



M.I.C.E.

My Image Classification Environment (beta-2)

Rogério Galante Negri

2017 (beta), 2020 (beta-2) Rogério Galante Negri

<https://github.com/rogerionegri/MICE>

República Federativa do Brasil – Ministério da Economia – Instituto Nacional da Propriedade Industrial – Diretoria de Patentes, Programas de Computador e Topografias de Circuitos Integrados – Certificado de Registro de Programa de Computador - Processo No: BR512018052367-6. Dados referentes a versão beta (2017).

O desenvolvimento deste sistema contou com o suporte financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (Processos 2014/14830-8 e 2018/01033-3).

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Conteúdo

I	M.I.C.E.	
1	M.I.C.E.	9
1.1	O Software	9
1.2	Módulos	9
1.3	Itens de Interface	10
1.3.1	Opções gerais	11
1.3.2	Exibição de informações sobre os dados	12
1.3.3	Opções de classificação baseada em pixels/regiões	13
1.3.4	Opções de segmentação	14
1.3.5	Opções de simulação	14
1.4	Formatos de Dados	15
1.5	Requisitos, Instalação e Execução	16
II	Classificação, Segmentação e Simulação	
2	Classificação Baseada em Pixels	19
2.1	Métodos supervisionados	20
2.1.1	MLC – Maximum Likelihood Classifier	20
2.1.2	KNN – K-Nearest Neighbors	20
2.1.3	SVM – Support Vector Machine	21
2.1.4	Minimum Distance Classifiers	22

2.2	Métodos não supervisionados	23
2.2.1	K-Means	23
2.2.2	Expectation Maximization	24
2.2.3	Fuzzy C-Means	24
3	Classificação Baseada em Regiões	27
3.1	Métodos supervisionados	28
3.1.1	MSDC – Minimum Stochastic Distance Classifier	28
3.1.2	MASDC – Minimum Average Stochastic Distance Classifier	28
3.1.3	NNSDC – Stochastic Distance Nearest Neighbor Classifier	29
3.1.4	KNN – K-Nearest Neighbors	29
3.1.5	SVM – Support Vector Machine	29
3.2	Método não supervisionado	30
3.2.1	K-Means	30
4	Segmentação de Imagens	31
4.1	Segmentação orientada por bordas	31
4.1.1	Homogeneic Ramification	32
4.2	Segmentação orientada por regiões	32
4.2.1	Region Growing	32
5	Simulação de Imagens	33
5.1	Simulação Orientada à Phantom	33
5.1.1	Simulação Bootstrap	34
5.1.2	Simulação Gaussiana	34
5.2	Simulação de Objetos Aleatórios	35
5.2.1	Simulação Bootstrap	35
5.2.2	Simulação Gaussiana	35
5.3	Simulação para Detecção de Mudanças	35
6	Detecção Não Supervisionada	37
6.1	CVA – Change Vector Analysis	37
6.2	PCA-KM	38
6.3	GKC+SA	38
6.4	USFA – Unsupervised Slow Feature Analysis	39
6.5	ASF – Alternating Sequential Filtering	39
6.6	FR – Floating References	39
6.7	HBSC – Homogeneous Block Single-class Classification	40
	Bibliografia	41
	Livros	41

Artigos
Anais

41
43



M.I.C.E.

1	M.I.C.E.	9
1.1	O Software	
1.2	Módulos	
1.3	Itens de Interface	
1.4	Formatos de Dados	
1.5	Requisitos, Instalação e Execução	

1. M.I.C.E.

1.1 O Software

O software M.I.C.E. (*my Image Classification Environment*) é resultado da junção de diferentes técnicas de classificação, segmentação e simulação de imagens. A maioria das implementações que compõem esta ferramenta ocorreram no período de 2014 a 2017, em consequência de pesquisas sobre métodos de classificação de imagens de sensoriamento remoto¹. A maioria das implementações são suportadas por funções disponíveis na biblioteca SLIC (*Small Library for Image Classification*) [1].

Dentre algumas motivações que levaram ao desenvolvimento desta ferramenta, são citadas: (i) o oferecimento de uma ferramenta com utilidade no ensino e pesquisa; (ii) a construção de uma interface de fácil extensão quanto a inclusão de novos métodos de classificação, segmentação e simulação de imagens; (iii) o fornecimento de um conjunto organizado de técnicas de processamento de imagens com aplicações direcionadas ao sensoriamento remoto; (iv) a realização de procedimentos simples de manipulação de imagens e análise estatística de imagens.

1.2 Módulos

A Figura 1.1 apresenta uma organização geral dos módulos que compõe o software M.I.C.E.. Em um primeiro nível, estão expostas funcionalidades relacionadas a ferramentas gerais, classificação baseada em pixels, baseada em regiões, segmentação e simulação.

O módulo de ferramentas gerais é composto por um conjunto de funções para edição e visualização de imagens assim como para coleta, organização e análise estatística de amostras.

¹FAPESP – Processo 2014/14830-8: “Desenvolvimento de novas funções kernel para classificação de imagens de sensoriamento remoto”.

Distintos segundo as abordagens baseada em pixel e baseada em regiões, dois módulos constituem as funções de classificação de imagens. Em um nível intermediário, cada um destes módulos estão subdivididos segundo os paradigmas de aprendizado supervisionado e não-supervisionado, que por sua vez possuem diferentes métodos de classificação associados. Os Capítulos 2 e 3 limitam-se a uma breve discussão destes métodos com relação às abordagens baseada em pixel e baseada em regiões, respectivamente.

Dois métodos, discutidos no Capítulo 4, compõem o módulo de Segmentação.

Por fim, o módulo de Simulação oferece alternativas para geração de imagens sintéticas segundo abordagens orientadas por *phantom* e objetos aleatórios, cujo comportamento dos objetos pode ser simulado diante o uso de distribuições gaussianas multivariadas ou por amostragem aleatória (*bootstrapping*). Tais formas de simulação são discutidas no Capítulo 5.

1.3 Itens de Interface

A interface do software M. I. C. E. é composto por uma barra de opções gerais (menu geral), ilustrada na Figura 1.2, e cinco abas de opções, sendo uma delas (Image/Data – Figura 1.3) responsável por exibir informações gerais sobre a imagem e conjuntos de amostras disponíveis e as demais relacionadas a cada um dos módulos que estruturam a ferramenta (i.e., funções de classificação baseada em pixel e regiões – Pixel Based e Based, Simulação e Segmentação – Simulation and Segmentation), apresentadas nas Figuras 1.4 a 1.6. Uma descrição geral sobre cada uma destes itens de interface é realizadas nas subseções seguintes.

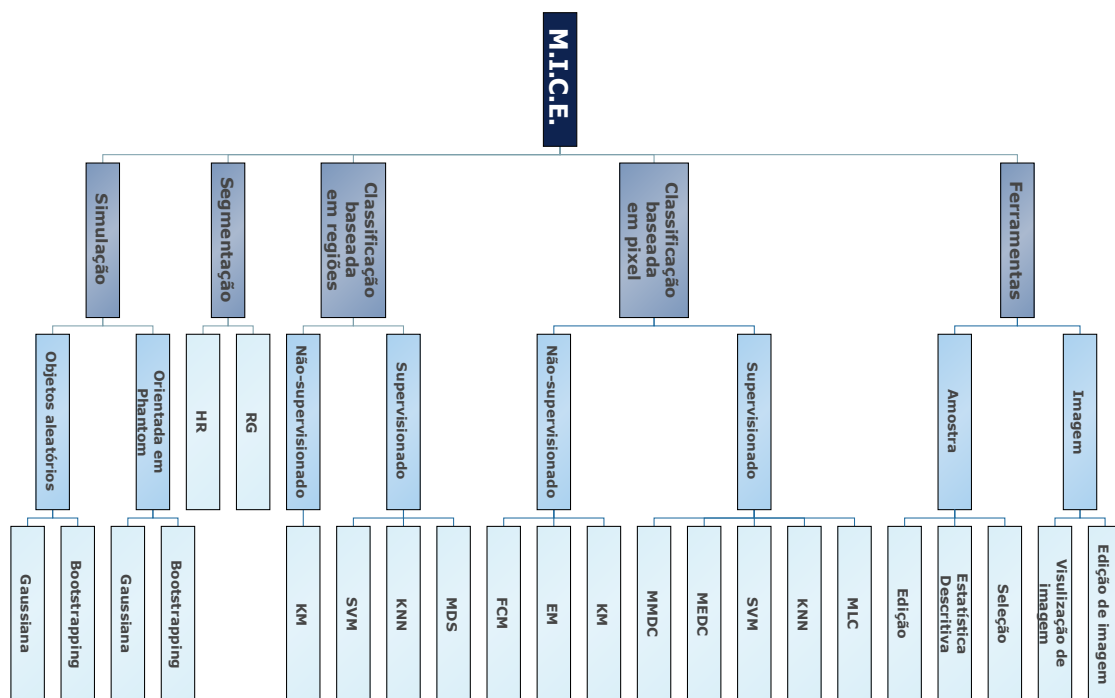


Figura 1.1: Organização do software M. I. C. E..

1.3.1 Opções gerais

A Figura 1.2 aponta os diferentes itens presentes no menu geral do software M.I.C.E., os quais são descritos individualmente a seguir.

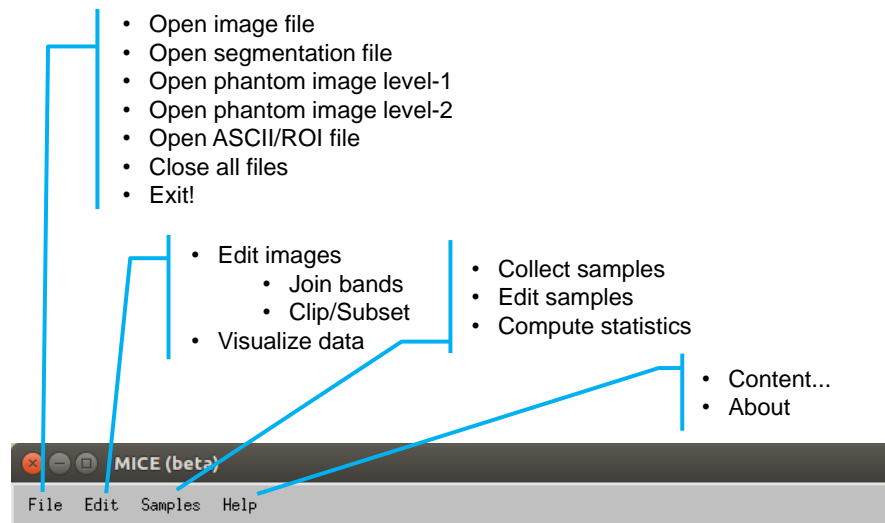


Figura 1.2: Menu de opções principais.

File: Opções para seleção de imagens, segmentações, phantoms em dois níveis hierárquicos, amostras relacionadas a objetos/alvos presentes na imagem selecionada, desassociação de arquivos (fechamento) e encerramento da execução do programa. As seguintes funcionalidades estão associadas a esta opção:

Open image file: Seleção de imagem base para os processos de classificação, segmentação e simulação;

Open segmentation file: Seleção de imagem segmentada, exigida pelos métodos de classificação baseada em regiões como meio de definir o suporte das regiões;

Open phantom image level-1/2: Seleção de imagem phantom necessária nos processos de simulação de imagens baseada em phantom. O nível 1 (level-1) determina uma associação direta entre cada objeto da imagem phantom e uma determinado alvo/classe previamente observada na imagem base e selecionada como amostra (ROI – Region of Interest). Possíveis variações intra-objeto são modeladas de forma opcional com auxílio da phantom de nível 2 (level-2);

Open ASCII/ROI file: Seleção de arquivos previamente obtidos e que contém amostras de objetos/alvos/regiões de interesse (ROI – Region of Interest) sobre a imagem base. Arquivos neste formato em questão podem ser obtidos via Menu→Samples→Collect samples;

Close all files: Opção destinada ao fechamento de arquivos (imagens e amostras) carregados e limpeza de memória;

Exit! Encerra a execução do software M.I.C.E..

Edit: Oferece funções de edição e visualização de imagens:

Edit imagens: Composta por Join bands e Clip/Subset, permite fundir arquivos de imagens distintos em um único arquivo assim como redimensionar ima-

gens em termos de espaciais (dimensão da imagem) ou espectrais (dimensão do espaço de atributos – número de bandas);

Visualize data: Permite visualizar uma imagem em escala de cinza ou pseudo-cor (quando composta por uma única banda) ou em composições coloridas RGB;

Samples: Disponibiliza ferramentas para coleta, edição e cálculos estatísticos sobre amostras identificadas sobre a imagem base. Em maior detalhe:

Collect samples: Oferece interface para visualização da imagem base e seleção de amostras sobre a mesma. Ainda, esta interface permite a associação de um nome e cor temática às amostras selecionadas;

Edit samples: Possibilita a edição de amostras, coletadas anteriormente pela opção *Collect samples*, em termos de troca de nome e cor temática assim como na junção (*Merge*) entre pares de amostras espacialmente disjuntas;

Compute statistics: Cálculo de estatísticas descritivas de amostras selecionadas na imagem base assim como medidas de separabilidade entre pares de amostras segundo a distancia de Bathacharyya.

Help: Itens de caráter informativos que fornecem:

Content...: Acesso a este manual;

About: Descrições sobre autoria, financiamento e suporte.

1.3.2 Exibição de informações sobre os dados

Os elementos identificados na Figura 1.3 compõem a aba de exibição de informações dos dados. Em maior detalhe:

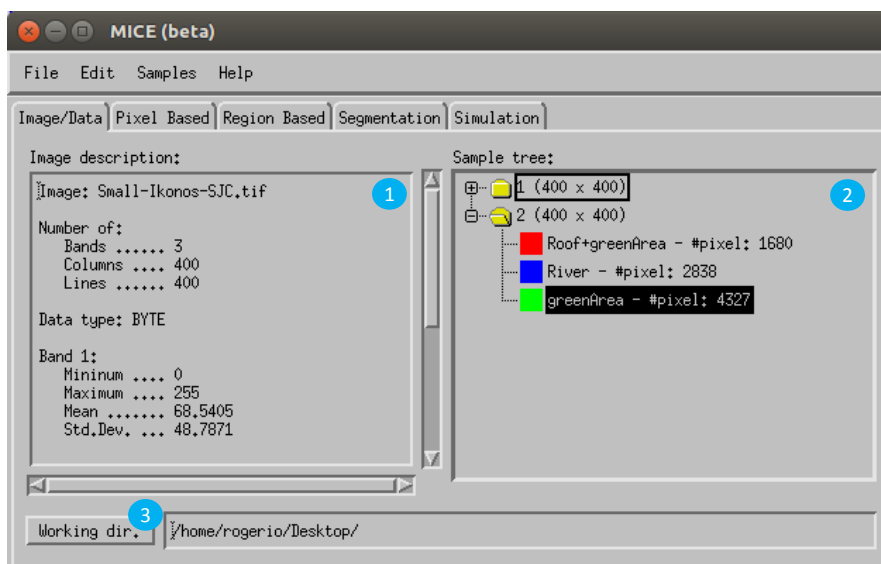


Figura 1.3: Aba de exibição de informações gerais (imagens e conjuntos de amostras disponíveis) e diretório de trabalho.

1. **Image description** apresenta dados superficiais (dimensões e estatísticas por banda) a respeito da imagem base previamente carregada pelo usuário;
2. **Sample tree** ilustra conjuntos de amostras carregadas pelo usuário. Tais conjuntos possuem uma denominação atribuída pelo usuário e contém as diferentes amostras

que os constituem, organizados por sua vez segundo respectivos nomes e cores temáticas;

3. **Working dir.** Permite definir um diretório alvo para onde são armazenados os resultados dos processamentos realizados.

1.3.3 Opções de classificação baseada em pixels/regiões

Os diferentes métodos de classificação baseada em pixels e regiões são disponibilizados nas abas Pixel Based e Region Based, as quais são análogas em termos de interface. Os elementos que compõem tais abas e respectivas finalidades são descritos abaixo:

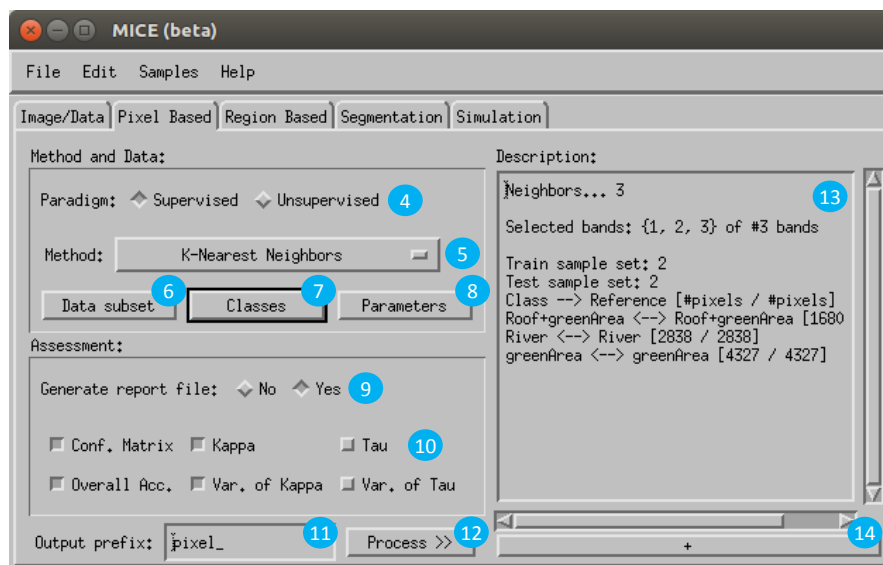


Figura 1.4: Aba de opções para classificação baseada em pixel (e analogamente, baseada em regiões).

4. **Paradigm** permite a seleção do paradigma de aprendizado dos métodos de classificação disponíveis;
5. **Method** uma vez selecionado o paradigma, diferentes métodos de classificação relacionados a abordagem de classificação em questão (aba) tornam-se disponíveis;
6. **Data subset** permite a seleção de todas ou parte das bandas presentes na imagem base;
7. **Classes** possibilita a seleção de todas ou parte das classes pertencentes a um dados conjunto de amostras disponível;
8. **Parameter** lança menu para ajuste de parâmetros do método definido no seletor Method;
9. **Generate report file** estipula a geração, ou não, de um relatório descritivo sobre o resultado de classificação obtido;
10. **Itens** que podem, ou não, compor o relatório descritivo;
11. **Output prefix** refere-se a um prefixo identificados que é incluído no início dos arquivos resultantes do processamento;
12. **Process** dispara o processamento;
13. **Description** consiste em um relatório dinâmico a respeito dos métodos, parâmetros e demais dados selecionados;

14. Expande o campo **Description**.

1.3.4 Opções de segmentação

As opções relacionadas aos métodos de segmentação, conforme Figura 1.5 são:

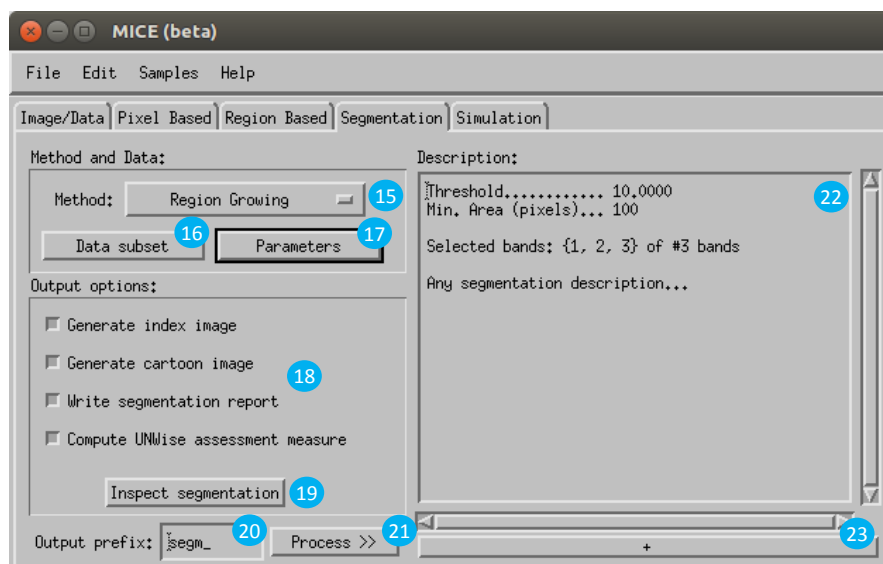


Figura 1.5: Aba de opções para segmentação de imagens.

15. **Method** permite a seleção de um dos diferentes métodos de segmentação disponíveis;
16. **Data subset** permite a seleção de todas ou parte das bandas presentes na imagem base;
17. **Parameter** lança menu para ajuste de parâmetros do método definido no seletor Method;
18. **Output options** oferece diferentes opções de produtos que o processo de segmentação pode gerar, os quais incluem: resultados de segmentação representados na forma de imagem de índices, imagem cartoon que exibe regiões a partir dos respectivos valores médios para cada uma das bandas selecionadas em Data subset, produzir um relatório de descritivo sobre o processo de segmentação e computar a medida UNWise que expressa a qualidade da segmentação obtida;
19. **Inspection segmentation** oferece uma ferramenta para visualizar segmentações previamente obtidas;
20. **Output prefix** refere-se a um prefixo identificados que é incluído no início dos arquivos resultantes do processamento;
21. **Process** dispara o processamento;
22. **Description** consiste em um relatório dinâmico a respeito dos métodos, parâmetros e demais dados selecionados;
23. Expande o campo **Description**.

1.3.5 Opções de simulação

Por fim, opções para geração de imagens sintéticas são disponibilizadas na aba Simulation, ilustrada na Figura 1.6. Os diferentes itens que compõem esta aba são:

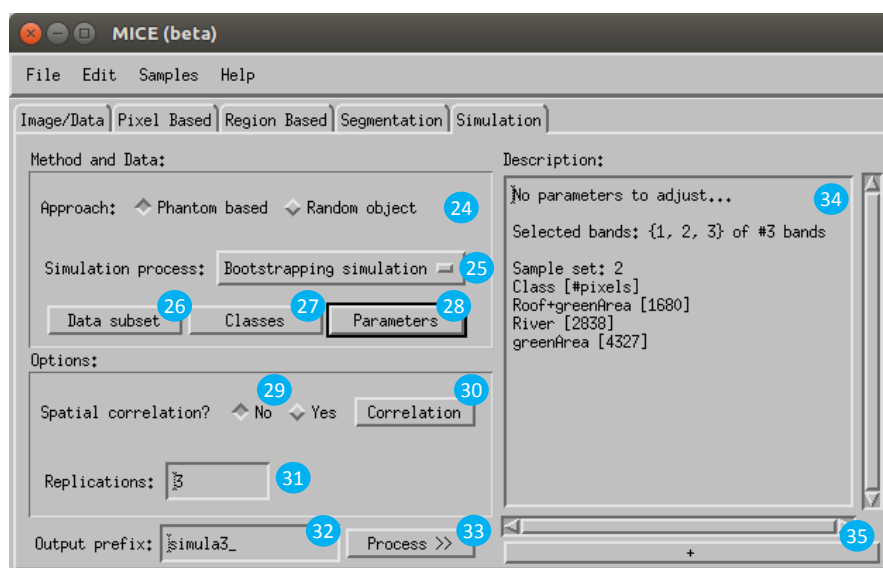


Figura 1.6: Aba de opções para simulação de imagens.

24. **Approach** permite a seleção da abordagem de simulação, com relação aos objetos presentes nas imagens simuladas;
25. **Simulation process** permite a seleção dos métodos para síntese do comportamento dos pixels associados aos diferentes objetos das imagens simuladas;
26. **Data subset** permite a seleção de todas ou parte das bandas presentes na imagem base;
27. **Classes** possibilita a seleção de todas ou parte das classes pertencentes a um conjunto de amostras disponível;
28. **Parameter** lança menu para ajuste de parâmetros do método definido no seletor Simulation process;
29. A inclusão de correlação espacial entre os pixels nas imagens simuladas é dependente da adesão a questão expressa por **Spatial correlation?**;
30. **Correlation** lança menu para ajuste dos pesos do filtro convolutivo empregado na modelagem da correlação espacial, caso a resposta em **Spatial correlation?** tenha sido positiva;
31. **Replications** permite definir a quantidade de imagens sintéticas a serem geradas;
32. **Output prefix** refere-se a um prefixo identificado que é incluído no início dos arquivos resultantes do processamento;
33. **Process** dispara o processamento;
34. **Description** consiste em um relatório dinâmico a respeito dos métodos, parâmetros e demais dados selecionados;
35. Expande o campo **Description**.

1.4 Formatos de Dados

Com intuito de garantir maior compatibilidade com demais ferramentas de processamento digital de imagens, o software M.I.C.E. adota o formato TIFF (*Tagged Image File Format* [2]) como padrão de representação de imagens de entrada assim como para os resul-

tados produzidos. A flexibilidade é outra justificativa associada à escolha deste formato, uma vez que não impõe limite a respeito da quantidade de bandas, possibilita considerar diferentes tipos de dados (e.g., *byte*, *integer*, *float*, *double*, *complex*, *double complex*, etc.), organizações estruturais (BIL, BIP e BSQ) e diversas outras opções/facilidades.

Além das imagens, elemento principal a ser processado, amostras que identificam alvos/objetos de interesse sobre tais imagens correspondem a um segundo tipo de informação com grande relevância perante o software M.I.C.E.. O formato para representação de amostras herda um dos padrões ASCII/ROI² estabelecidos pelo software ENVI [3]. A escolha deste padrão foi baseada em sua simplicidade de representação e edição além de permitir compatibilidade com o software ENVI.

1.5 Requisitos, Instalação e Execução

Para uso do software M.I.C.E., é sugerido o emprego de plataforma com 8 Gigabytes de memória RAM e processador Intel Core i5 ou equivalente. A execução em sistema operacional Linux é mandatória.

O processo de instalação consiste em copiar o conteúdo do diretório `instMICE` para um diretório local com permissão de acesso, acessar tal diretório e, através do comando `./MICE`, efetuar sua execução.

²ROI – *Region of Interest*

Classificação, Segmentação e Simulação

2	Classificação Baseada em Pixels	19
2.1	Métodos supervisionados	
2.2	Métodos não supervisionados	
3	Classificação Baseada em Regiões	27
3.1	Métodos supervisionados	
3.2	Método não supervisionado	
4	Segmentação de Imagens	31
4.1	Segmentação orientada por bordas	
4.2	Segmentação orientada por regiões	
5	Simulação de Imagens	33
5.1	Simulação Orientada à Phantom	
5.2	Simulação de Objetos Aleatórios	
5.3	Simulação para Detecção de Mudanças	
6	Detecção Não Supervisionada	37
6.1	CVA – Change Vector Analysis	
6.2	PCA-KM	
6.3	GKC+SA	
6.4	USFA – Unsupervised Slow Feature Analysis	
6.5	ASF – Alternating Sequential Filtering	
6.6	FR – Floating References	
6.7	HBSC – Homogeneous Block Single-class Classification	
	Bibliografia	41
	Livros	
	Artigos	
	Anais	

2. Classificação Baseada em Pixels

Um classificador é representado por uma função $F : \mathcal{X} \rightarrow \mathcal{Y}$, que associa elementos do espaço de atributos \mathcal{X} a uma das classes de $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$, com $c \in \mathbb{N}^*$, a partir de um dado rótulo (indicador) de classe em $\mathcal{Y} = \{1, 2, \dots, c\}$. Nestas condições, para $\mathbf{x} \in \mathcal{X}$ e $y \in \mathcal{Y}$, $y = F(\mathbf{x})$ indica que \mathbf{x} pertence à classe ω_y .

Classificação de imagem refere-se à aplicação de F sobre uma imagem \mathcal{I} , definida sobre um reticulado (suporte) $\mathcal{S} \subset \mathbb{N}^2$, a fim de gerar uma representação simplificada que identifica diferentes classes/objetos presentes em \mathcal{I} . De modo geral, é possível organizar os métodos de classificação de imagens segundo duas características fundamentais: *abordagem* e *tipo de aprendizado*.

Para os métodos cujo aprendizado é supervisionado, são fornecidos conjuntos de treinamento que contêm o conhecimento prévio a respeito das classes que compreendem o problema de classificação. A partir desse conhecimento são extraídos parâmetros necessários para a modelagem da regra de decisão do classificador, e então é definido um mapeamento capaz de relacionar padrões ainda não conhecidos a uma determinada classe. Em oposição, os métodos não supervisionados não usufruem de informações a respeito das classes que envolvem o problema de classificação. Nesse caso, o aprendizado é baseado em analogias captadas das relações entre os padrões da imagem.

A abordagem distingue os métodos de classificação de imagens entre os baseados em regiões e os baseados em pixels. Na classificação por métodos baseados em regiões, a imagem é inicialmente segmentada para que os alvos presentes nesta sejam delimitados, e em seguida classificados. Esta abordagem é discutida em maiores detalhes no Capítulo 3.

A classificação de imagens baseada em pixels consiste na aplicação de F sobre os vetores de atributos dos pixels (padrões) que compõem uma imagem \mathcal{I} , cujo resultado de classificação pode ser denotado por $F(\mathcal{I})$. Com relação à imagem em que é conduzido o processo de classificação, $\mathcal{I}(s) = \mathbf{x}$ denota que o pixel $s \in \mathcal{S}$ de \mathcal{I} possui atributos representados pelo vetor $\mathbf{x} \in \mathcal{X}$.

Os diferentes métodos de classificação de imagem propostos na literatura podem

ser entendidos como maneiras distintas de modelar a função $F : \mathcal{X} \rightarrow \mathcal{Y}$ e aplicá-la na classificação de \mathcal{S} . Para os métodos baseados em pixels cujo aprendizado é supervisionado, são utilizadas informações disponíveis em um *conjunto de treinamento* $\mathcal{D} = \{(\mathbf{x}_i, \omega_j) \in \mathcal{X} \times \Omega : i = 1, \dots, m; j = 1, \dots, c\}$ composto por $m \in \mathbb{N}^*$ vetores de treinamento. A notação (\mathbf{x}_i, ω_j) indica que \mathbf{x}_i está associado à classe ω_j .

Em oposição ao supervisionado, os métodos de aprendizado não supervisionado não usufruem de conjuntos de treinamento como fonte de informação, logo, não são capazes de definir um mapeamento entre um conjunto de padrões e um conjunto de classes. Neste caso o processo de aprendizado é baseado em analogias construídas ao observar todos os padrões de \mathcal{D} a ser classificado. Como resultado são gerados subconjuntos compostos por padrões similares, sem uma classe associada a estes. Nas Seções 2.1 e 2.2 são apresentados os diferentes métodos de classificação de imagens baseados em pixel disponíveis no software M.I.C.E. distintos segundo os paradigmas supervisionados e não-supervisionados, respectivamente.

2.1 Métodos supervisionados

2.1.1 MLC – Maximum Likelihood Classifier

Sejam \mathcal{S} e \mathcal{D} uma dada imagem e um conjunto de treinamento composto por exemplos de c classes diferentes, respectivamente. Admitindo distribuições de probabilidade $P(\mathbf{x}|\omega_k)$, com $k = 1, \dots, c$ que modelam as classes ω_k em \mathcal{D} , o Classificador de Máxima Verossimilhança (*Maximum Likelihood Classifier* – MLC) consiste em classificar os pixels s_i tais quais $\mathcal{S}(s_i) = \mathbf{x}_i$ de acordo com a seguinte regra de decisão:

$$\mathbf{x}_i \text{ é associado a } \omega_j \Leftrightarrow j = \arg \max_{k=1, \dots, c} \{P(\mathbf{x}_i|\omega_k)\}. \quad (2.1)$$

Na implementação de M.I.C.E., $P(\mathbf{x}_i|\omega_k)$ é modelada segundo as distribuições Gaussiana Univariada ou Multivariada quando \mathcal{S} possui, nesta ordem, uma ou mais de uma banda.

Parâmetros
Este método não possui parâmetros a serem ajustados.

2.1.2 KNN – K-Nearest Neighbors

Segundo o método K-Vizinhos Mais Próximos (*K-Nearest Neighbors* – KNN), a classificação de cada pixels s_i de \mathcal{S} é determinada a partir das k maiores similaridades entre o vetor de atributos deste pixel, expresso por \mathbf{x}_i , e os padrões rotulados existentes em \mathcal{D} . A similaridade é representada pelo inverso da distância euclidiana. Uma formalização para o método KNN é [4]:

$$(\mathbf{x}_i, \omega_j) \Leftrightarrow j = \arg \min_{j=1, \dots, c} \left\{ e^{-\#\{(\bar{\mathbf{x}}_i, \omega_j) \in \mathcal{N}_k(\mathbf{x}_i)\}} \right\}, \quad (2.2)$$

onde $\mathcal{N}_k(\mathbf{x}_i)$ é o conjunto dos elementos em que são mais próximos de \mathbf{x}_i . Tal conjunto pode ser expresso por:

$$\mathcal{N}_k(\mathbf{x}_i) = \{(\bar{\mathbf{x}}_p, \omega_q) : 0 \leq E(\mathbf{x}_i, \bar{\mathbf{x}}_1) \leq \dots \leq E(\mathbf{x}_i, \bar{\mathbf{x}}_k)\}, \quad (2.3)$$

sendo $E(\cdot, \cdot)$ a distância euclidiana e $\bar{\mathbf{x}}_p$ uma nova indexação dos padrões de \mathcal{D} em função da proximidade à \mathbf{x}_i .

Parâmetros
k : Número de vizinhos no espaço de atributos utilizados para determinação de \mathcal{N}_k , conforme definido em (2.3).

2.1.3 SVM – Support Vector Machine

Máquina de Vetores Suporte (*Support Vector Machine* – SVM) tem como princípio a classificação de dados, representados neste contexto pelos vetores \mathbf{x}_i associados aos pixels s_i de \mathcal{S} , a partir de hiperplanos de separação cuja margem de separação entre exemplos de treinamento em \mathcal{D} seja máxima. Os hiperplanos de separação são representados pelo lugar geométrico que anula (2.4), cujos parâmetros \mathbf{w} e b são obtidos via otimização de (2.5):

$$f(\mathbf{x}) = K(\mathbf{w}, \mathbf{x}) + b, \quad (2.4)$$

$$\begin{aligned} \min_{\lambda} \quad & \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to: } & \begin{cases} 0 \leq \lambda_i \leq C, i = 1, \dots, m \\ \sum_{i=1}^m \lambda_i y_i = 0. \end{cases} \end{aligned} \quad (2.5)$$

onde λ são multiplicadores de Lagrange, limitados superiormente pelo parâmetro C , $y_i \in \{-1, +1\}$ são indicadores de classe e $K(\cdot, \cdot)$ é uma função kernel. Sendo SVM um “classificador binário”, sua aplicação em problemas envolvendo mais de duas classes torna necessário o emprego de estratégias multiclases. Uma descrição abrangente sobre este método, respectivos parâmetros, exemplos de funções kernel e estratégias multiclases, é apresentada em [5]. Na implementação de M.I.C.E, o problema (2.5) é otimizado com uso da biblioteca LibSVM [6].

Parâmetros
<p>C: Penalidade definida nas restrições do problema de otimização (2.5);</p> <p>Função kernel: Adotadas para remapeamento e separação dos vetores de atributo dos pixels em espaços mais convenientes. Dentre diferentes kernels, estão disponíveis:</p> <p>Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$;</p> <p>Polinomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^q$, $q \in \mathbb{N}^*$;</p> <p>RBF: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \ \mathbf{x}_i - \mathbf{x}_j\ ^2}$, $\gamma \in \mathbb{R}^*$;</p> <p>Sigmóide: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \beta)$, $\gamma, \beta \in \mathbb{R}^*$.</p> <p>Estratégias multiclasse: Procedimentos para decompor problemas multiclasse em subproblemas binários e recombinar os respectivos resultados. Duas estratégias oferecidas são:</p> <p>Um-Contra-Todos (One-Against-All) Decompõe um problema com c classes em c subproblemas binários, de forma que cada subproblema será responsável pela separação de uma dada classe com relação as demais;</p> <p>Um-Contra-Um (One-Against-One) Decompõe um problema com c classes em $\binom{c}{2}$ subproblemas binários, de modo que cada subproblema realiza separação entre um par de classe específico.</p>

2.1.4 Minimum Distance Classifiers

De modo geral, a regra de decisão que caracterizam os métodos de classificação orientados pela mínima distância, e equivalentemente na maior similaridade, é expressa por:

$$(\mathbf{x}_i, \omega_j) \Leftrightarrow j = \arg \min_{j=1, \dots, c} M(\mathbf{x}_i, \omega_j), \quad (2.6)$$

onde $M(\cdot, \cdot)$ é uma dada distância que quantifica a similaridade entre \mathbf{x}_i e a classe ω_j . A escolha de $M(\cdot, \cdot)$ definem os métodos apresentados a seguir.

MEDC – Minimum Euclidean Distance Classifier

O Classificador de Mínima Distância Euclidiana (*Minimum Euclidean Distance Classifier* – MEDC) é definido ao considerar a distância euclidiana como quantificador de similaridade na Equação (2.6), dada por:

$$M_e(\mathbf{x}_i, \mathbf{d}_j) = \|\mathbf{x}_i - \mathbf{d}_j\|, \quad (2.7)$$

onde \mathbf{x}_i é um padrão não rotulado e \mathbf{d}_j é o vetor médio computado a partir dos elementos do subconjunto \mathcal{D}_j , o qual é composto por sua vez pelos elementos de \mathcal{D} que estão associados à classe ω_j .

MMDC – Minimum Mahalanobis Measure Classifier

De modo similar ao MEDC, o Classificador de Mínima Medida de Mahalanobis (*Minimum Mahalanobis Measure Classifier* – MMDC) é definido ao considerar a medida de

Mahalanobis para como quantificador de similaridade, cuja expressão é [7]:

$$M_m(\mathbf{x}_i, \mathcal{D}_j) = \left((\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j) \right)^{\frac{1}{2}}, \quad (2.8)$$

onde \mathbf{x}_i é um padrão não rotulado, μ_j e Σ_j são o vetor médio e matriz de covariância computados a partir dos elementos de \mathcal{D}_j .

Parâmetros
Estes métodos não possuem parâmetros a serem ajustados.

2.2 Métodos não supervisionados

2.2.1 K-Means

O algoritmo K-Médias (*K-Means*) consiste em particionar um conjunto de dados em K agrupamentos. Para isso, são realizados iterativamente dois passos básicos: associação e atualização [8].

No primeiro passo, cada elemento do conjunto de dados é associado a um dos K agrupamentos mais próximo, representado por um centróide. Os centroides são elementos definidos inicialmente como representantes para os agrupamentos, já a relação de proximidade é baseada em uma medida de distância adotada. Em seguida, no segundo passo, os centroides são atualizados segundo a média dos elementos previamente associados a eles. Este algoritmo termina quando ocorre convergência na atualização dos centroides.

Os agrupamentos finais determinam as classes, não rotuladas, que cada elemento do conjunto de dados pertence. A Figura 2.1 exibe um fluxograma do algoritmo K-Médias, segundo as etapas descritas anteriormente.

Formalmente, K-Médias efetua o particionamento do conjunto de dados de modo que a soma do quadrado dos desvios internos de cada grupo seja minimizado, diminuindo assim a variância intra-agrupamento total V , representada por [5]:

$$V = \sum_{j=1}^K \sum_{x_i \in \mathbf{c}_j} d(x_i, \mu_j) \quad (2.9)$$

onde μ_j é o centróide do agrupamento \mathbf{c}_j , com $j = 1, \dots, K$, representado pelo vetor média de tal agrupamento; x_i são vetores de atributos que representam cada posição pertencente a \mathbf{c}_j e $d(x_i, \mu_j)$ é uma medida de similaridade, neste caso, a distância euclidiana.

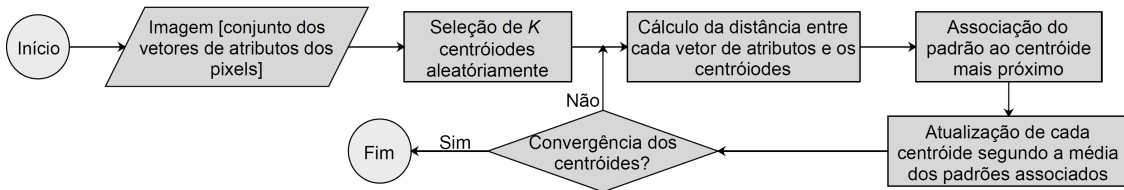


Figura 2.1: Fluxograma do algoritmo K-Médias.

Parâmetros
K: Número de agrupamentos adotados para o particionamento da imagem;
Epsilon: Limiar adotado para verificar convergência entre duas atualizações consecutivas nos centróides;
Max. Iter: Número máximo de iterações permitido, no caso de divergência no processo iterativo.

2.2.2 Expectation Maximization

Maximização da Esperança (*Expectation-Maximization* – EM) consiste em um processo iterativo, baseado no conceito de máxima verossimilhança, para determinar modelos de distribuição em um conjunto de dados. O processo iterativo é composto pela etapa de Esperança/Estimação (E), que será responsável por determinar uma função log-verossimilhança a partir de parâmetros calculados, e a etapa de Maximização (M), que irá calcular novos parâmetros a partir da maximização da função log-verossimilhança definida na etapa anterior. Sucessivamente, os parâmetros obtidos em M são reavaliados em E. Para maiores detalhes sobre este método, é sugerida a referência [9].

Em termos formais, o método EM realiza a classificação de \mathcal{S} através do agrupamento dos vetores de atributos \mathbf{x}_i em G grupos distintos, denotados por G_j , $j = 1, \dots, c$. A construção dos agrupamentos é realizada segundo a regra:

$$\mathbf{x}_i \text{ é associado a } G_j \Leftrightarrow j = \arg \max_{k=1, \dots, c} \{P(G_k | \mathbf{x}_i; \Theta_k)\}, \quad (2.10)$$

sendo Θ_k o conjunto de parâmetros estimados para a distribuição $P(G_k | \cdot)$ na etapa E.

Parâmetros
k: Número de agrupamentos adotados para o particionamento da imagem;
Epsilon: Limiar adotado para verificar convergência entre duas atualizações consecutivas nos centróides;
Max. Iter: Número máximo de iterações permitido, no caso de divergência no processo iterativo.

2.2.3 Fuzzy C-Means

Baseado em conceitos da lógica e conjuntos nebulosos, o método C-Medias Nebuloso (*Fuzzy C-Means* – FCM) é uma alternativa de classificação não supervisionada. O agrupamento de dados cuja separação não é bem definida torna-se uma das principais motivações quanto o uso do método FCM. As formalizações apontadas abaixo são baseadas no desenvolvimento apresentado em [10].

Seja \mathbf{x}_i , com $i = 1, \dots, m$ vetores de atributos dos pixels de uma dada imagem \mathcal{S} , e G_j , com $j = 1, \dots, c$, agrupamentos que particionam tal imagem, é definida a matriz W cujos elementos $w_{ij} \in [0, 1]$ representam a pertinência de \mathbf{x}_i ao agrupamento G_j . Ainda, com relação aos elementos de W , é assegurado que:

$$\sum_{k=1}^c w_{ik} = 1, \quad (2.11)$$

$$0 < \sum_{k=1}^m w_{kj} < m. \quad (2.12)$$

O processo iterativo que compreende o método FCM é similar ao diagrama apresentado na Figura 2.1. Diante as restrições sobre os elementos de W , a minimização da variância intra-grupal é alcançada iterativamente com atualizações dos valores w_{ij} , segundo a expressão:

$$w_{ij} = \frac{d(\mathbf{x}_i, \mu_j)^{\frac{1}{1-p}}}{\sum_{k=1}^c d(\mathbf{x}_i, \mu_k)^{\frac{1}{1-p}}} \quad (2.13)$$

onde $p > 1$ é um parâmetro (fator de nebulosidade – *fuzzifier factor*) que controla os pesos das associações entre dados/vetores e agrupamentos. $d(\cdot, \cdot)$ equivale à distância euclidiana. FCM torna-se equivalente ao KM ao passo que $p \rightarrow 1$.

A atualização dos centróides μ_j é obtida pela média ponderada dos vetores associados a cada agrupamento, ou seja:

$$\mu_j = \frac{\sum_{i=1}^m w_{ij}^p \mathbf{x}_i}{\sum_{i=1}^m w_{ij}^p} \quad (2.14)$$

Parâmetros
k: Número de agrupamentos adotados para o particionamento da imagem;
Epsilon: Limiar adotado para verificar convergência entre duas atualizações consecutivas nos centróides;
Fuzzifier Factor: Parâmetro de decisão fuzzy.

3. Classificação Baseada em Regiões

As discussões conduzidas no início do Capítulo 2, respeito do conceito de classificação e paradigmas de aprendizado, são aplicáveis à classificação baseada em regiões. A principal distinção entre tais abordagens de classificação reside nos elementos considerados. Enquanto a primeira abordagem realiza a classificação de uma dada imagem \mathcal{I} a partir da aplicação de F sobre o vetor de atributo de cada pixel s , na classificação baseada em regiões, F é aplicada sobre regiões previamente identificadas em \mathcal{I} .

Em termos formais, seja \mathcal{I} uma imagem definida sobre um suporte $\mathcal{S} \subset \mathbb{N}^2$ e \mathcal{X} um espaço de atributos, $\mathcal{I}(s) = \mathbf{x}$ denota que um pixel $s \in \mathcal{S}$ de \mathcal{I} possui vetor de atributos $\mathbf{x} \in \mathcal{X}$. O processo de classificação baseada em regiões consiste em associar a classe ω_j , $j = 1, \dots, c$, à região $R_i \subset \mathcal{S}$, $i = 1, \dots, r$. Uma região R_i é um conjunto de pixels s_a , $a = 1, \dots, \#R_i$, onde os atributos de s_a são obtidos a partir de $\mathcal{I}(s_a)$ e $\#$ é o operador de cardinalidade. Neste contexto, o suporte de \mathcal{I} é particionado em r regiões disjuntas por um processo de segmentação. Regiões representam um conjunto de pixels espacialmente conectados cujos vetores de atributos atendem a um determinado critério de uniformidade. No processo de classificação, todos os pixels de uma mesma região são associados a uma mesma classe.

Para realização do aprendizado supervisionado, é empregado um conjunto de treinamento $\mathcal{D} = \{(R_i, \omega_j) \in \mathcal{S} \times \Omega : i = 1, \dots, m; j = 1, \dots, c\}$, onde m é o número de regiões de treinamento. A notação (R_i, ω_j) indica que R_i está associada à classe ω_j . De modo análogo, no aprendizado não supervisionado, os agrupamentos de elementos similares são compostos por regiões R_i .

3.1 Métodos supervisionados

3.1.1 MSDC – Minimum Stochastic Distance Classifier

Assim como formalizado na Seção 2.1.4, a regra de decisão que caracterizam os métodos de classificação orientados pela mínima distância é expressa por:

$$(R_i, \omega_j) \Leftrightarrow j = \arg \min_{j=1, \dots, c} M(f_{R_i}, f_{\omega_j}), \quad (3.1)$$

onde $M(\cdot, \cdot)$ é uma dada distância que quantifica a similaridade entre as funções de distribuição de probabilidade f_{R_i} e f_{ω_j} que modelam os comportamentos da região R_i e da classe ω_j , respectivamente.

A fim de efetuar comparações entre regiões e exemplos rotulados de determinada classe, atentando ao fato de ambos tratarem-se de conjuntos de vetores, a adoção de distâncias estocásticas torna-se conveniente. Segundo tais distâncias, a similaridade entre conjunto de informação é verificada diante a similaridade entre as distribuições de probabilidade dos respectivos conjuntos. Bathacharyya e Jeffries-Matusita são dois exemplos de distâncias estocásticas de uso comum em classificação, cujas expressões são [11]:

$$B(C, D) = \frac{1}{8} (\mu_C - \mu_D)^T \left(\frac{\Sigma_C + \Sigma_D}{2} \right)^{-1} (\mu_C - \mu_D) + \frac{1}{2} \ln \left(\frac{|\Sigma_C + \Sigma_D|}{\sqrt{|\Sigma_C| |\Sigma_D|}} \right), \quad (3.2)$$

onde μ e Σ são respectivamente o vetor médio e a matriz de covariância, ambos estimados a partir de um dado conjunto Z . Neste contexto, Z pode representar os vetores de atributos associados a uma região R_i quanto a uma classe ω_j . T , $|\cdot|$ e $(\cdot)^{-1}$ representam as operações de transposição, determinante e inversão matricial, respectivamente.

$$JM(C, D) = 2 \left(1 - e^{-B(C, D)} \right), \quad (3.3)$$

sendo $B(\cdot, \cdot)$ a distância de Bhattacharyya.

Diante da regra de decisão expressa em (3.1) e com adoção de uma distância estocástica, é definido o Classificador de Mínima Distância Estocástica (*Minimum Stochastic Distance Classifier* – MSDC).

Parâmetros
Stochastic distance: Permite a seleção dentre Bathacharyya e Jeffries-Matusita, considerando distribuições Gaussiana Multivariadas, como opções de distância estocástica.

3.1.2 MASDC – Minimum Average Stochastic Distance Classifier

O Classificador de Mínima Média Distância Estocástica (*Minimum Average Stochastic Distance Classifiers* – MASDC) possui regra de decisão similar ao MSDC, conforme apresentado pela Equação (3.1.1). A distinção entre tais métodos encontra-se no modo como a distância estocástica entre a região não rotulada e as classes é computada. Enquanto em MSDC $M(f_{R_i}, f_{\omega_j})$ corresponde a uma distância estocástica entre as distribuições de

probabilidade que modelam o comportamento de R_i e das regiões de treinamento em \mathcal{D} associadas à classe ω_j , no métodos MASDC é computada uma distância média entre R_i e as distribuições de probabilidade que modelam cada uma das regiões de treinamento em \mathcal{D} e com rótulo ω_j . De modo genérico, MASDC é definido com a substituição de $M_{med}(R_i, f_{\omega_j})$ em (3.1.1), cuja expressão é [4]:

$$M_{\text{mean}}(f_{R_i}, f_{\omega_j}) = \frac{1}{t_j} \sum_{l=1}^{t_j} M(f_{R_i}, f_{\omega_j R_l}), \quad (3.4)$$

onde $f_{\omega_j R_l}$ é a distribuição de probabilidade que modela a l -ésima região de treinamento associada a ω_j , a qual contém t_j regiões de treinamento em \mathcal{D} .

Parâmetros
Stochastic distance: Permite a seleção dentre Bathacharyya e Jeffries-Matusita, considerando distribuições Gaussiana Multivariadas, como opções de distância estocástica.

3.1.3 NNSDC – Stochastic Distance Nearest Neighbor Classifier

Ainda de modo similar ao MSDC, o Classificador de Distância Estocástica do Vizinho Mais Próximo (*Stochastic Distance Nearest Neighbor Classifier* – SDNNC) consiste na substituição de $M(\cdot, \cdot)$ em (3.1.1) por $M_{\min}(\cdot, \cdot)$, o qual retorna a menor distância entre R_i e uma das regiões de treinamento associada a ω_j . $M_{\min}(\cdot, \cdot)$ é definida por [4]:

$$M_{\min}(f_{R_i}, f_{\omega_j}) = \min \{M(f_{R_i}, f_{\omega_j R_l}) : l = 1, \dots, t_j\}. \quad (3.5)$$

Parâmetros
Stochastic distance: Permite a seleção dentre Bathacharyya e Jeffries-Matusita, considerando distribuições Gaussiana Multivariadas, como opções de distância estocástica.

3.1.4 KNN – K-Nearest Neighbors

Na classificação baseada em regiões, o método KNN é adaptado pela simples substituição de $E(\cdot, \cdot)$ em (2.3) por uma distância estocástica $M(\cdot, \cdot)$ qualquer e pela consideração de regiões (ou conjunto de vetores) ao invés de vetores únicos como elementos no processo de treinamento e classificação.

Parâmetros
k: Número de regiões utilizadas para determinação de \mathcal{N}_k , conforme definido em (2.3).
Stochastic distance: Permite a seleção dentre Bathacharyya e Jeffries-Matusita, considerando distribuições Gaussiana Multivariadas, como opções de distância estocástica.

3.1.5 SVM – Support Vector Machine

Para aplicação do método SVM na classificação baseada em regiões, além da consideração de regiões (ou conjunto de vetores) ao invés de vetores únicos como elementos

no processo de treinamento e classificação, é feita a adoção da função kernel Jeffries-Matusita[12]:

$$K_{JM}(R_i, R_j) = e^{-\gamma \overline{JM}(f_{R_i}, f_{R_j})}, \quad (3.6)$$

onde $\gamma \in \mathbb{R}_+$ é um parâmetro de regularização e \overline{JM} é a seguinte medida:

$$\overline{JM}(R_i, R_j) = \begin{cases} 0, & \text{se } R_i = R_j \\ JM(R_i, R_j) + 2, & \text{caso contrário} \end{cases} \quad (3.7)$$

Parâmetros
<p>C: Penalidade definida nas restrições do problema de otimização (2.5);</p> <p>γ: Parâmetro de regularização na função kernel Jeffries-Matusita;</p> <p>Estratégias multiclasse: Procedimentos para decompor problemas multiclases em subproblemas binários e recombina os respectivos resultados. Duas estratégias oferecidas são:</p> <p>Um-Contra-Todos (<i>One-Against-All</i>) Decompõe um problema com c classes em c subproblemas binários, de forma que cada subproblema será responsável pela separação de uma dada classe com relação as demais;</p> <p>Um-Contra-Um (<i>One-Against-One</i>) Decompõe um problema com c classes em $\binom{c}{2}$ subproblemas binários, de modo que cada subproblema realiza separação entre um par de classe específico.</p> <p>Epsilon: Tolerância adotada como critério de parada no processo de otimização inerente ao método.</p>

3.2 Método não supervisionado

3.2.1 K-Means

A versão baseada em regiões do algoritmo KM possui estrutura similar à versão baseada em pixel, conforme discutido na Seção 2.2.1. As principais diferenças residem no fato de adotar distância estocástica, ao invés da distância euclidiana, e no critério de convergência. Em especial, a convergência dos centróides é verificada segundo p -valor obtido por teste de hipótese derivado do conceito de distâncias estocásticas [13]. Para maiores detalhes sobre esta versão do algoritmo KM, é indicada a referência [8].

Parâmetros
<p>k: Número de agrupamentos adotados para o particionamento da imagem;</p> <p>Convergence p-value: Limiar de p-valor adotado para verificar convergência entre duas atualizações consecutivas nos centróides;</p> <p>Stochastic distance: Permite a seleção dentre Bathacharyya e Jeffries-Matusita, considerando distribuições Gaussiana Multivariadas, como opções de distância estocástica.</p>

4. Segmentação de Imagens

Segmentação de Imagem é um processo que consiste em dividir as imagens em segmentos internamente homogêneos e externamente heterogêneos baseados em alguma característica específica [14]. A segmentação pode ser vista como um processo que particiona uma dada imagem \mathcal{I} em sub-regiões R_i , com $i = 1, \dots, n$, de modo que:

- (i) $\bigcup_{i=1}^n R_i = \mathcal{I}$;
- (ii) R_i é um conjunto conectado, $i = 1, \dots, n$;
- (iii) $R_i \cap R_j = \emptyset$, $i, j = 1, \dots, n$, $i \neq j$;
- (iv) $Q(R_i) = \text{verdadeiro}$, $i = 1, \dots, n$;
- (v) $Q(R_i \cup R_j) = \text{falso}$, sendo R_i e R_j regiões adjacentes.

Nas propriedades (iv) e (v), $Q(R_k)$ representa uma propriedade lógica sobre os pixels de determinada região, sendo verdadeiro se todos os pixels de R_k são similares.

Conforme estruturado por Gonzalez e Woods [14], os métodos de segmentação de imagem baseiam-se em duas propriedades básicas dos pixels com relação a sua vizinhança local: descontinuidade e similaridade. Quando fundamentados nas propriedades de descontinuidade dos pixels, a segmentação é “orientada bordas”, já quando fundamentada nas características de similaridade, é dita “orientada por regiões”.

4.1 Segmentação orientada por bordas

A segmentação orientada por bordas usa o postulado que mudanças no nível de cinza dos pixels acontecem quando existe um limite entre duas regiões. A detecção de zonas de fronteira entre regiões é usualmente realizada por operações de diferenciação de primeira ordem, por exemplo, através dos filtros de gradiente. A vantagem essencial da abordagem orientada por bordas refere-se a sua arquitetura e facilidade de implementação.

4.1.1 Homogeneic Ramification

Remificação Homogênea (*Homogeneic Ramification* – HR) é um método de segmentação que consiste em identificar bordas das regiões da imagem, a partir de um processo de diferenciação, e em seguida determinar a expansão (ramificação) das regiões a partir dos locais de maior homogeneidade e menor gradiente. A formalização deste método é apresentada em [15].

Parâmetros
Threshold: Limiar (superior) para identificar a similaridade entre pixels; Min. Area: Área mínima para distinguir regiões pequenas e proceder com união entre regiões vizinhas.

4.2 Segmentação orientada por regiões

A abordagem orientada por regiões tenta isolar áreas da imagem que são homogêneas segundo determinadas características. Para isso, os pixels são submetidos a um processo de agrupamento. Durante o processo de segmentação, as áreas candidatas podem aumentar de tamanho, serem comprimidas, divididas, criadas ou eliminadas. Dentre diferentes métodos de segmentação baseado em regiões, o Crescimento de Regiões (*Region Growing*), introduzido em [16], é um dos mais populares.

4.2.1 Region Growing

O Crescimento de Regiões (*Region Growing* – RG) é baseado no agrupamento de pixels, ou sub-regiões, em regiões maiores [14]. O Crescimento de Regiões se inicia com a escolha de um conjunto de pixels chamados sementes e a partir destes, outros pixels adjacentes são incluídos na região segundo um grau de semelhança, como intensidade e textura similares. O processo ocorre iterativamente até que todos os pixels sejam avaliados e as regiões possam ser delimitadas por polígonos fechados, onde os pixels dentro de cada polígono devem possuir certa similaridade.

Parâmetros
Threshold: Limiar (superior) para identificar a similaridade entre pixels; Min. Area: Área mínima para distinguir regiões pequenas e proceder com união entre regiões vizinhas.

5. Simulação de Imagens

O uso de imagens simuladas permite a condução de experimentos controlados, isto é, experimentos cujo comportamento dos alvos é conhecido *a priori*, além de permitir a realização de avaliações sem a influência de uma imagem particular. Os métodos de simulação implementados no software M.I.C.E. são ditos “supervisionados” uma vez que a simulação é baseada no comportamento de objetos observados em uma imagem base. Para uma dada imagem \mathcal{I} , o comportamento dos objetos observados é representado pelo conjunto $\mathcal{D} = \{(\mathbf{x}_i, \omega_j) : i = 1, \dots, m; j = 1, \dots, c\}$, tal que \mathbf{x}_i é um pixel de \mathcal{I} observado sobre um objeto da classe ω_j .

Dois processos de simulação, com relação a determinação dos objetos que compõem as imagens sintéticas, são oferecidos no M.I.C.E.: simulação orientada à *phantom* e simulação por objetos aleatórios. Enquanto o primeiro processo emprega formas fixas quanto a determinação dos objetos na imagem, o segundo produz objetos cuja distribuição espacial é alterada em cada simulação. Além da distribuição espacial dos alvos, a sintetização do comportamento dos pixels pode ser conduzida nesta ferramenta por dois processos estatísticos distintos: amostragem aleatória (*bootstrapping*) e distribuição gaussiana. As seções seguintes formalizam cada um dos processos mencionados.

5.1 Simulação Orientada à Phantom

Neste modelo, as simulações são realizadas sobre um modelo idealizado de imagem, denominado *Phantom* (ou *phantom* de nível 1), denotado por \mathcal{A} , o qual é composto por a_j regiões, com $j = 1, 2, \dots, c$ tal que $\bigcup_{j=1}^c a_j = \mathcal{A}$. As regiões de \mathcal{A} simulam alvos distintos, segundo as informações de \mathcal{D} . Por sua vez, as regiões $a_j \subset \mathcal{A}$ podem estar divididas em b_{jk} segmentos, com $k = 1, 2, \dots, z$, tal que $\bigcup_{k=1}^z b_{jk} = a_j$. Assim, \mathcal{B} , denominada por *Segmentação da Phantom* (ou *phantom* de nível 2), é uma partição de \mathcal{A} em $c \times z$ subconjuntos.

De acordo com a notação introduzida, denota-se por $\tilde{\mathbf{p}}_{jk}$ os pixels que compõe o segmento b_{jk} da imagem simulada. Quando $a_j = b_{jk}$ as regiões não estão divididas em segmentos, e nesse caso, $\tilde{\mathbf{p}}_j$ representa os pixels que compõe a região a_j da imagem simulada. Os processos para simulação de $\tilde{\mathbf{p}}_{jk}$ são descritos nas subseções seguintes.

5.1.1 Simulação Bootstrap

A simulação *bootstrap* consiste basicamente em um processo de reamostragem aleatória e permite a estimação da maioria das distribuições estatísticas. O procedimento bootstrap é recomendado em diferentes situações, por exemplo, quando a distribuição dos dados é complicada ou desconhecida.

Para simulação de imagens segundo o procedimento *bootstrap*, é adotado inicialmente um conjunto de pixels rotulados $\mathcal{D} = \{(\mathbf{x}_i, \omega_j) : i = 1, \dots, m; j = 1, \dots, c\}$ observados de \mathcal{I} . Assim, os pixels com comportamento da forma $\tilde{\mathbf{p}}_j$ e que compõem a_j são determinados a partir da seleção aleatória de $(\mathbf{x}_i, \omega_j) \in \mathcal{D}$, fazendo $\tilde{\mathbf{p}}_j = \mathbf{x}_i$.

Parâmetros

Este método não possui parâmetros a serem ajustados.

5.1.2 Simulação Gaussiana

A simulação de objetos com comportamento expresso por distribuição Gaussiana faz uso de uma *Phantom* \mathcal{A} e de um conjunto de pixels \mathcal{D} observados em alvos de \mathcal{I} , e em adicional, é possível considerar uma segmentação sobre as regiões da Imagem da Phantom. A simulação de pixels distribuídos de forma Gaussiana é dada pela seguinte expressão [17]:

$$\tilde{\mathbf{p}}_{jk} = [(E_j \times L_j \times \mathbf{v}) \times \zeta_k] + (\mu_j \times \psi_k) \quad (5.1)$$

onde μ_j e Σ_j referem-se ao vetor média e a matriz de covariância estimadas a partir de todos os vetores \mathbf{x}_i associados à classe ω_j presentes em \mathcal{D} , com $j = 1, \dots, c$. A matriz R_j^{-1} é o inverso da matriz de autovetores de Σ_j , L_j é a matriz diagonal composta pela raiz quadrada dos autovalores de Σ_j , \mathbf{v} é um vetor aleatório de mesma dimensão de μ_j , gerado por uma distribuição Gaussiana Multivariada padrão. Os elementos ζ_k e ψ_k são escalares distribuídos uniformemente, em intervalos definido pelo usuário, e são responsáveis por modelar variações existente entre segmentos de uma mesma região.

Cabe ressaltar que as colunas da matriz R_j estão organizadas de acordo com a ordem crescente dos respectivos autovalores, assim como os elementos da diagonal de L_j , também dispostos em ordem crescente. Ainda, a matriz R_j é responsável por descrever o comportamento da estrutura de correlação entre bandas para um dado alvo, já a informação sobre a variância em cada banda é dada por L_j .

Parâmetros

Lower/higher psi: Determina os limitantes inferior e superior do intervalo sobre o qual são selecionados aleatoriamente os valores de ψ_k considerados na expressão 5.1;

Lower/higher zeta: Determina os limitantes inferior e superior do intervalo sobre o qual são selecionados aleatoriamente os valores de ζ_k considerados na expressão 5.1;

5.2 Simulação de Objetos Aleatórios

Diferente da simulação orientada à *phantom*, este processo de simulação consiste em determinar objetos aleatórios a partir do particionamento de um suporte, de dimensões fixadas, em regiões contínuas com formas irregulares, respeitando ainda restrições bases relacionadas à área mínima (em pixels) e quantidade máxima de objetos/regiões. Uma vez realizado tal particionamento, os pixels das regiões definidas são determinados utilizando os procedimentos de simulações Bootstrap ou Gaussiana segundo o comportamento de alvos conhecidos *a priori* e selecionados aleatoriamente de \mathcal{D} . Este processo de simulação foi introduzido e é descrito em maiores detalhes em [15].

5.2.1 Simulação Bootstrap

A simulação do comportamento dos objetos para este caso é equivalente ao descrito na Subseção 5.1.1.

Parâmetros
Columns: Número de colunas que define o suporte da imagem simulada;
Lines: Número de linhas que define o suporte da imagem simulada;
Min. object area (pixels): Quantidade mínima de pixels que os objetos simulados devem apresentar;
Max. of objects: Quantidade máxima de objetos que deve compor a imagem simulada.

5.2.2 Simulação Gaussiana

A simulação do comportamento dos objetos para este caso é equivalente ao descrito na Subseção 5.1.2.

Parâmetros
Columns: Número de colunas que define o suporte da imagem simulada;
Lines: Número de linhas que define o suporte da imagem simulada;
Min. object area (pixels): Quantidade mínima de pixels que os objetos simulados devem apresentar;
Max. of objects: Quantidade máxima de objetos que deve compor a imagem simulada.
Lower/higher psi: Determina os limitantes inferior e superior do intervalo sobre o qual são selecionados aleatoriamente os valores de ψ_k considerados na expressão 5.1;
Lower/higher zeta: Determina os limitantes inferior e superior do intervalo sobre o qual são selecionados aleatoriamente os valores de ζ_k considerados na expressão 5.1;

5.3 Simulação para Detecção de Mudanças

Esta ferramenta oferece uma alternativa para geração de conjuntos de dados simulados destinados à avaliação e/ou comparação de métodos de detecção de mudança. De modo geral, o processo simula um par de modelos conceituais (i.e., imagens *Phantom*), sendo

um destes modelos obtido através de alterações geométricas sobre o outro modelo. A simulação espectral é efetuada segundo a Equação 5.1. Além disso, com intuito de simular distorções espectrais entre instantes, é opcional a aplicação de uma rotação espectral sobre o espaço de atributos original. Para maiores detalhes, é sugerida a leitura do artigo **NegriFrery2020**.

Parâmetros
<p>Columns: Número de colunas que define o suporte da imagem simulada;</p> <p>Lines: Número de linhas que define o suporte da imagem simulada;</p> <p>Amostragem aleatória: Porcentagem de pixels considerados para extração e geração de conjuntos de treinamento como resultado adicional do processo de simulação;</p> <p>Faixa de densidade: Intervalo de valores de densidade de cobertura dos objetos simulados sobre o suporte (colunas/linhas) estabelecido;</p> <p>Faixa de escala: Intervalo de valores de escala dos objetos simulados em relação ao suporte estabelecido;</p> <p>Faixa de expansão: Intervalo de valores (ou razão) de expansão dos objetos simulados no segundo instante em relação ao primeiro;</p> <p>Flutuação sobre a média: Intervalo de multiplicadores considerados na expressão 5.1;</p> <p>Flutuação sobre as covariâncias: Intervalo de multiplicadores considerados na expressão 5.1;</p> <p>Distorção espectral: Efetua uma rotação aleatória sobre os eixos do espaço de atributos dos dados;</p>

6. Detecção Não Supervisionada

A detecção de mudanças consiste em identificar alterações ocorridas entre duas ou mais imagens adquiridas em instantes diferentes. Sejam $\mathcal{I}^{(1)}$ e $\mathcal{I}^{(2)}$ duas imagens definidas sobre um suporte $\mathcal{S} \subset \mathbb{N}^2$ e referentes a uma mesma localização, porém tomadas em instantes distintos. Denota-se por $\mathbf{x}_i^{(j)} = \mathcal{I}^{(j)}(s_i)$ o vetor de atributos associado à posição $s_i \in \mathcal{S}$, sendo $\mathbf{x}_i^{(j)}$ um elemento do espaço de atributos \mathcal{X} . Os valores associados às componentes de $\mathbf{x}_i^{(j)}$ representam as respectivas medidas associadas.

As técnicas de detecção de mudança são fortemente suportadas por conceitos derivados das áreas de Reconhecimento de Padrões e Processamento Digital de Imagens. De acordo com [18], as técnicas de detecção de mudança podem ser agrupados dentre as seguintes categorias: (i) álgebra entre imagens; (ii) transformação de imagens; (iii) extração de atributos; (iv) classificação de imagens; (v) outros métodos. Porém, autores como [19], distinguem os métodos de detecção de mudança de uma forma menos específica dentre os tipos supervisionado e não supervisionado. Enquanto os métodos supervisionados necessitam de dados para o treinamento de um classificador, a abordagem não supervisionada é guiada de forma automática com base em processos como limiarização e agrupamento de dados. Esta característica de independência de conhecimento *a priori* torna os métodos não supervisionados mais atraentes em comparação aos supervisionados.

Nas seções que seguem contemplam diferentes métodos não supervisionados de detecção de mudanças disponíveis no software M. I. C. E. .

6.1 CVA – Change Vector Analysis

Análise de Vetor de Mudanças (*Change Vector Analysis* – CVA), consiste em uma abordagem clássica de detecção não supervisionada de mudanças. O conceito principal deste método envolve a obtenção de uma imagem de amplitude de diferenças $\mathcal{I}_a = \|\mathcal{I}^{(1)} - \mathcal{I}^{(2)}\|$, seguida pela distinção dos respectivos pixels segundo sua tendência em expressar ou não

a ocorrência de mudança.

Tal distinção é alcançada com uso de técnicas de limiarização, como por exemplo, os algoritmos de Otsu [20] ou Kittler-Illingworth [21].

Parâmetros
Método de limiarização: Permite a seleção dentre os métodos de Otsu e Kittler-Illingworth para limiarização da imagem de diferenças; Regra de discretização: Possibilita a escolha dentre as regras de Scott e Freedman-Diaconis para discretização do histograma gerado a partir da imagem de amplitude de diferenças.

6.2 PCA-KM

Em [22] é proposto um método que une Análise de Componentes Principais (PCA) e o algoritmo K-Médias, originando assim o método PCA-KM. De forma superficial, este método é iniciado com o particionamento de uma imagem de amplitude de diferenças em blocos de dimensão $h \times h$ não sobrepostos, sobre os quais são construídos vetores cujas componentes são os valores internos a tais blocos. Os vetores extraídos são empregados na obtenção de eixos ortogonais, com uso de PCA, que define uma transformação linear sobre os vetores originais. Posteriormente, novas representações vetoriais são extraídas para cada pixel da imagem de amplitude de diferenças, a partir de blocos centrados em tais pixels, e então projetados para o espaço determinado pela transformação linear revelada pela técnica PCA, considerando apenas as s primeiras componentes desta projeção. Por último, os dados projetados são submetidos ao algoritmo K-Médias a fim de detectar dois agrupamentos, de modo que o agrupamento mais próximo à origem do espaço projetado determina os pixels associados a não ocorrência de mudança, ao contrário do outro agrupamento, composto por pixels onde ocorrem mudanças temporais.

Parâmetros
h : Dimensão dos blocos para geração de uma representação local das diferenças de amplitude; s : Quantidade de componentes principais consideradas.

6.3 GKC+SA

Com motivação nos valores de amplitude de diferenças, o método de agrupamento de Gustafson-Kessel Clustering (*Gustafson-Kessel Clustering* – GKC) [23] efetua a análise de duas fontes de informação: a amplitude de diferenças em si; e a informação contextual das amplitudes de diferença computada através de uma média móvel espacial. Assim como PCA-KM, o objetivo final é a identificação de dois agrupamentos, relacionados às regiões de mudança e não mudança.

O algoritmo GKC é parametrizado com uso do algoritmo *Simulated Annealing* (SA) [24] e da medida de Xie-Beni [25] como função objetivo do problema de otimização.

Parâmetros
ε : Usado como referencial de convergência pelo método GKC; α : Parâmetro “fator de resfriamento” adotado pelo algoritmo SA.

6.4 USFA – Unsupervised Slow Feature Analysis

[18] propõe o uso da técnica *Slow Feature Analysis* (SFA) no tratamento de variações relevantes e irrelevantes observadas sobre um par de imagens multitemporais. Nestas condições, os dados de entrada são transformados e alocados sobre um novo espaço de atributos cuja identificação de mudanças e não mudanças pode ocorrer de forma simplificada. O uso de técnicas de limiarização ou mesmo de agrupamento podem ser usadas neste propósito.

Parâmetros
Método de limiarização: Permite a seleção dentre os métodos de Otsu e Kittler-Illingworth para limiarização da imagem de diferenças; Regra de discretização: Possibilita a escolha dentre as regras de Scott e Freedman-Diaconis para discretização do histograma gerado a partir da imagem de amplitude de diferenças.

6.5 ASF – Alternating Sequential Filtering

Em [26], com atenção voltada à detecção de mudança a partir de imagens de radar, é proposto um método baseado no conceito de Filtros Sequenciais Alternantes (*Alternating Sequential Filters* – ASF).

Analogamente aos métodos CVA e PCA-KM, inicialmente é determinada a imagem de amplitude de diferenças. Posteriormente, são aplicados ASF a fim de simplificar as diferenças expressas na imagem de diferenças e ao mesmo tempo conservar a informação geométrica dos objetos existentes. O conceito de ASF corresponde a uma sequência de operações morfológicas de abertura e fechamento [14]. Na proposta de Mura (2008), tais operações são aplicadas de modo iterativo e com aumento gradativo nas dimensões do elemento estruturante. O número de iterações realizadas corresponde a um parâmetro neste método. Por último, é aplicada uma limiarização pelo método de Kittler-Illingworth sobre a filtragem alcançada. Ao passo que pixels associados a valores abaixo do limiar estabelecido são caracterizados como não mudança, e valores acima deste limiar corresponde a ocorrência de mudança.

Parâmetros
Iterações: Quantidade de iterações do tipo abertura-fechamento aplicadas sobre a imagem de amplitude de diferenças.

6.6 FR – Floating References

Proposto por NegriFrery2020, o método de Referências Flutuantes (*Floating References* – FR) efetua a distinção entre áreas de mudança e não mudança segundo o desvio de comportamento observado sobre objetos similares.

Inicialmente, para fins de regularização, é considerada sobre instante inicial (i.e., $\mathcal{J}^{(1)}$) uma suavização espacial baseada no valor médio dos atributos internos a um dado raio de vizinhança. Posteriormente, é conduzido um processo de agrupamento hierárquico divisivo com objetivo de particionar o instante inicial em subconjuntos com tamanho mínimo predefinido. Por último, sobre cada um dos agrupamentos identificados, são esta-

belecidos referenciais de tendência sobre as amplitudes dos desvios entre $\mathcal{J}^{(1)}$ e $\mathcal{J}^{(2)}$. Valores extremos em relação a estes referenciais são caracterizados como mudanças relevantes.

Parâmetros
Raio de vizinhança: Alcance considerado no processo de suavização espacial; Tamanho mínimo dos agrupamentos: Quantidade, em porcentagem, referente ao tamanho mínimo permitido aos agrupamentos que caracterizam o instante inicial.

6.7 HBSC – Homogeneous Block Single-class Classification

O método HBSC, acrônimo de *Homogeneous Block Single-class Classification* [27], emprega o conceito de detecção de anomalias via Single-Class SVM (SC-SVM) [28] para caracterizar regiões de mudança a partir de um par de imagens multitemporais. O treinamento do método SC-SVM é efetuado via áreas de não mudança, identificadas inicialmente através de blocos homogêneos segundo testes de hipótese derivados de distância estocástica.

Parâmetros
α : Significância adotada para caracterização dos blocos homogêneos; ν : Parâmetros associado ao método SC-SVM; γ : Parâmetro associado a função kernel RBF, considerada no método SC-SVM.

Bibliografia

Livros

- [5] A. R. Webb. *Statistical Pattern Recognition*. 2nd. Chichester: John Wiley & Sons, 2002, p. 514.
- [7] S. Theodoridis e K. Koutroumbas. *Pattern Recognition*. 4th. San Diego: Academic Press, 2008, p. 984. ISBN: 9781597492720.
- [10] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981. ISBN: 0306406713.
- [11] J. A. Richards e J. Xiuping. *Remote Sensing Digital Image Analysis: An Introduction*. Springer, dez. de 2005. ISBN: 3540251286.
- [14] R. C. Gonzalez e R. E. Woods. *Digital Image Processing*. 3rd. Prentice Hall, 2007. ISBN: 013168728X.

Artigos

- [4] R. G. Negri, L. V. Dutra, S. J. S. Sant'Anna et al. "Examining region-based methods for land cover classification using stochastic distances". Em: *International Journal of Remote Sensing* 37.8 (2016), pp. 1902–1921. DOI: 10.1080/01431161.2016.1165883.
- [6] C. Chang e C. Lin. "LIBSVM: A library for support vector machines". Em: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011), pp. 1–27.
- [8] R. G. Negri, W. B. da Silva e T. S. G. Mendes. "K-means algorithm based on stochastic distances for polarimetric synthetic aperture radar image classification". Em: *Journal of Applied Remote Sensing* 10.4 (2016), pp. 045005–045005.

- [12] R. G. Negri, W. C. O. Casaca e E. A. Silva. “Region-based Classification of Pol-SAR Data Through Kernel Methods and Stochastic Distances”. Em: *Lecture Notes in Computer Science* in press (2017).
- [13] A.C. Frery, A.D.C. Nascimento e R.J. Cintra. “Analytic Expressions for Stochastic Distances Between Relaxed Complex Wishart Distributions”. Em: *Geoscience and Remote Sensing, IEEE Transactions on* 52.2 (fev. de 2014), pp. 1213–1226. ISSN: 0196-2892. DOI: 10.1109/TGRS.2013.2248737.
- [15] R. G. Negri e M. F. S. Silva. “Um novo método de segmentação de imagem com abordagem baseada em bordas e regiões”. Em: *Revista Brasileira de Cartografia* 3.65 (2013), pp. 441–453.
- [16] S. W. Zucker. “Region growing: Childhood and Adolescence”. Em: *Computer Graphics and Image Processing* 5.3 (set. de 1976), pp. 382–399.
- [17] R. G. Negri, S. J. S. Sant’Anna e L. V. Dutra. “Aplicação de Modelos de Aprendizado Semissupervisionado na Classificação de Imagens de Sensoriamento Remoto”. Em: *Revista de Informática Teórica e Aplicada* 20.2 (2013), pp. 32–55. ISSN: 2175-2745.
- [18] Chen Wu, Bo Du e Liangpei Zhang. “Slow Feature Analysis for Change Detection in Multispectral Imagery”. Em: *IEEE Transactions on Geoscience and Remote Sensing* 52.5 (2014), pp. 2858–2874. DOI: 10.1109/TGRS.2013.2266673.
- [19] M. Volpi, D. Tuia, G. Camps-Valls et al. “Unsupervised Change Detection With Kernels”. Em: *IEEE Geoscience and Remote Sensing Letters* 9.6 (2012), pp. 1026–1030. ISSN: 1545-598X. DOI: 10.1109/LGRS.2012.2189092.
- [20] N. Otsu. “A Threshold Selection Method from Gray-Level Histograms”. Em: *IEEE Trans. on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66. DOI: 10.1109/TSMC.1979.4310076.
- [21] J. Kittler e J. Illingworth. “Minimum error thresholding”. Em: *Pattern Recognition* 19.1 (1986), pp. 41–47. ISSN: 0031-3203. DOI: [https://doi.org/10.1016/0031-3203\(86\)90030-0](https://doi.org/10.1016/0031-3203(86)90030-0).
- [22] T. Celik. “Unsupervised Change Detection in Satellite Images Using Principal Component Analysis and k -Means Clustering”. Em: *IEEE Geoscience and Remote Sensing Letters* 6.4 (out. de 2009), pp. 772–776. ISSN: 1545-598X. DOI: 10.1109/LGRS.2009.2025059.
- [24] S. Kirkpatrick, C. D. Gelatt e M. P. Vecchi. “Optimization by Simulated Annealing”. Em: *Science* 220.4598 (1983), pp. 671–680. ISSN: 0036-8075. DOI: 10.1126/science.220.4598.671. URL: <http://science.sciencemag.org/content/220/4598/671>.
- [25] X. L. Xie e G. Beni. “A validity measure for fuzzy clustering”. Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.8 (ago. de 1991), pp. 841–847. ISSN: 0162-8828. DOI: 10.1109/34.85677.
- [26] M. Dalla Mura, J. A. Benediktsson, F. Bovolo et al. “An Unsupervised Technique Based on Morphological Filters for Change Detection in Very High Resolution Images”. Em: *IEEE Geoscience and Remote Sensing Letters* 5.3 (jul. de 2008), pp. 433–437. ISSN: 1545-598X. DOI: 10.1109/LGRS.2008.917726.

- [27] R. G. Negri, A. C. Frery, W. Casaca et al. “Spectral-Spatial-Aware Unsupervised Change Detection With Stochastic Distances and Support Vector Machines”. Em: *IEEE Transactions on Geoscience and Remote Sensing* (2020), pp. 1–14.
- [28] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor et al. “Estimating the Support of a High-Dimensional Distribution”. Em: *Neural Computation* 13.7 (jul. de 2001), pp. 1443–1471. ISSN: 0899-7667. DOI: 10.1162/089976601750264965.

Anais

- [9] T. S. Korting, G. Erthal, L. V. Dutra et al. “Improvements to Expectation-Maximization approach for unsupervised classification of remote sensing data”. en. Em: *Anais... Simpósio Brasileiro de Geoinformática*, 9. (GEOINFO). São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2007, pp. 3–11. ISBN: 978-85-17-00036-2. URL: <http://urlib.net/sid.inpe.br/mtc-m18/2014/12.09.13.41>.
- [23] D. E. Gustafson e W. C. Kessel. “Fuzzy clustering with a fuzzy covariance matrix”. Em: *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*. Jan. de 1978, pp. 761–766. DOI: 10.1109/CDC.1978.268028.

