Herond Robaina Salles

Clustering de imagens via redes neurais de Kohonen associadas a momentos invariantes

Campos dos Goytacazes/RJ

2012

Herond Robaina Salles

Clustering de imagens via redes neurais de Kohonen associadas a momentos invariantes

Monografia apresentada ao Curso de Graduação em Ciência da Computação da Universidade Estadual do Norte Fluminense Darcy Ribeiro como requisito para obtenção do título de Bacharel em Ciência da Computação, sob orientação da Prof[a]. Annabell Del Real Tamariz, DrSc.

Tutor: Annabell Del Real Tamariz, DrSc.

Universidade Estadual do Norte Fluminense Darcy Ribeiro

Campos dos Goytacazes/RJ

2012

"E posto que se infligiram inutilmente ao corpo social tantos sistemas, que se termine por onde se deveria ter começado: que se rejeitem os sistemas; que se coloque, por fim, a Liberdade à prova - a Liberdade, que é um ato de fé em Deus e em sua obra."

Frederic Bastiat

Agradecimentos

AGRADECIMENTOS AQUI.

Lista de Figuras

Lista de Códigos

Resumo

RESUMO AQUI.

Sumário

1 Introdução

INTRODUÇÃO AQUI.

2 Descritores de imagens

Tendo em vista que as imagens, devido a grade quantidade de informações contidas em sua representação, não servem como entrada para uma rede de Kohonen, é necessário extrair delas um conjunto resumido e mensurável de características que possam servir de entrada para rede. É sobre a extração deste conjunto de características que trata este capítulo. Serão abordados os descritores escolhidos para caracterizar uma imagem e também todo o tratamento que a imagem sofre até que estes descritores possam ser extraídos.

2.1 Conceitos introdutórios sobre imagens digitais

Antes de inicar qualquer discução a respeito da caracterização e agrupamento de imagens é necessário discorrer sob o modo com as imagens são representadas computacionalmente, os termos comumente empregados nestas representações e, em particular, estabelecer quais destas representações serão adotadas neste trabalho.

Existe basicamente duas formas de representação computacional de imagens, mapa de bits (*bitmap*): uma matriz de pontos (*pixels*) que representam cores; ou vetoriais: um conjunto de descrições de formas geométricas, cores e texturas que, precisamente por serem equações vetoriais ou trasformações matemáticas, não perdem a qualidade quando redimensionadas ou rotacionadas; a comparação entre estas duas representações pode ser observada na Figura 1.

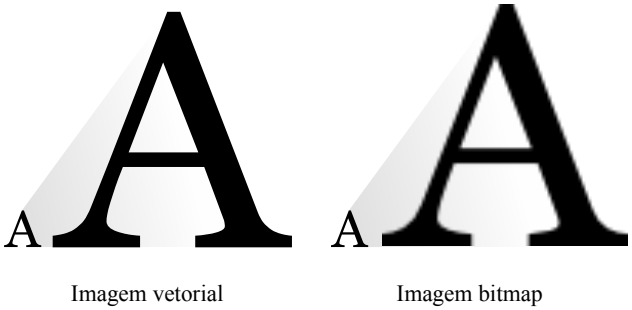

Figura 1: Diferença entre uma imagem vetorial e uma imagem *bitmap*. Uma imagem *bitmap* perde a qualidade quando ampliada, o que não ocorre com uma imagem vetorial

Quanto a cores, existem diversas padrões de representação, a codificação RGB (sigla para *Red*, *Green*, *Blue*) é a mais comum e define três bytes para armazenar, respectivamente, o vermelho, o verde e o azul, cada uma sendo um inteiro na faixa de 0 a 255. Outros padrões de representação são HLS (sigla para *Hue*, *Lightness*, *Saturation*), HSB (sigla para *Hue*, *Saturation*, *Brightness*), HSV (sigla para *Hue*, *Saturation*, *Value*), Hunter Lab, CIE 1976 Lab e CMYK (sigla para *Cian*, *Magenta*, *Yelow*, *Black*), este último utilizado em mídias impressas.

Embora uma imagem *bitmap* seja armazenada na RAM com todos os *pixels* é comum, por uma questão de economia de espaço e tempo de trasmissão, a compressão destes arquivos. Entre todos os formatos de compressão os mais conhecidos são o GIF (*Graphics Inter-change Format*), o JPEG (*Joint Photographic Experts Group*) e o PNG (*Portable Network Graphics*).

Neste trabalho as imagens sempre serão *bitmap*, com as cores codificadas no padrão RGB e comprimidas no formato JPEG. As imagens serão tratadas como equações, notacionadas na forma $f(x, y)$, onde x e y são inteiros e indicam a posição de um *pixel* específico, e os pixels são interpretados como tuplas na forma (r, g, b), onde r, g e b pertencem ao subintervalo inteiro de 0 a 255 e representam as cores vermelha, verde e azul respectivamente.

2.2 Simplificação de imagens e extração de características

Qualquer método de agrupamento depende sensivelmente do critério de semelhança adotado nas comparações entre os elementos, será esse critério que, basicamente, determinará a classe de cada elemento. O critério de semelhança deve ser baseado em alguma carac-

terística mensurável e comparáveis entre sí, ou seja, deve haver uma forma de se estabelecer a distância entre diferentes valores desta característica. Esta distância determinará a semelhança entre os elementos, onde quanto mais próximos mais semelhantes.

Em imagens existem diversas características que servem como critérios de semelhança, do ponto de vista da percepção humana, estas características são comumente ligados as cores, texturas ou formas presentes na imagen, ou ainda, a uma combinação delas. Em relação as cores, medidas de histograma são as mais populares; em texturas é comum a utilização de momentos do histograma de brilho, matriz de co-ocorrência, granulometria e informações do aspectro de Fourier; para formas se destacam os algorítimos de detecção de formas de interesse e os momentos invariantes; uma abordagem mista de cores e formas é possível atravéz de modelos de misturas gaussianas.

2.3 Momentos invariantes como descritores de imagens

Supondo que uma forma particular estaja presente numa imagem A, e que outra bastante parecida esteja presente numa imagem B, e que em relação a A a forma em B está invertida, ou, para o exemplo ficar mais claro, que esta forma seja a silhueta de um rosto, que em A está virado a esquerda e em B virado a direita, como indicado na Figura 2, é um objetivo particular deste trabalho que ambas as imagens possuam descritores (características extraídas) bastante semelhantes, senão identicos; afinal, em termos perceptivos, ou seja, em termos de significado que um observador atribui as imagens, neste exemplo A e B, ambas possuem a figura de um rosto e estar cada um virado numa direção é uma caracterrística marginal e não deve influenciar no agrupamento.

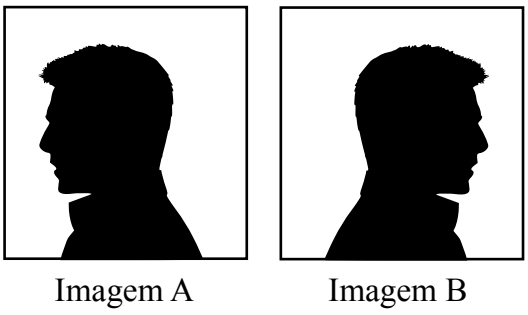

Figura 2: Duas imagens idênticas porém espelhadas.

O mesmo pode ser dito para rotação, traslação e escala de formas em diferentes ima-

gens, o que se deseja é a forma em sí, algo como seu protótipo, independente destas trasformações, como indica a Figura 3. A pretenção é, ao se descartar estas transformações, simular o que aparentemente é o comportamento natural de um indivíduo ao, sem ajuda do computador, categorizar e agrupar imagens.

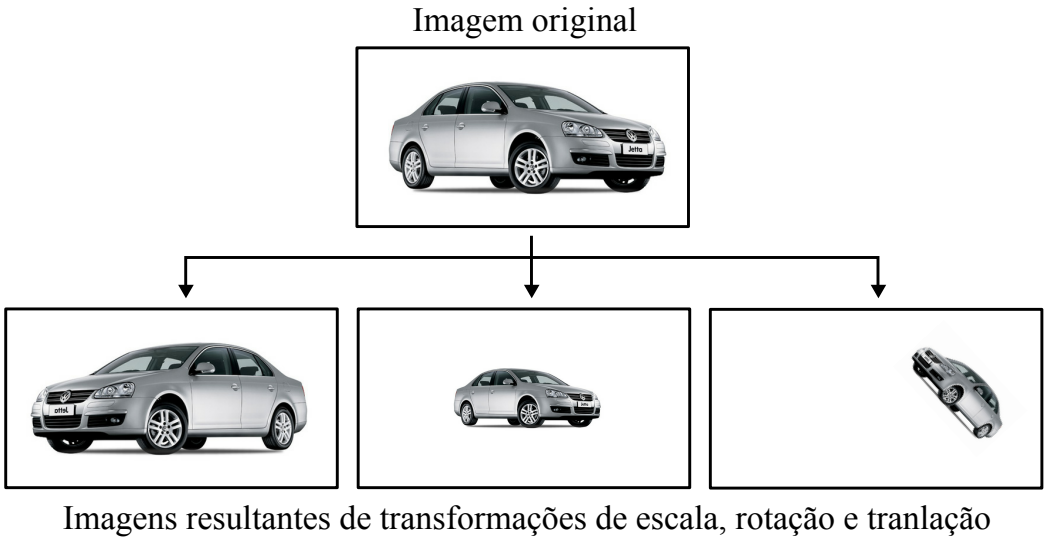

Figura 3: Exemplos de transformações de rotação, translação e escala sobre uma imagem, e como elas não alteram a essência das formas presentes na imagem original.

Um conjunto de descritores atende aos propósitos indicados acima, são os descritores de Hu, mais comumente chamados de momentos invariantes. Os momentos invariantes são um conjunto de sete descritores reais que independem de rotação, translação ou escala, isto é, quando aplicados a uma forma qualquer retornará os mesmos valores se aplicado a outra forma resultande de uma das três transformações citadas, ou até mesmo de uma combinação delas.

2.3.1 Formulação matemáticas dos momentos invariantes

Passemos então agora para formalização matemática desses momentos.

O momento bidimensional de ordem $(p+q)$ é dado pela equação 2.1:

$$m_{pq} = \iint x^p y^q f(x,y)\mathrm{d}x\mathrm{d}y, p, q \in \tag{2.1}$$

A equação num domínio discreto, pode ser reescrita na forma:

$$m_{pq} = \sum_{x,y} x^p y^q f(x,y), p, q \in \tag{2.2}$$

A massa total da função $f(x,y)$ é determinado pelo momento m_{00}, conforme a equação 2.3:

$$m_{pq} = \sum_{x,y} f(x,y), p, q \in \tag{2.3}$$

Existe um ponto no qual a aplicação pontual da massa total gera o mesmo momento que a massa distribuída, este ponto é dito centroide de $f(x,y)$ e suas coordenadas x e y são dadas pela equação 2.4:

$$\bar{x} = \frac{1}{m_{00}} \sum x f(x,y) = \frac{m_{10}}{m_{00}} \tag{2.4a}$$

$$\bar{y} = \frac{1}{m_{00}} \sum y f(x,y) = \frac{m_{01}}{m_{00}} \tag{2.4b}$$

O momento central é obtido se deslocando a imagem para o centroide, da seguinte forma:

$$\mu_{pq} = \sum_{x,y} (x - \bar{x})^p (y - \bar{y})^q f(x,y) \tag{2.5}$$

Ainda é necessário normalizar o momento para que os valores resultantes não sejam extremos a ponto de serem ignorados pelo sistema de reconhecimento de padrões. O momento central de ordem $(p+q)$ normalizado é obtido dividindo o momento central de y mesma ordem por um fator definido por μ_{00}^{γ} , conforme indicado pela equação 2.6:

$$\gamma = 1 + \frac{p+q}{2} \tag{2.6a}$$

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{2.6b}$$

A partir dessas equações são estabelecidos sete momentos invariantes à translação, rotação e escala, chamados de momentos de Hu, ou descritores de Hu. São eles:

$$\varphi_1 = \eta_{20} + \eta_{02} \tag{2.7a}$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{2.7b}$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.7c}$$

$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \tag{2.7d}$$

$$\begin{aligned} \varphi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ &+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{aligned} \tag{2.7e}$$

$$\begin{aligned} \varphi_6 &= (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\ &+ 4\eta_{11}(\eta_{30} - \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \tag{2.7f}$$

$$\begin{aligned} \varphi_7 &= (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ &+ (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{aligned} \tag{2.7g}$$

Observe que os momentos são definidos para um ponto de valor discreto, isto implica que devemos abandonar qualquer descriçao vetorial de cores, neste caso, devemos passar uma cor do formato RGB para seu tom de cinza. Neste trabalho o tom de cinza para uma cor RGB é o valor médio para os canais vermelhor, verde e azul.

2.4 Binarização de imagens

Mesmo que os momentos invariantes sejam, a princípio, bons descritores, eles não podem ser extraídos sem que a imagem tenha passado por algumas transformações. Estas transformações não são obrigatórias, isto é, não são restrições necessárias a aplicação dos momentos, mas são transformações que fazem sentido no processo de agrupamento, mais especificamente, no subprocesso de extração de características relevantes.

É perfeitamente válido supor que nem todas os *pixels* de uma imagem são relevantes, ou no mínimo, que determinados *pixels* são mais relevantes que outros, estes *pixels* mais relevantes podem ser interpretados como regiões de interesse, isto é, regiões que despertam maior atenção dos observadores. Em suma, podemos dividir a imagem em duas regiões, uma de interesse chamada de primeiro plano (*foreground*) e outra que pode ser negligenciada chamada plano de fundo (*background*), como no exemplo da Figura 4. A separação entre essas duas regiões é chamado de limiarização, ou ainda, remoção de fundo.

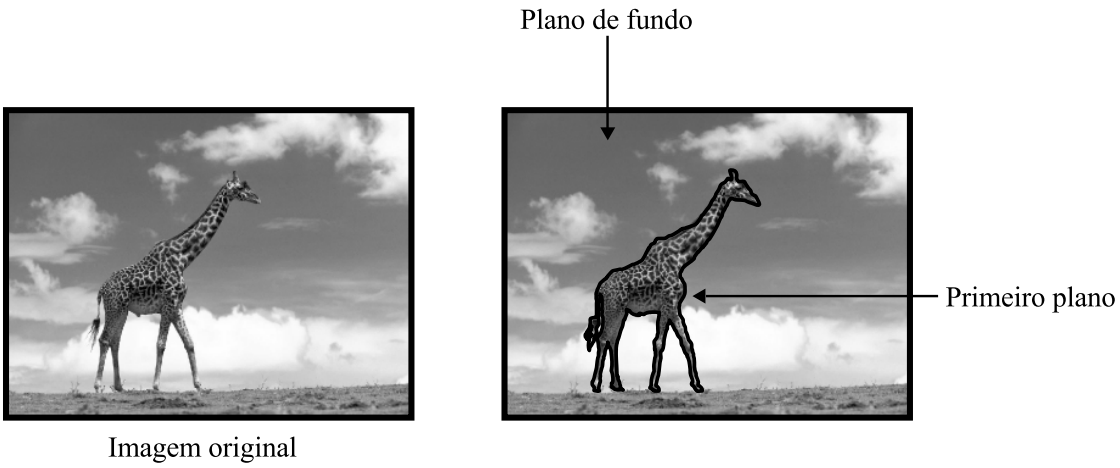

Figura 4: Imagem com o primeiro plano e o plano de fundo destacados.

Extrair da imagem os momentos invariantes apenas do primeiro plano torna os descritores mais interessantes para classificação, afinal, os valores ficam restritos apenas a região de maior interesse, sendo o plano de fundo totalmente ignorado na extração destas características.

Outro ponto a ser considerado, como visto na seção anterior, é que a extração dos momentos depende da intensidade de cada *pixel*, de modo que uma variação na intensidade de um *pixel* interfere no resultado dos momentos. Como agora apenas o primeiro plano é aplicado na extração, apenas as variações de intensidade nesta região são consideradas; contudo, estas variações podem em determinadas ocasiões gerar momentos muito distintos para regiões que, morfologicamente, são bem parecidas. Suponha o caso de, por exemplo, duas imagens que no primeiro plano apresentam a figura de uma flor, como na Figura 5, na primeira a flor tem a coloração clara, e na segunda escura, ao aplicar o momento sobre estas duas imagens, mesmo que tenham uma forma bem parecida, teremos resultados significativamente diferentes para os momentos das duas. É desejável eliminar este tipo de discrepância, isto é possível tornando todas as informações da primeiro plano homogêneas, ou seja, fazer com que cada *pixel* do primeiro plano tenha o mesmo peso para extração dos momentos. Esta homogenização sobre uma imagem já limiarizada é chamada de binarização, isto porque teremos duas regiões, uma irrelevante onde cada *pixel* terá o valor nulo, e outra relevante onde cada *pixel* terá seu máximo valor.

IMAGEM AQUI.

Figura 5: Imagens visualmente semelhantes mas com relativa diferença nos valores dos momentos devido as grande diferença de tons no primeiro plano.

Binarizar uma imagem, o que implicitamente também implica em limiarizá-la, é um processo bem simples e pode ser feito apenas como base no histograma. O que se deseja é anular todos os *pixels* abaixo de um limiar e potencializar os que estão acima dele. Como indicado no Algoritmo 1:

Algorithm 1: Binarização de uma imagem

Entrada: $f(x, y)$, l

início

  para cada $p \in f(x, y)$ **faça**

    se $p < l$ **então**

      $p \leftarrow 0$

    senão

      $p \leftarrow 255$

    fim se

  fim para cada

fim

Contudo, o Algorítimo 1 não indica como definir o limiar ótimo, isto é, aquele que melhor separa o primeiro plano do plano de fundo, esta operação é realizada, neste trabalho, através do método de Otsu, descrito na próxima seção.

2.5 Método de Otsu

O método de Otsu é um método de *thresholding* global, isto é, o valor obtido é uma constante, para escolha do melhor limiar. A base deste método é sua interpretação do histograma como como uma função de densidade de probabilidade discreta, da seguinte maneira:

$$p_r(r_q) = \frac{n_q}{n}, q = 0, 1, 2, ..., L - 1 \tag{2.8}$$

Onde:

- n é o total de *pixels* da imagem;
- n_q é o total de *piixels* que tem intensidade r_q e
- L é o total de níveis de intensidade na imagem.

O método de Otsu escolhe o limiar de valor k, tal que k é um nível de intensidade

que divide o histograma em duas classes $C_0 = [0, 1, ..., k-1]$ e $C_1 = [k, k+1, ..., L-1]$, e que maximise a variância σ_B^2 definida como:

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \tag{2.9}$$

Sendo:

$$\omega_0 = \sum_{q=0}^{k-1} p_q(r_q) \tag{2.10a}$$

$$\omega_1 = \sum_{q=k}^{L-1} p_q(r_q) \tag{2.10b}$$

$$\mu_0 = \sum_{q=0}^{k-1} \frac{qp_q(r_q)}{\omega_0} \tag{2.10c}$$

$$\mu_1 = \sum_{q=k}^{L-1} \frac{qp_q(r_q)}{\omega_1} \tag{2.10d}$$

$$\mu_T = \sum_{q=0}^{L-1} qp_q(r_q) \tag{2.10e}$$

O resultado da binarização com limiar ajustado segundo o método de Otsu pode ser observado na Figura 6

IMAGEM AQUI.

Figura 6: Imagem binarizada com limiar definito pelo método de Otsu.

2.6 Resumo do processo de extração de características

Como discutido nas seções anteriores deste capítulo, os momentos invariantes foram eleitos como os descritores a serem utilizadas para determinar a similaridade entre as imagens, contudo, estes descritores são extraídos somente após as imagens terem passado por determinadas transformações que visam simplificá-las e potencializar as regiões de maior interesse, e assim, produzir valores mais significativos para os momentos. As transformações aplicadas as imagens são a dessaturação e a binarização, nesta ordem.

Podemos resumir visualmente o processo de extração de característica na Figura 7:

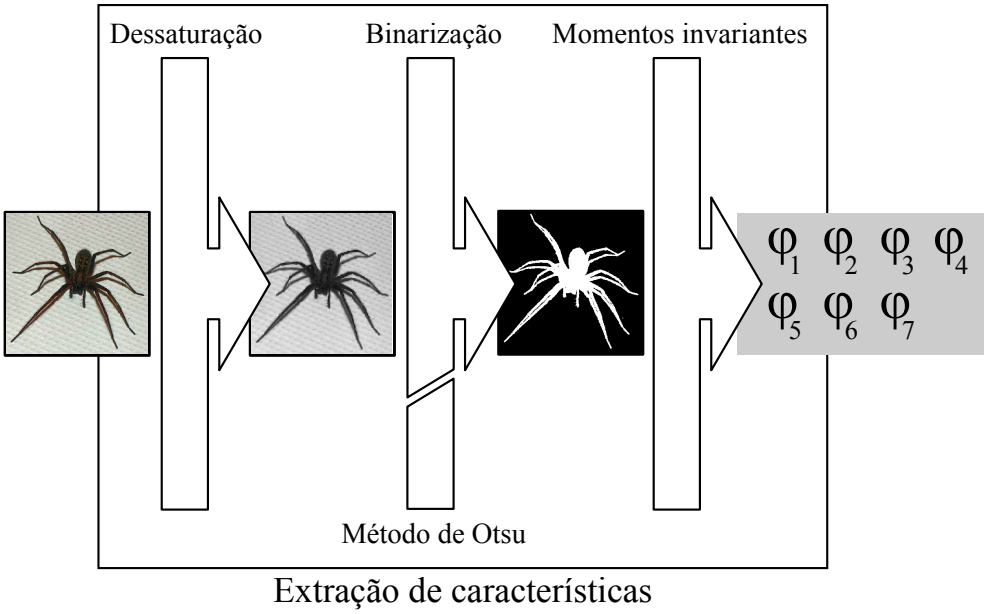

Figura 7: Esquema do processo de extração de caracteristicas.

3 Rotulação das imagens

TEXTO AQUI.

3.1 Mapas auto organizáveis

A maneira mais intuitiva de se agrupar imagens, ou qualquer outro tipo de informação, é estabelecer um determinado número de classes e mapear cada imagem para uma das classes, de modo que imagens semelhantes pertençam a mesma classe. Entretanto, este tipo de abordagem desconsidera graus diferentes de semelhança intra-classes, e até mesmo, graus diferentes de semelhança inter-classes, isto é, havendo somente a ligação das imagens com suas classes como será possível determinar, numericamente, o quão semelhante duas imagens são? Ou até mesmo, quão diferentes são duas classes uma da outra?

Estas perguntas são relevantes porque, para os casos onde uma classe possui centenas de imagens, muitas vezes o que se deseja é apenas uma amostra significativa da classe, ou em outra situação, estando em pesse de uma determinada imagem, deseja-se um número definido de imagens similares. Por isto, um método voltado exclusivamente para agrupamentos pode não fornecer um conjunto adequado de parâmetros que permitam extrair das classes informações como as necessárias para responder as questões acima.

Partindo de um outro ponto, em vez de iniciar o projeto do *clusterig* pelas classes, mas sim pela distância entre as imagens, surge a possibilidade de criar espaços para posicionar as imagens e, tendo a distância como medida de diferença, definir as classes como regiões ou intervalos dentro destes espaços. Deste modo, as classes não serão um parâmetro para a classificação, elas serão definidas automaticamente pela dispersão das imagens nestes espaços através de um processo dinâmico e automático. O resultado será não somente um método que agrupa imagens mas que também define, sem a participação ativa do usuário, as próprias classes utilizadas no agrupamento.

Estes espaços onde as imagens serão posicionadas podem ser de qualquer dimensão,

contudo, um espaço bidimensional de intervalos discreto é o suficiente para os propósitos deste trabalho. Outro ponto importante é que estes espaços não podem ser infinitos, pois obviamente estão limitados pela memória e pela capacidade de processamento do computador, por isto, serão utilizados sempre espaços limitados. Em resumo, podemos chamar estes subespaços bidimensionais discretos de mapas, onde cada ponto é uma posição potencial para uma imagem, e as imagens estão tão próximas quanto forem semelhantes.

Em resumo, a localização espacial de cada imagem e sua vizinhança topológica representam, em um domínio ou característica particular, nestes casos os momentos invariantes, uma classe, e estas classes são construídas de forma emergente, sem nenhuma comparação com padrões desejados, por isto são ditos auto organizáveis, como indica a Figura 8.

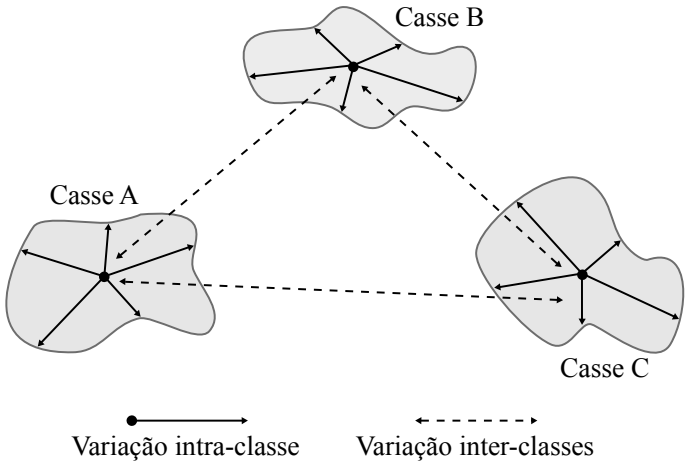

Figura 8: Organização das classes em um mapa auto organizável, coesão interna entre os elementos e isolamento externo entre as classes.

3.2 Redes de Kohonen como mapa auto organizável

Dentro da Inteligência artificial, mais especificamente no contexto do aprendizado de máquina, as redes neurais artificiais são sistemas computacionais inspirados na estrutura do cérebro, em particular dos neurônios, que adquirem conhecimento através da experiência.

As redes neurais se assemelham a grafos direcionados, onde os nós são os neurônios, ou unidades de processamento, que possuem uma quantidade indefinida de conexões de entrada e saída. As conexões são o equivalente às arestas do grafo, e são responsáveis por transmitir informações entre os neurônios, podendo amplificar ou reduzir a acuidade destas informações. A Figura 9 apresenta o esquema típico de um neurônio.

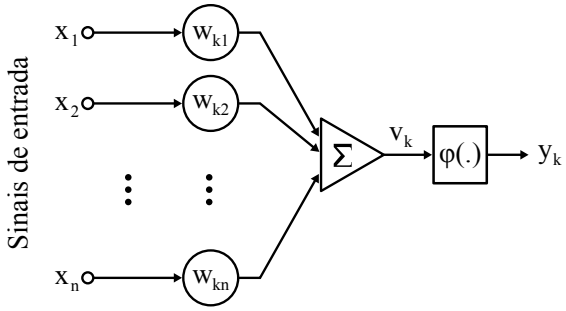

Figura 9: Modelo de neurônio.

Sinais de entrada provenientes de fora da rede chegam por meio de conexões originadas do mundo externo, de modo semelhante, saídas da rede para o mundo externo são conexões que deixam a rede.

A configuração da rede, ou seja, os peso atual das conexões, determina como os dados de entrada irão ativar os diferentes neurônios e gerar um determinado resultado. Para grande maioria dos tipos de redes neurais, uma configuração particular é obtida através de um algoritmo de treinamento. O treinamento em geral busca reforçar as conexões que geram bons resultados e penalizar as que não geram.

As redes de Kohonen apresentam apenas duas camadas de neurônios, a camada de entrada e a de saída. A camada de saída é uma espécie de malha de neurônios não conectados entre si, mas amplamente conectados com os neurônios da camada de entrada, como indicado na Figura 10. Esta malha funciona como um mapa, onde para cada padrão de entrada apenas um neurônio é ativado, padrões semelhantes ativam neurônios dentro de uma mesma região da malha.

IMAGEM AQUI.

Figura 10: Representação de uma rede de Kohonen.

As redes de Kohonen possuem um algoritmo próprio de treinamento, dividido em três etapas; na primeira, chamado de processo competitivo, uma determinada entrada ativa apenas um neurônio da malha; na segunda, chamado de processo cooperativo, o neurônio escolhido estabelece uma vizinhança de neurônios que serão ajustados para, junto com ele, identificar padrões semelhantes ao que foi apresentado; e por fim, na terceira etapa, chamada de processo adaptativo, os pesos são atualizados com base no neurônio vencedor e na vizinhança topológica. Este algoritmo de treinamento é dito não supervisionado,

pois não depende de um par *(entrada, saída esperada)*, já que a própria rede estabelece como será a configuração dos resultados, o processo de auto organização da rede.

3.3 Características gerais de uma rede neural de Kohonen

Esta seção irá apresentar mais detalhadamente como é a configuração de uma rede de Kohonen e seu algoritmo de treinamento.

3.3.1 Topologia de uma rede de Kohonen

Como dito anteriormente, a rede de Kohonen apresenta apenas duas camadas de neurônios, a camada de entrada e a camada de saída. A camada de entrada deve possuir tantos neurônios quanto forem à quantidade de elementos do padrão de entrada. A camada de saída é uma grade de geometria livre, geralmente retangular, de neurônios que não estão ligados entre si, mas estão, cada um, ligados a todos os neurônios da camada de entrada. As conexões apresentam pesos para escalar o sinal enviado. O esquema conceitual de uma rede de Kohonen é demonstrado na Figura 11:

IMAGEM AQUI.

Figura 11: Esquema detalhado de uma rede de Kohonen.

3.3.2 Treinamento da rede

O treinamento requer que os pesos sinápticos sejam iniciados com valores bem pequenos, para que a rede não apresente inicialmente nenhuma configuração. Três processos são executados para cada entrada do conjunto de treinamento, o processo competitivo, o processo cooperativo e o processo adaptativo.

Processo competitivo

Quando uma entrada $x = [x_1, x_2, ..., x_n]^T$ é apresentado à rede, o neurônio da grade que melhor responder a este padrão será ativado, este neurônio é dito vencedor, e será recompensado ajustando-se seus componentes para mais próximo do vetor de entrada.

O critério escolhido para determinar o neurônio vencedor é a distância euclidiana entre o vetor de entradas e o vetor de pesos das sinapses do neurônio, como indicado na equação 3.1:

$$d_i(t) = \sqrt{\sum_{j=1}^{N} \left(x_j(t) - w_{ij}(t)\right)^2} \tag{3.1}$$

Onde:

- $d_i(t)$ é a distância euclidiana entre o vetor de pesos do neurônio i e o vetor de entradas na iteração t;
- i é o índice do neurônio da grade;
- j é o índice do neurônio de entrada;
- N é o número de entradas;
- $x_j(t)$ é o sinal de entrada na entrada j na iteração t;
- $w_{ij}(t)$ é o valor do peso sináptico entre o neurônio de entrada j e o neurônio da grade i na iteração t.

Processo cooperativo

Estudos biológicos indicam que ao ser excitado, um neurônio estimula seus vizinhos topológicos, de forma que quanto mais próximo um neurônio está do neurônio ativo, mais excitado pelo estímulo do neurônio ativo ele é. O processo cooperativo busca simular este mecanismo biológico.

Em termos matemáticos, o que se deseja é um parâmetro h_{ik} , dito *vizinhança topológica*, que indica o gral de cooperação entre o neurônio vencedor i e o seu vizinho k, que deve ser simétrico em relação ao neurônio k e deve decrescer constantemente com o aumento da distância l_{ik} , até que $\lim_{l_{ik}\to\infty} h_{ik} = 0$. A função gaussiana 3.2 atende a estas duas exigências:

$$h_{ik} = e^{\left(\frac{l_{ik}^2}{2\sigma^2}\right)} \tag{3.2}$$

O parâmetro σ é denominado *largura efetiva da vizinhança*, e deve diminuir a cada iteração, indicando uma tendência de especialização da rede. Neste trabalho o parâmetro σ é a equação 3.3:

$$\sigma(t) = \sigma_0 e^{t/\tau_l} \tag{3.3}$$

Onde:

- σ_0 é o valor inicial de σ;
- t é a iteração atual;
- τ_l é uma constante de tempo.

Processo adaptativo

O processo adaptativo atualiza os pesos sinápticos a cada iteração, levando em consideração o neurônio vencedor e a vizinhança topológica. O ajuste dos pesos deve decrescer com o tempo, para evitar que novos dados comprometam seriamente o conhecimento já adquirido, substituindo padrões já estabelecidos por novos. Algo semelhante ocorre com o cérebro humano, ao decorrer do envelhecimento o aprendizado vai se tornando mais difícil.

O ajuste Δw_{ij} que a sinapse entre o neurônio de entrada i e um neurônio da malha j deve sofrer é expresso pela equação 3.4:

$$\Delta w_{ij} = \eta(t) h_{ik}(t)(x_j - w_{ij}) \tag{3.4}$$

Onde $h_{ik}(t)$ é o parâmetro vizinhança topológica na iteração t , referente ao neurônio vencedor k . O parâmetro *taxa de aprendizagem* $\eta(t)$ é definido pela expressão 3.5:

$$\eta(t) = \eta_0 e^{t/\tau_l}, \eta_0 \in [0, 1] \tag{3.5}$$

Onde τ_l é uma constante de tempo.

Algoritmo geral de treinamento

O algoritmo 2 resume as três etapas anteriores e descreve todo o processo de treinamento de uma rede de Kohonen:

Algorithm 2: Treinamento de uma rede de Kohonen

Entrada: σ_0 , τ_l , η_0 e o valor do *erro*
início
  repita
    Calcular a *largura efetiva* $\sigma(t)$;
    Calcular a *vizinhança topológica* h;
    Calcular a *taxa de aprendizado* $\eta(t)$;
    para cada *conexão* **faça**
      Calcular Δw;
      Ajustar o arco;
    fim para cada
  até *distâncias auclidianas* $\leq$ *erro*;
fim

3.4 Normalização dos dados de entrada da rede

Neste ponto já deve estar claro que a entrada da rede será o conjunto dos sete momentos invariantes. Os momentos serão utilizados tanto na etapa de treinamento quanto na classificação das imagens propriamente dita. Contudo, da forma como os momentos são calculados ainda é necessário que eles passem por uma normalização com o propósito de equalizar os valores dos momentos segundo sua real contribuição para caracterização das imagens, isto é, para cada um dos conjuntos de momentos, sete no total, um ajuste é feito sobre cada valor conforme o gral de variação dos dados, isto porque, se os valores de um conjunto são muito parecidos, pode-se concluir que este conjunto não é uma característica forte para distinguir as diferentes classes de imagens, então deve ter uma influência menor na classificação que os demais com alta variação dos dados.

A normalização também ajusta a faixa dos valores dos momentos. Dado que os valores geralmente são bem pequenos, a normalização amplifica proporcionalmente todos eles e evita problemas envolvendo operações computacionais com números muito próximos de zero.

Sendo M a matriz com os valores originais dos momentos, na forma:

$$M = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{17} \\ m_{21} & m_{22} & \dots & m_{27} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \dots & m_{n7} \end{bmatrix} \tag{3.6}$$

A tranformação $N(M)$ que normaliza todos dos elementos de M é dada por:

$$N_{ij} = \frac{m_{ij} - \overline{H}_j}{\sigma_j} \tag{3.7}$$

Onde:

- m_{ij} é o momento da linha i coluna j de M;
- $\overline{H}_j$ é a média de todos os elementos da coluna j de M;
- σ_j é desvio padrão $\sqrt{\frac{\sum_{i=1}^{n}(m_{ij}-\overline{H}_j)^2}{n-1}}$ dos elementos da coluna j.

Por fim, cada linha da matriz N é enviada para rede tanto na etapa do treinamento quanto para classificação da imagem.

3.5 A rede neural de Kohonen e o agrupamento das imagens

O conjunto dos pesos sinápticos é o resultado obtido pelo processo de treinamento, eles determinarão a posição de cada imagem no mapa, deste modo, os momentos normalizados de cada imagem devem ser apresentados a rede numa segunda vez para serem classificados.

Contudo, a posição das imagens apenas não indica a que classe pertencem, e na verdade, até agora as classes ainda não foram determinadas, para isso é necessário a adição de outros formalismos que através das posições das imagens e dos pesos sinápticos identifiquem as classes e rotule cada imagem para uma delas.

Nas subseções abaixo serão apresentados dois conceitos que determinam as classes de uma rede de Kohonen, a matriz de distâncias unificadas e a transformada de *watershed*.

3.5.1 Matriz de distâncias unificadas

Tendo executado o treinamento da rede de Kohonen, pode parecer que a distância euclidiana entre os elementos mapeados é o único parâmetro para a identificação das classes, contudo, existe também uma distância do vetor de entrada para cada posição do mapa, isto implica que também há uma outra distância entre os neurônios além da distância euclidiana, uma distância que representa o grau de dificuldade de se classificar uma elemento, no caso as imagens, em outra posição diferente da que naturalmente lhe seria atribuída. Isto faz muita diferença porque, mesmo tendo as classes que serem formadas por elementos adjacentes, esta não é condição suficiente para identificação dos grupos, é possível que dois elementos próximos pertençam a duas classes distintas, para isto basta que a dificuldade de se classificar um desses elementos na posição do outro seja superior a um limite estipulado.

Fazendo uma análise visual destas considerações, seria como se o mapa possuísse campos de atração, formando vales e picos, sendo os vales as classes e os picos os seus limites, ou fronteiras, como indica a Figura 12.

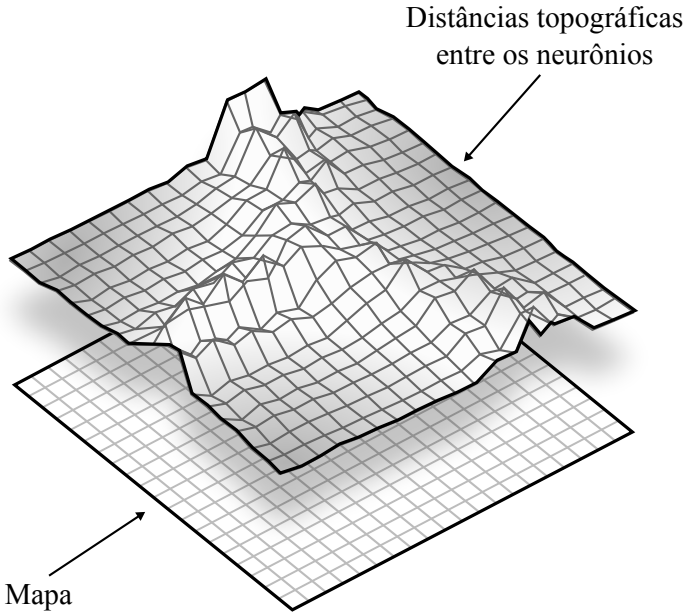

Figura 12: Representação da malha de distâncias topográficas entre os neurônios

O método denominado matriz de distâncias unificadas, ou simplesmente U-Matriz, tem o objetivo de identificar estas relações topológicas, definindo uma função tridimensional onde cada ponto do plano apresenta um valor de distância entre neurônios adjacentes, de modo que valores baixos correspondem a neurônios vizinhos semelhantes e valores al-

tos correspondem a neurônios vizinhos diferentes. Em termos matemáticos, regiões com baixos valores do gradiente, vales, são classes de neurônios especializados em padrões similares e regiões com valores altos correspondem a fronteiras entre as classes.

Considere o mapa retangular discreto limitado de tamanho $N \times M$, para cada neurônio p da camada de saída existe, na U-matriz, três distâncias, d_x, d_y e d_{xy}, em relação a seus vizinhos, como indicado na Figura 13.

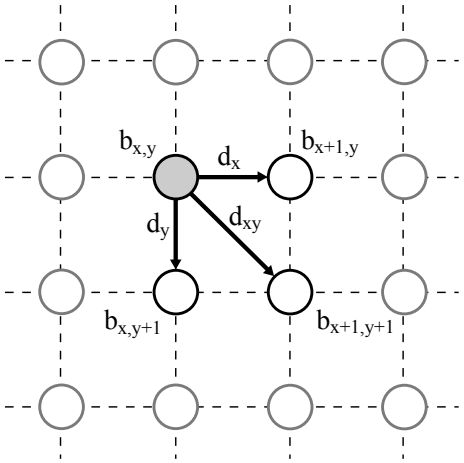

Figura 13: Distâncias d_x, d_y e d_{xy} entre o neurônio $b_{x,y}$ e seus visinhos.

Os valores de d_x, d_y e d_{xy} são calculados da seguinte maneira:

$$d_x(x,y) = \sqrt{\sum_i \left(w_{i(x,y)} - w_{i(x+1,y)}\right)^2} \tag{3.8a}$$

$$d_y(x,y) = \sqrt{\sum_i \left(w_{i(x,y)} - w_{i(x,y+1)}\right)^2} \tag{3.8b}$$

$$d_y(x,y) = \frac{1}{2\sqrt{2}} \sqrt{\sum_i \left(w_{i(x,y)} - w_{i(x+1,y+1)}\right)^2 + \sum_i \left(w_{i(x,y+1)} - w_{i(x+1,y)}\right)^2} \tag{3.8c}$$

Estes valores são inseridos em uma matriz de tamanho $(N-1) \times (M-1)$ de acordo com a seguinte tabela:

i	**j**	$\boldsymbol{U_{ij}}$
$2x+1$	$2y$	$d_x(x,y)$
$2x$	$2y+1$	$d_y(x,y)$
$2x+1$	$2y+1$	$d_{xy}(x,y)$
$2x$	$2y$	$d_u(x,y)$

Sendo $c = [c_1, c_2, ..., c_k]$ o vetor ordenado de elementos circunvizinhos com cardinalidade k, ainda levando em consideração um mapa retangular, o cálculo de d_u é obtido pela mediana dos valores circunvizinhos, do seguinte modo:

$$d_u(x,y) = \begin{cases} c_{(k+1)/2}, & \text{se } k \text{ for ímpar} \\ \frac{c_{k/2}+c_{(k+1)/2}}{2}, & \text{se } k \text{ for par} \end{cases} \tag{3.9}$$

Deste modo, a organização da U-Matriz é ilustrada pela Figura 14:

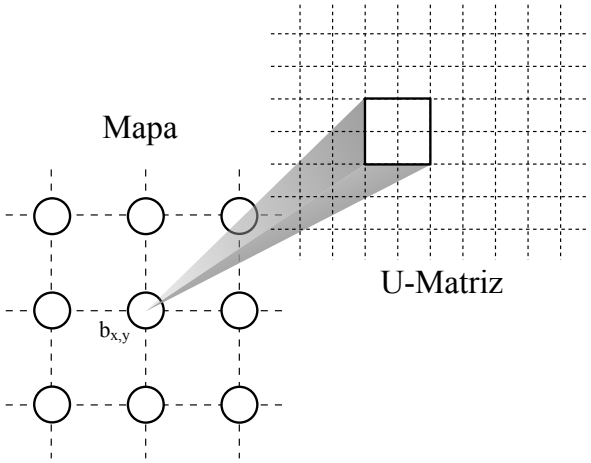

Figura 14: Organização da U-Matriz em relação a grade de neurônios.

3.5.2 Transformada *watershed* para rotulação automática da U-Matriz

A transformada de *watershed* estabelece as regiões e fronteiras das classes com base na U-matriz, faz isso apoiada no conceito intuitivo de inundação, vales e diques, da seguinte maneira:

Como visto na Seção 3.5.1, a U-matriz pode ser interpretada como uma superfície que contem vales e montanhas, considerando uma inundação, a água escorre pelas montanhas até os vales, que se inundam com o tempo, formando bacias. Em um determinado momento certas bacias tenderão a se unir, esta união é impedida pela "construção"de diques entre as regiões de fronteira. Ao fim do processo, ou seja, quando a água chegar ao nível da maior montanha, os diques formarão as fronteiras e as bacias formarão as classes. A Figura 15 Ilustra esse processo:

IMAGEM AQUI.

Figura 15: Legenda.

Uma definição formal da transformada de *watershed* passa pela consideração de dois períodos, isto é, pelo processo de expansão das bacias, em outros termos, pela elevação de um nível h para $h+1$.

Uma bacia associada a um mínimo m é denominada de $B(m)$, os pontos dessa bacia que possuem altitude menor ou igual a h são denominados $B_h(m)$, isto é:

$$B_h(m) = \{p \in B(m) | f(p) \leq h\} \tag{3.10}$$

O subconjunto de todas as bacias que possuem pontos com altitude menor ou igual a h é denominado $X(h)$, formalmente:

$$X(h) = \bigcup_i B_h(m_i) \tag{3.11}$$

Junto a esses conceitos fundamentais, o conjunto de todas os pontos que pertencem ao mínimo regional m_h de elevação h é denominado $R_{min_h}(f)$. Esta região será definida posteiormente ainda nesta seção.

Considerando que os primeiros pontos a serem inundados são os pontos mínimos h_{min}, podemos aplicar a Equação 3.11 da forma:

$$X(h_{min}) = R_{min_{h_{min}}}(f) = T_{h_{min}}(f) \tag{3.12}$$

Onde T obedece a relação:

$$T_h(f(x)) = \begin{cases} x, & \text{se } h \leq f(x) \\ 0, & \text{qualquer outro} \end{cases} \tag{3.13}$$

Utilizando h_{min} como ponto de partida, agora é necessário avançar para o estágio onde o nível sobe uma unidade, ou seja, para h_{min+1}. Neste ponto três situações podem ocorrer, isoladas ou simultaneamente, 1) um novo mínimo será encontrado no ponto h_{min+1} e formará uma nova bacia, 2) ocorrerá uma expansão da bacia da região cujo mínimo é h_{min} e 3) duas ou mais bácias distintas de nível h_{min} estão se expandindo e se encontrarão

juntas. Estas três situações são ilustradas na Figura 16:

IMAGEM AQUI.

Figura 16: Legenda.

1) $Y_1 \cap X_{h_{min}} = \emptyset$: Nenhuma bacia foi formada, o que ocorrerá apenas no próximo avanço de nível. Neste caso, vale a relação:

$$\forall p \in Y_1 \begin{cases} p \notin X_{h_{min}} \Rightarrow & f(p) \geq h_{min} + 1 \\ p \in Y_1 \Rightarrow & f(p) \leq h_{max} \end{cases} \tag{3.14}$$

2) $Y_2 \cap X_{h_{min}} \neq \emptyset$ e é conectado: Neste caso a inundação já atingiu o mínimo de Y_2 e o processo se encaminha numa expansão da bacia, o que pode ser descrito como:

$$Y_2 = B_{h_{min}+1}(Y_2 \cap X_{h_{min}}) = Z_{Y_2}(Y_2 \cap X_{h_{min}}) \tag{3.15}$$

Onde $Z_{Y_2}(Y_2 \cap X_{h_{min}})$ é a zona de influência geodésica de $Y_2 \cap X_{h_{min}}$ contida em Y_2. Esta zona de influência geodésica $Z_A(K_i)$ de um componente conectado K_i dentro de um conjunto A é o lugar geométricos dos pontos de A que a distância geodésica para K_i é a menor que a distância geodésica para qualquer outro ponto de A, em outros termos:

$$Z_A(K_i) = \{p \in A, \forall j \in [1, N] - \{i\}, d_A(p, K_i) < d_A(p, K_j)\} \tag{3.16}$$

Onde a distância geodésica $d_A(p, q)$ entre dois pontos p e q pertencentes a A é o menor caminho entre todos os caminhos possíveis de pontos, também pertencentes a A, que ligam p e q.

3) $Y_3 \cap X_{h_{min}} \neq \emptyset$ e não é conectado: Neste caso Y_3 contém dois ou mais mínimos e eles estão expandindo juntos, denotados de $M_1, M_2, \ldots, M_k$. Sendo M_i uma destas regiões a melhor aproximação para $B_{h_{min+1}}(M_i)$ corresponde a zona de influência geodésica de M_i dentro de $Y_3(M_Y)$:

$$B_{h_{min+1}}(M_i) = Z_Y(M_i) \tag{3.17}$$

Como em 2) e 3) são bacias que estão em expansão, podemos definir estas regiões em termos de uma única zona de influência geodésica $X_{h_{min}}$, deste modo $X_{h_{min+1}}$ é definido

como a união destas zonas de influência geodésicas onde os mínimos regionais foram os mais recentemente descobertos, formulado em termos de uma recursão:

$$\begin{cases} X_{h_{min}} = \bigcup_i h_{min_i} \in f \\ X_{h+1} = R_{min_{h+1}}(f) \cup Z_{T_{t \leq h+1(f)}}(X_h) \end{cases} \tag{3.18}$$

Por fim, o conjunto de bacias encontradas na U-matriz representam as classes e servirão para rotular as imagens, para cada mínimo local haverá uma bacia e com isso uma classe. Como o conjunto de mínimos locais pode ser grande, existe então a chance de uma sobre segmentação da U-matriz, por isso é conveniente restringir a quantidade de mínimos locais, no caso, os mínimos locais que geram bacias com regiões muito pequenas, logo, é comum a utilização de filtros gaussianos sobre a U-matriz antes de se aplicar a transformada de *watershed* para regularizá-la.

3.6 Resumo do processo de classificação das imagens e do *clustering* como um todo

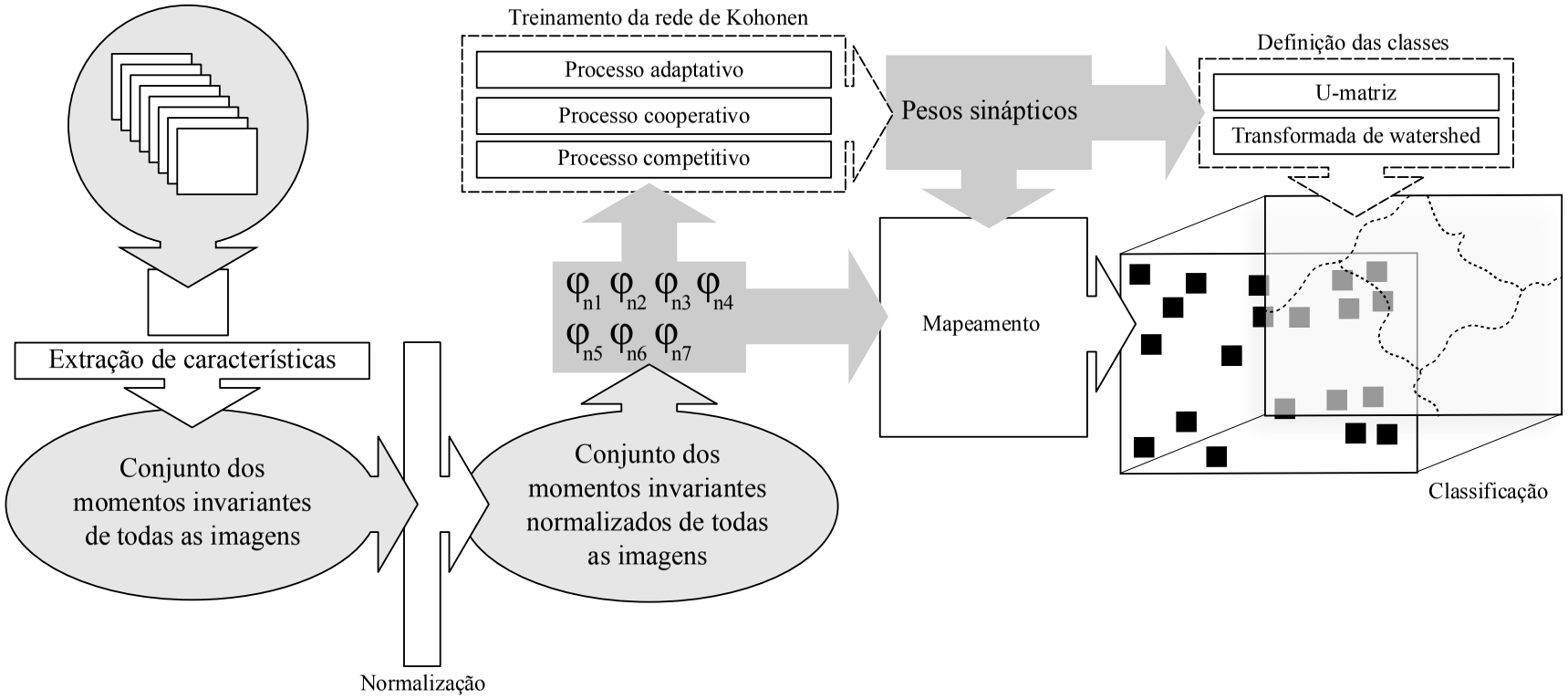

Figura 17: Legenda.

4 Testes empíricos e análise dos resultados

TEXTO AQUI.

5 Conclusões

TEXTO AQUI.

Referências Bibliográficas