $R = U/I$ erhöht also beim Erwärmen der Widerstand des Eisens $R_{288.4}$ beim Erhitzen konstanten Widerstand. Deshalb brennt in V 76 der Widerstand trotz der Erwärmung des Konstantendrahts durch den Strom konstant. Aus dem gleichen Grund bewirkt man diese und ähnliche Leistungen in Maßgeräten. Bei Kohle sinkt beim Erwärmen der Widerstand.



288.3 Stromkreis der Angaben?



288.4 Beim Erwärmen steigt der Widerstand des Metall

288.4 Temperaturabhängigkeit von Widerständen

Mit nach V 76 des Widerstand einer Glühlampe bei sehr kleiner Stromstärke. Auf welchen Wert steigt er bei normaler Belastung infolge der Erwärmung?

Der Widerstand von Metallen nimmt im allgemeinen beim Erwärmen zu.

Stellt man durch Versuch den Zusammenhang zwischen Widerstand und Temperatur fest, so kann man nachher aus dem gemessenen Widerstand auf die Temperatur schließen. Diese Widerstandsthermometer partieren auf elektrischem Wege Temperaturen an entfernten oder schwer zugänglichen Orten sowie bei großer Hitze zu messen (Flugzeugmotoren, Öfen). Siehe auch § 112.

288 IV. Elektrische Gesetze und Größen

Wenn zwischen den Enden eines Leiters die Spannung U liegt und in ihm Strom der Stärke I fließt, berechnet man den Widerstand R des Leiters nach

$$R = \frac{U}{I} \text{ (Definition des Widerstands), (288.1)}$$

Die Einheit des Widerstands ist 1 Ohm (Ω) = 1 Volt/Ampere.

3. G. S. Ohne untersucht auch, wie sich der Widerstand eines Drahtes beim Erwärmen ändert:

Erhalte mit dem Boussolemeter eine Wendel aus einem Eisendraht, die nach Abb. 288.3 in einem Stromkreis liegt. Die Stromstärke I sinkt erheblich, sobald die Spannung U konstant bleibt. Nach $R = U/I$ erhöht also beim Erwärmen der Widerstand des Eisens $R_{288.4}$ beim Erhitzen konstanten Widerstand. Deshalb brennt in V 76 der Widerstand trotz der Erwärmung des Konstantendrahts durch den Strom konstant. Aus dem gleichen Grund bewirkt man diese und ähnliche Leistungen in Maßgeräten. Bei Kohle sinkt beim Erwärmen der Widerstand.



288.3 Stromkreis der Angaben?



288.4 Beim Erwärmen steigt der Widerstand des Metall

288.4 Temperaturabhängigkeit von Widerständen

Mit nach V 76 des Widerstand einer Glühlampe bei sehr kleiner Stromstärke. Auf welchen Wert steigt er bei normaler Belastung infolge der Erwärmung?

Der Widerstand von Metallen nimmt im allgemeinen beim Erwärmen zu.

Stellt man durch Versuch den Zusammenhang zwischen Widerstand und Temperatur fest, so kann man nachher aus dem gemessenen Widerstand auf die Temperatur schließen. Diese Widerstandsthermometer partieren auf elektrischem Wege Temperaturen an entfernten oder schwer zugänglichen Orten sowie bei großer Hitze zu messen (Flugzeugmotoren, Öfen). Siehe auch § 112.

Figura 18: Experimento E (segmentação de texto): imagem de teste e resultado do operador de dois níveis, respectivamente.

No quinto conjunto de dados, consideramos um problema de restauração. As imagens utilizadas foram simuladas seguindo-se um modelo booleano cujas primitivas são retângulos e os ruídos são subconjuntos do quadrado 3×3 , ambos uniformemente distribuídos no domínio da imagem. Resultados da restauração são mostrados na figura 19.

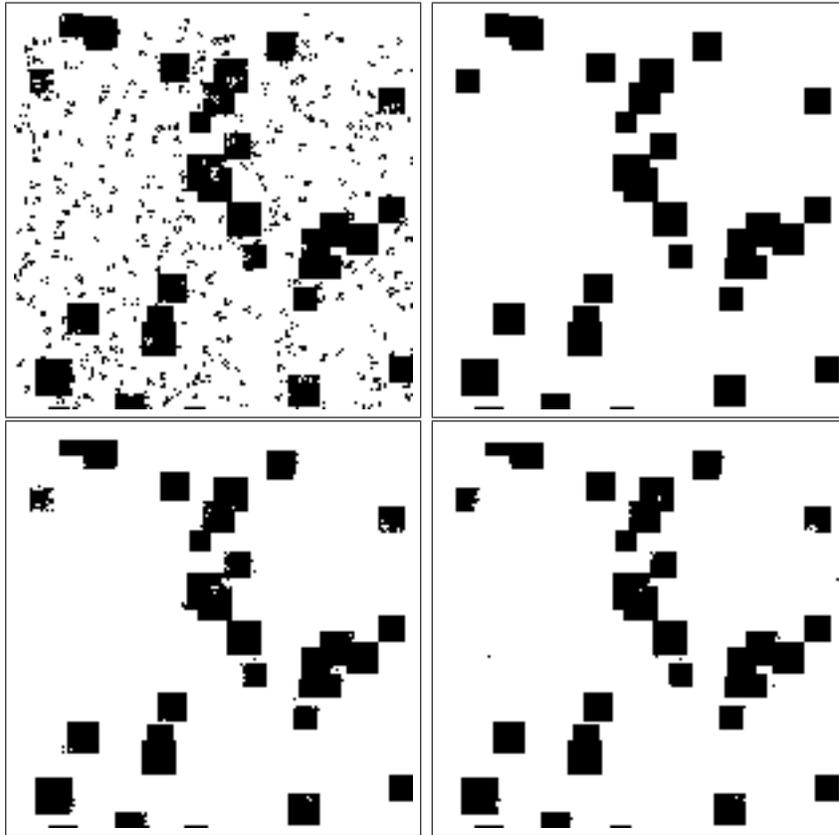


Figura 19: Experimento F (filtragem de ruído booleano): imagens de teste, ideal, resultado dos operadores de um e de dois níveis, respectivamente.

No conjunto de dados G, consideramos o problema de filtragem de ruído simulado em borda de caracteres de texto. Os ruídos são aditivos e subtrativos.

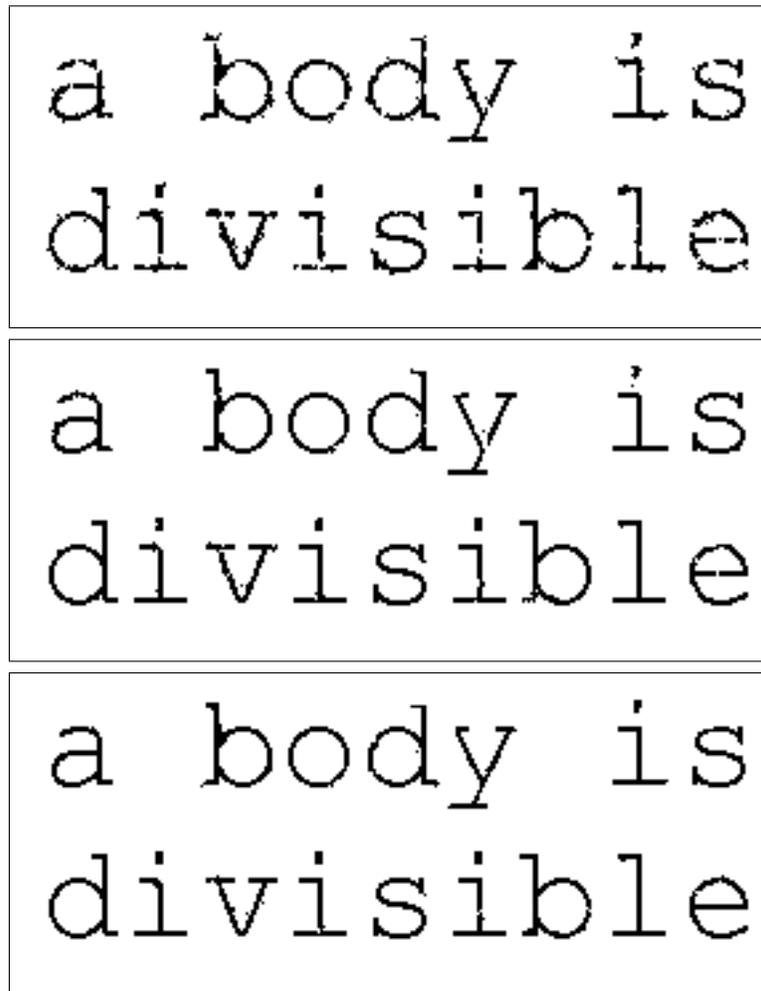


Figura 20: Experimento F (filtragem de ruído de borda): imagens de teste, resultados dos operadores de um e de dois níveis, respectivamente.

6 Discussão e Perspectivas futuras

Neste trabalho foi proposto o projeto multi-níveis de operadores como uma abordagem para contornar o problema de *overfitting* e excessivo tempo de treinamento de operadores projetados sobre janelas grandes. Observamos experimentalmente que dois níveis de treinamento resultam em operadores com desempenho consistentemente superior aos operadores com apenas um nível de treinamento, tanto em relação ao erro MAE quanto em relação ao tempo total de processamento.

Destacamos que a abordagem de projetar operadores morfológicos é abrangente no sentido de que pode ser aplicada a diferentes problemas de processamento de imagens (no caso de imagens binárias, notadamente em segmentação e filtragem). Claramente, uma abordagem que seja específica para um determinado problema de segmentação, por exemplo para a segmentação de textos, provavelmente apresentaria um resultado melhor que o obtido aqui. Porém um procedimento tão específico só poderia ser aplicado para aquele problema. Em contraposição, o procedimento descrito neste relatório é o mesmo independente do problema tratado: basta treinarmos os operadores com os dados de treinamento adequados. A expectativa é de que esses operadores, embora possam não resolver totalmente um problema, possam compor parte de uma solução. Por exemplo, no problema de segmentação de textos, uma solução customizada provavelmente envolveria vários parâmetros que precisariam ser ajustados na base da tentativa e erro para diferentes tipos de documentos (fontes, estilos, layouts, etc). Acreditamos que uma solução mista, em que parte da solução advém de treinamento como o proposto neste trabalho poderá reduzir enormemente a necessidade de ajuste destes parâmetros.

6.1 Discussões

Os resultados experimentais mostram que operadores projetados em dois níveis de treinamento, conforme o modelo proposto, têm desempenho consistentemente superior aos operadores projetados diretamente sobre uma janela grande. Essa superioridade é tanto em relação ao erro MAE quanto em relação ao tempo de treinamento. Mais ainda, não foi necessário um esforço muito grande para se encontrar uma combinação de subjanelas (janelas dos operadores de nível 1) cuja combinação dos resultados dos operadores fosse superior ao do operador projetado diretamente sobre a união das subjanelas.

Nesta seção pretendemos argumentar por quê a combinação proposta funciona, ou seja, apresenta desempenho superior. Uma explicação intuitiva é a seguinte. Devido ao fenômeno de *overfitting*, sabemos que para uma determinada quantidade de dados de treinamento existe uma “janela ótima” W_1 que resulta em MAE empírico mínimo. Janelas W_2 maiores que a janela W_1 , com os mesmos dados de treinamento, resultam em operadores com desempenho inferior. Considere então diferentes janelas com tamanho similar ao da janela W_1 e projete um operador para cada uma dessas janelas. Uma clara opção para o operador de segundo nível seria escolher como saída o resultado do operador de nível 1 com melhor desempenho. Logo, se supormos que a janela W_1 é uma das janelas no nível 1, temos a garantia de que o operador de dois níveis terá desempenho melhor ou

igual ao do operador de nível 1 com respeito à janela W_1 . É razoável esperar que essa característica seja preservada se, em vez de escolhermos um dos operadores de nível 1 como saída do operador de nível 2, determinarmos a saída do operador de nível 2 através de treinamento.

Observamos também que nos experimentos realizados o operador de segundo nível são classificadores do tipo *voto majoritário*: ou seja, se mais da metade dos operadores de primeiro nível classificam um padrão como da classe 1, o operador de segundo nível associa a classe 1 ao padrão; e vice-versa. Mais rigorosamente, todos eles são do tipo *ordem estatística*, ou seja, se pelo menos k , com $1 \leq k \leq n_1$, operadores de primeiro nível classificam um padrão como sendo da classe 1, então o operador de segundo nível associa a classe 1 ao padrão.

Uma análise rigorosa dos resultados ainda está para ser realizada e será incorporada nas futuras versões deste relatório.

6.2 Trabalhos futuros

Nos experimentos realizados utilizamos apenas janelas de tamanho 1 no operador de segundo nível. Se janelas maiores fossem utilizadas, a janela total associada ao operador poderia ficar muito grande (pois este tamanho corresponde à soma do tamanho das janelas associadas a cada imagem de entrada), resultando em *overfitting*. Um dos problemas a serem considerados é se o uso de uma janela de tamanho maior que 1, eventualmente reduzindo o número total de janelas envolvidas, poderia resultar em operadores com desempenho melhor. O projeto iterativo mostra que há melhora no desempenho. Portanto é de se esperar que a resposta a esta pergunta seja afirmativa. Da mesma forma, projetar operadores de mais de dois níveis também é um tópico a ser investigado.

Outra questão crucial no projeto multi-níveis é a escolha de sua arquitetura, ou seja, a escolha do número de níveis, número de operadores em cada nível e as respectivas janelas. Para imagens de alta resolução, o uso de sub-janelas adequadas pode ser utilizado para tratar imagens em diferentes resoluções. Portanto, investigar a relação entre o projeto multi-níveis e o projeto multiresolução [VBH05] de operadores morfológicos é também outro tópico para futura investigação. Além disso, nos experimentos realizados, apenas um processo aleatório (imagens observadas) foi utilizada como entrada para os operadores de primeiro nível. No entanto, é possível considerarmos outros processos que, a cada pixel associam diferentes tipos de características que não as geométricas capturadas pelas subjanelas. Por exemplo, algumas das possíveis características são medidas tais como área, altura e largura máximas, existência de buracos ou protuberâncias, entre outras no componente que contém o pixel, além de distâncias do pixel ao centróide ou borda do componente. Essas características dificilmente são capturadas pelas janelas se os componentes forem grandes.

Ressaltamos também que este trabalho foi restrito às imagens binárias, porém pode ser bastante útil no contexto de imagens em níveis de cinza, uma vez que o *overfitting* é um problema mais grave no caso de imagens em níveis de cinza.

Finalmente, do ponto de vista mais formal, acreditamos que o modelo proposto assim como diferentes esquemas de combinação de operadores (ou classificadores) podem ser vistos sob a perspectiva de composição de funções. Desta forma, um problema interessante é estudarmos, uma vez dado um esquema de combinação, qual é a classe de operadores associada a esse esquema. Será possível descrever esses operadores, encontrar propriedades algébricas entre outros? Sob outra perspectiva, dado um operador, é possível sabermos se esse operador pertence a classe de operadores associados a um dado esquema de composição de operadores?

Essas são as questões fundamentais que poderão ser investigadas em prosseguimento ao trabalho descrito neste relatório. Alguns destes, como a escolha de subjanelas associadas aos operadores de nível 1, estão sendo atualmente investigados [SHR07].

Agradecimentos

Este trabalho foi desenvolvido durante a vigência do projeto FAPESP 04/11586-7 (“Combinação de classificadores”). A autora recebe apoio parcial do CNPq (processo 312482/2006-0).

Referências

- [AGK98] K. Al-Ghoneim and B. V. K. V. Kumar. Unified Decision Combination Framework. *Pattern Recognition*, 31(12):2077–2089, December 1998.
- [Alp98] E. Alpaydin. Techniques for Combining Multiple Learners. In *Engineering of Intelligent Systems EIS’98*, volume 2, pages 6–12, Tenerife, Spain, February 1998.
- [BB91] G. J. F. Banon and J. Barrera. Minimal Representations for Translation-Invariant Set Mappings by Mathematical Morphology. *SIAM J. Applied Mathematics*, 51(6):1782–1798, December 1991.
- [BB93] G. J. F. Banon and J. Barrera. Decomposition of Mappings between Complete Lattices by Mathematical Morphology, Part I. General Lattices. *Signal Processing*, 30:299–327, 1993.
- [BK99] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–139, 1999.
- [Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

- [Die00] T. G. Dietterich. Ensemble Methods in Machine Learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems: First International Workshop, MCS 2000*, volume LNCS 1857 of *Lecture Notes in Computer Science*, pages 1–15. 2000.
- [Gho02] J. Ghosh. Multiclassifier Systems: Back to the Future. In F. Roli and J. Kittler, editors, *Multiple Classifier Systems: Third International Workshop, MCS 2002*, volume LNCS 2364 of *Lecture Notes in Computer Science*, pages 1–15. 2002.
- [GHvL03] G. Goos, J. Hartmanis, and J. van Leeuwen, editors. *Multiple Classifier Systems: Fourth International Workshop, MCS 2003*, volume 2709 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.
- [GMS02] N. Giusti, F. Masulli, and A. Sperduti. Theoretical and experimental analysis of a two-stage system for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):893–904, 2002.
- [HBBD02] R. Hirata Jr., Marcel Brun, Junior Barrera, and Edward R. Dougherty. Multiresolution Design of Aperture Operators. *Journal of Mathematical Imaging and Vision*, 6(3):199–222, 2002.
- [HBFL99] R. Hirata Jr., J. Barrera, F. C. Flores, and R. A. Lotufo. Automatic Design of Morphological Operators for Motion Segmentation. In J. Stolfi and C. L. Tozzi, editors, *Proc. of Sibgrapi'99*, pages 283–292, Campinas, SP, Brazil, 1999.
- [HBTD02] N. S. T. Hirata, J. Barrera, R. Terada, and E. R. Dougherty. The Incremental Splitting of Intervals Algorithm for the Design of Binary Image Operators. In H. Talbot and R. Beare, editors, *Proceedings of the 6th International Symposium: ISMM 2002*, pages 219–228, 2002.
- [HDB00a] N. S. T. Hirata, E. R. Dougherty, and J. Barrera. Iterative Design of Morphological Binary Image Operators. *Optical Engineering*, 39(12):3106–3123, December 2000.
- [HDB00b] R. Hirata Jr., E. R. Dougherty, and J. Barrera. Aperture Filters. *Signal Processing*, 80(4):697–721, April 2000.
- [Hei94] H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, Boston, 1994.
- [HHS94] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.
- [Hir00] Nina S. T. Hirata. *Projeto Automático de Operadores – Explorando Conhecimentos a Priori*. PhD thesis, Instituto de Matemática e Estatística da USP, São Paulo, Brazil, October 2000.

- [Hir01] R. Hirata Jr. *Projeto de Operadores Morfológicos para Imagens e Sinais – Abordagem de Reticulados Finitos Discretos*. PhD thesis, Instituto de Matemática e Estatística da USP, São Paulo, Brazil, December 2001.
- [Hir07] Nina S. T. Hirata. Document processing via trained morphological operators. In *To appear in Proceedings of ICDAR*, 2007.
- [HKR05] M. Haindl, J. Kittler, and F. Roli, editors. *Multiple Classifier Systems: 7th International Workshop, MCS 2007*, volume 4472 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.
- [HP93] F. J. Hill and G. R. Peterson. *Computer Aided Logical Design with Emphasis on VLSI*. John Wiley & Sons, fourth edition, 1993.
- [HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.
- [JDM00] A. K. Jain, R. P.W. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), January 2000.
- [KA03] J. Kittler and F. M. Alkoot. Sum versus vote fusion in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):110–115, 2003.
- [KHDM98] J. Kittler, M. Hatef, R. P.W. Duin, and J. Matas. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.
- [KR94] K. Konstantinides and J. Rasure. The KHOROS Software Development Environment for Image and Signal Processing. *IEEE Transactions on Image Processing*, 3(3):243–252, 1994.
- [KR00] J. Kittler and F. Roli, editors. *Multiple Classifier Systems: First International Workshop, MCS 2000*, volume 1857 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.
- [KR01] J. Kittler and F. Roli, editors. *Multiple Classifier Systems: Second International Workshop, MCS 2001*, volume 2096 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.
- [Kun02] L. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):281–286, 2002.
- [Kun04] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.

- [Mar85] P. A. Maragos. *A Unified Theory of Translation-invariant Systems with Applications to Morphological Analysis and Coding of Images*. PhD thesis, School of Elect. Eng. - Georgia Inst. Tech., 1985.
- [Mar89] P. Maragos. A Representation Theory for Morphological Image and Signal Processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):586–599, June 1989.
- [Mat75] G. Matheron. *Random Sets and Integral Geometry*. John Wiley, 1975.
- [Mit97] T. M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill, March 1997.
- [OPKR05] N. C. Oza, F. Polikar, J. Kittler, and F. Roli, editors. *Multiple Classifier Systems: 6th International Workshop, MCS 2005*, volume 3541 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.
- [RF99] A. F. R. Rahman and M. C. Fairhurst. Serial combination of multiple experts: A unified evaluation. *Pattern Analysis & Applications*, 2:292–311, 1999.
- [RK02] F. Roli and J. Kittler, editors. *Multiple Classifier Systems: Third International Workshop, MCS 2002*, volume 2364 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.
- [RKW04] F. Roli, J. Kittler, and T. Windeatt, editors. *Multiple Classifier Systems: 5th International Workshop, MCS 2004*, volume 3077 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.
- [Ser82] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [Ser88] J. Serra. *Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances*. Academic Press, 1988.
- [SHR07] C. S. Santos, N. S. T. Hirata, and R. Hirata Jr. Two-stage binary image operator design: an approach based on interaction information. In *To appear in Proceedings of Proceedings of SIBGRAPI*, 2007.
- [Soi03] P. Soille. *Morphological Image Analysis*. Springer-Verlag, Berlin, 2nd edition, 2003.
- [TK06] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, third edition, 2006.
- [Tom96] N. S. Tomita. Programação Automática de Máquinas Morfológicas Binárias baseada em Aprendizado PAC. Master’s thesis, Instituto de Matemática e Estatística - Universidade de São Paulo, São Paulo, SP - Brasil, março 1996.

- [TvBDK00] D. M. J. Tax, M. van Breukelen, R. P.W. Duin, and J. Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition*, 33(9):1475–1485, 2000.
- [VBH05] D. A. Vaquero, J. Barrera, and R. Hirata Jr. A maximum-likelihood approach for multiresolution W-operator design. In *Proceedings of the XVIII SIBGRAPI*, pages 71–78, 2005.
- [Wol92a] D. H. Wolpert. *Fast Learning and Invariant Object Recognition*, chapter How to Deal with Multiple Possible Generalizers, pages 61–79. John Wiley and Sons, 1992.
- [Wol92b] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.