



UNIVERSIDADE DO VALE DO TAQUARI
CURSO DE ENGENHARIA DA COMPUTAÇÃO

**AVALIAÇÃO DA TÉCNICA GLCM PARA ANÁLISE DE TEXTURAS
EM IMAGENS DIGITAIS**

Bruno Luis Büchner

Lajeado, novembro de 2020

BRUNO LUIS BÜCHNER

AVALIAÇÃO DA TÉCNICA GLCM PARA ANÁLISE DE TEXTURAS EM IMAGENS DIGITAIS

Trabalho de conclusão de curso apresentado no Centro de Ciências Exatas e Tecnológicas, da Universidade do Vale do Taquari – Univates, como parte dos requisitos para obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Prof. Me. Juliano Dertzbacher.

Lajeado, novembro de 2020

AGRADECIMENTOS

À minha família, meus pais Delise e Clécio e ao meu irmão Gabriel, por sempre me concederem a condição necessária para cursar uma graduação, por atenderem as minhas necessidades de deslocamento sempre que ocorresse algum imprevisto, por tudo.

Ao professor e orientador Me. Juliano Dertzbacher pelas orientações, conselhos e principalmente pela paciência. Por sua dedicação e atenção que possibilitaram a realização desse trabalho.

Aos demais professores com que tive contato, pelos ensinamentos transmitidos durante os anos de graduação.

Ao professor Dr. Marcelo de Gomensoro Malheiros por ter sido o melhor coordenador de curso que eu poderia pedir, por sempre estar presente quando foi necessário.

Aos colegas e amigos que fiz durante a graduação, pelo conhecimento e caronas compartilhadas durante as viagens de carro e de ônibus de Venâncio Aires até Lajeado. Aos meus amigos de coração, que se mostraram presente em toda a minha jornada acadêmica, na motivação para a conclusão da graduação, nos auxílios com trabalhos e estudos e, principalmente aos amigos dedicados que me auxiliaram nas ideias de implementação, revisões e sugestões no decorrer de todo o trabalho de conclusão.

RESUMO

A área de processamento de imagens e análise de textura apresentam técnicas que nem sempre correspondem às necessidades. Estas técnicas, chamadas descritores de textura, utilizam características de objetos presentes na imagem, como brilho, cor, tamanho e formato. O presente trabalho tem como objetivo discorrer sobre as dificuldades existentes na discriminação de textura através do uso de GLCM como ferramenta e, através de estudos em trabalhos relacionados e em outras técnicas presentes na literatura, encontrar uma forma de aprimorar a classificação de texturas a partir de descritores obtidos pela GLCM. Para isto, foram realizados testes para validação do método proposto, utilizando um conjunto de imagens sintéticas divididas em 4 classes diferentes. As imagens foram submetidas a um processo de descrição textural por meio de equações, resultando em valores numéricos para cada característica da textura. Através de comparação de resultados, tem-se o uso de GLCM como descritor textural comprovado, pois nos testes realizados, obteve-se uma precisão de acerto próximo a 98% em imagens de estruturas texturais simples, e uma precisão variável de 80% a 60% em testes sobre imagens de estruturas complexas. O aprimoramento das métricas utilizadas assim como a configuração ideal da ferramenta podem substituir a necessidade do uso de outras técnicas de processamento de imagem.

Palavras-chave: Processamento de Imagens. Análise de textura. Aprimoramento da técnica. GLCM.

ABSTRACT

The field of image processing and texture analysis shows techniques which not always correspond to needs. These techniques, called texture descriptors, use characteristics of objects present in the image, such as brightness, color, size and shape. The present work aims to discuss the existing difficulties in texture discrimination through the use of GLCM as a tool and, through studies in related work and other techniques present in the literature, to find a way to improve the texture classification from descriptors obtained by GLCM. For this, tests were carried out to validate the proposed method, using a set of synthetic images divided into 4 different classes. The images were subjected to a process of textural description using equations, resulting in numerical values for each characteristic of the texture. Through comparison of results, the use of GLCM is a proven textural descriptor, because in the tests performed, accuracy close to 98% was obtained in images of simple textural structures, and a variable precision of 80% to 60% in tests on images of complex structures. The improvement of the metrics used as well as the ideal configuration of the tool can replace the need to use other image processing techniques.

Keywords: Image Processing. Texture Analysis. Improvement of the technique. GLCM.

LISTA DE FIGURAS

Figura 1: Esquema de um sistema de Visão Computacional.	19
Figura 2: exemplo de processamento de imagem no programa IMAGE LAB.	20
Figura 3: Características encontradas em imagem derivadas por equações.	24
Figura 4: Onde a) Imagem original com 4 texturas diferentes e b) resultado esperado da segmentação.	27
Figura 5: Diagrama do processo utilizado para segmentação de texturas.	27
Figura 6: Exemplos de imagens geradas para testes de segmentação: a) imagem com 4 texturas (Tipo A); b) imagem com 5 texturas (Tipo B); e c) imagem com região circular (Tipo C).	29
Figura 7: Resultado da segmentação utilizando decomposição Quadtree e o índice Q para comparar as janelas usando descritores de textura: a) GLCM (35 regiões); b) LBP (142 regiões) e c) SDH (152 regiões).	30
Figura 8: Representação do LBP para diferentes valores de P e R.	31
Figura 9: Imagem Quickbird-2 utilizada no trabalho. Classificação de Uso (A); Mapa de uso do solo (B).	32

Figura 10: Comparação de resultados da classificação com os descritores LBP, LPQ e GLCM das áreas de estudo.....	35
Figura 11: Abordagem de multi-escala em GLCM. (a) Decomposição em pirâmide; (b) Suavização Gaussiana.	38
Figura 12: Exemplificação das amostras de texturas utilizadas no trabalho extraídas dos respectivos conjuntos de dados. (a) extraídos por UMD; (b) extraídos por UIUC; (c) extraídos por OuTex; (d) extraídos por VisTex; (e) extraídos por Brodatz.	39
Figura 13: Visão geral da metodologia proposta. Consiste em seis etapas: extração, normalização, suavização, remoção de ruído, agrupamento e segmentação.....	42
Figura 14: Extração original da GLCM (esquerda) e imagem processada pela equação (2) (direita).	43
Figura 15: Exemplo de processamento de redução de ruído. Imagem pré-processamento (esquerda) e imagem pós-processamento (direita).	44
Figura 16: Amostras de figuras antes (esquerda) e depois (direita) da segmentação.	44
Figura 17: Características utilizadas neste trabalho para descrição de textura.....	49
Figura 18: Importação da biblioteca Matplotlib.	50
Figura 19: Importação das bibliotecas utilizadas, Matplotlib e Scikit-Image.....	51
Figura 20: Modelo utilizado nos testes preliminares.....	51
Figura 21: Seleção das categorias das imagens a serem analisadas.	51
Figura 22: Seção da leitura das imagens em tons de cinza.	52
Figura 23: Seleção das características e métricas a serem extraídas da GLCM. .	52
Figura 24: Exibição dos resultados obtidos.	52
Figura 25: Algoritmo utilizado para realização dos experimentos.	54

Figura 26: Conjunto de amostras de imagens sintéticas classificadas por ratio. (a) Ratio 4; (b) Ratio 5; (c) Ratio 8; (d) Ratio 12.	56
Figura 27: Resultados encontrados a partir do primeiro conjunto de amostras.....	57
Figura 28: Exemplificação dos dados extraídos pela GLCM.	58

LISTA DE GRÁFICOS

Gráfico 1: Representação dos resultados da primeira rodada de teste.....	60
Gráfico 2: Resultado em porcentagem da primeira rodada de teste.	60
Gráfico 3: Representação dos resultados da segunda rodada de teste.	61
Gráfico 4: Resultado em porcentagem da segunda rodada de teste.	62
Gráfico 5: Representação dos resultados da terceira rodada de teste.....	63
Gráfico 6: Resultado em porcentagem da terceira rodada de teste.	63
Gráfico 7: Representação dos resultados da quarta rodada de teste.	64
Gráfico 8: Resultado em porcentagem da quarta rodada de teste.	65

LISTA DE TABELAS

Tabela 1: Comparação Inter-Classes utilizando o índice Q sobre a GLCM.	28
Tabela 2: Resultados da classificação para descritores de textura mono-escalares formados por diferentes combinações de $LBP_{P,R}$ e $VAR_{P,R}$	33
Tabela 3: Resultados da classificação para descritores de textura multi-escalares formados por diferentes combinações de $LBP_{P,R}$ e $VAR_{P,R}$	34
Tabela 4: Espécies que compõem a Base de Dados.	36
Tabela 5: Taxa de reconhecimento por classificador.	37
Tabela 6: Taxas de classificação (%) adquiridas quando decomposição em pirâmide é considerada na GLCM e diversas abordagens para combinação de descritores são empregados.	40
Tabela 7: Taxas de classificação (%) adquiridas quando suavização Gaussiana é considerada na GLCM e diversas abordagens para combinação de descritores são empregados.	41
Tabela 8: Taxas de classificação (%) adquiridas por diversos descritores texturais em diferentes conjuntos de dados. A segunda coluna indica a quantidade de amostras utilizadas.	41

LISTA DE ABREVIações E SIGLAS

BSD	<i>Berkeley Software Distribution</i>
GLCM	<i>Grey Level Co-Occurrence Matrix</i>
ILBP	<i>Improved Local Binary Patterns</i>
LBP	<i>Local Binary Patterns</i>
LPQ	<i>Local Phase Quantization</i>
MM	Morfologia matemática
PSF	<i>Python Software Foundation</i>
SDH	<i>Sum and Difference Histograms</i>
SVM	<i>Support Vector Machine</i>
VAR	Variância local

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Problema de Pesquisa	14
1.2 Objetivos	15
1.3 Estrutura do Trabalho	16
2 REFERENCIAL TEÓRICO.....	17
2.1 Visão Computacional	17
2.2 Processamento de Imagens Digitais.....	19
2.3 Análise de Textura	21
2.4 GLCM	23
3 TRABALHOS RELACIONADOS.....	26
3.1 Texture-Based Segmentation Using Image Fidelity Indexes	26
3.2 Uma avaliação de descritores de textura baseados em códigos binários locais para classificação de imagens de sensoriamento remoto	30
3.3 Uso de descritores de textura extraídos de GLCM para o reconhecimento de padrões em diferentes domínios de aplicação.....	35
3.4 Multi-scale gray level co-occurrence matrices for texture description	37
3.5 Texture Segmentation Using Different Orientations of GLCM Features	42
3.6 Comparativo com o trabalho proposto	45
4 MATERIAIS E MÉTODOS	46
4.1 Tecnologias	48
4.1.1 Python.....	48
4.1.2 Jupyter	49

4.1.3 Scikit-Image	49
4.1.4 Matplotlib	50
4.2 Desenvolvimento	50
5 TESTES E RESULTADOS	54
5.1 Rodadas de testes	59
5.1.1 Rodada de teste 1	59
5.1.2 Rodada de teste 2	61
5.1.3 Rodada de teste 3	62
5.1.4 Rodada de teste 4	64
5.2 Síntese dos testes	65
6 CONCLUSÃO	67
REFERÊNCIAS BIBLIOGRÁFICAS	70

1 INTRODUÇÃO

A Visão Computacional está diretamente ligada à capacidade de classificação e reconhecimento de imagens, na maneira que ocorre um repasse de informação da imagem ao computador. Esse repasse é compreendido desde a aquisição da figura até o reconhecimento de padrões que permite a classificação da imagem.

Uma das principais áreas de Visão Computacional é o Processamento de Imagens, o qual possui a responsabilidade de tratamento das características presentes na imagem, desde seu pré-processamento que engloba sua criação, até seu pós-processamento que abrange o refinamento de suas características conforme a configuração da aplicação utilizada. Entre as principais características de uma imagem está a textura.

Define-se textura como um padrão de cores e estruturas que recobre objetos. Dentro do campo de processamento de imagens, textura é um dos atributos mais importantes presentes nelas, devido a um padrão visual que geralmente está relacionado à distribuição de pixels (elemento da imagem ao qual é atribuído cor). Essa distribuição resulta nas características de objetos e regiões da imagem.

Existem diversas técnicas de análise de textura na literatura, as quais tentam representar o conteúdo presente em imagens digitais através de equações, que resultam em valores numéricos preparados para serem analisados e classificados. Estas técnicas, chamadas descritores de textura, utilizam características de objetos

presentes na imagem, como brilho, cor, tamanho e formato. Os atributos calculados neste trabalho derivam das matrizes de co-ocorrência (*Grey Level Co-Occurrence Matrix* – GLCM).

Considerando o elevado número de variáveis presentes na classificação de texturas, este trabalho busca estudar e testar as melhores condições para a análise de textura em níveis de cinza. A GLCM permite destacar e isolar os principais parâmetros na análise, possibilitando o desenvolvimento de uma ferramenta comparativa e analítica sobre imagens semelhantes, com o intuito de avaliar e apurar a classificação de imagens.

1.1 Problema de Pesquisa

O processamento de imagens e análise de texturas nem sempre são eficientes. A eficiência da análise de textura está diretamente relacionada a configuração utilizada na extração das características, caso a configuração da ferramenta que será utilizada para a extração não estiver orientada à situação a ser analisada, pode ocorrer de não ser encontrado nenhuma relação entre os dados extraídos.

Um dos motivos que dificultam a eficiência dos extratores de textura é que nem todas as características seguem padrões, a irregularidade presente em algumas texturas prejudica a extração de recursos, comprometendo sua análise e dificultando sua classificação.

Na literatura de processamento de imagens e análise de textura existem diversas técnicas já testadas por diversos autores e, entre elas, a GLCM. O uso da GLCM como ferramenta de extração implica na melhor forma de configuração que propicie resultados comparativos, mas a configuração em questão muitas vezes necessita de um tratamento designado ao tentar comparar imagens, utilizando características isoladamente ou simultaneamente na comparação.

As características, também chamadas de propriedades, a serem extraídas possuem grande importância no processo de classificação, de maneira que

algumas possam ser responsivas à extração de forma a gerar resultados suficientes para uma comparação digna, enquanto algumas características não possuem propriedades vinculadas o suficiente para que seja utilizada na classificação.

1.2 Objetivos

O presente trabalho foi elaborado com o objetivo de validar a técnica de análise de textura GLCM como um descritor textural digno, através de testes em imagens sintéticas. Mediante a exploração de recursos de processamento de imagem, é compreendido como objetivo, além da validação da técnica proposta, o refinamento da classificação de imagem na sequência do trabalho.

No decorrer do desenvolvimento do trabalho, objetiva-se um estudo sobre métodos de análise de textura existentes na literatura, assim como em trabalhos relacionados ao tema, com o intuito de estabelecer fundamentos para o desenvolvimento da ferramenta, padrões para a avaliação dos resultados e uma comparação literária com outros trabalhos.

Com o objetivo de explorar as capacidades da técnica GLCM, serão realizados testes preliminares para serem tomados como base nos testes subsequentes. A validação da técnica será tomada através da comparação e análise de resultados, e a partir da validação, será feita uma tentativa de refiná-la através da inclusão de outras configurações ao experimento. Por fim, será realizada uma comparação digna das imagens a partir do processo de classificação para determinar a eficiência da ferramenta a partir das configurações aplicadas na técnica GLCM.

Após a finalização do trabalho, será disponibilizado o estudo realizado para auxiliar em trabalhos futuros relacionados ao tema, elucidando as melhores características extraídas da GLCM, sugerindo possíveis melhorias a serem feitas e configurações para serem aplicadas juntas à GLCM.

1.3 Estrutura do Trabalho

O trabalho está estruturado em seis capítulos, sendo que o primeiro deles faz uma breve introdução ao trabalho, falando sobre os problemas relacionados à análise de textura, integrando também a justificativa e objetivos do trabalho. O segundo capítulo refere-se ao referencial teórico, o qual tem o intuito de explicar melhor o assunto de Visão Computacional, técnicas de processamento de imagens digitais, análise de textura e a técnica GLCM, servindo para fundamentar os objetivos propostos. No terceiro capítulo é detalhado trabalhos existentes com uma abordagem similar, onde serão explanados e comparados ao presente trabalho. O quarto capítulo do estudo apresenta as tecnologias utilizadas e a serem utilizadas, além de um teste preliminar de funcionamento para validar a proposta deste trabalho, pois esse método será utilizado como a principal ferramenta de testes sequentes neste estudo. O quinto capítulo compreende os testes realizados a partir do teste preliminar, onde são empregados a metodologia e o desenvolvimento definidos no capítulo 4. O sexto capítulo contempla as considerações finais onde serão comentados os resultados do experimento, as dificuldades encontradas e possíveis continuidades para o trabalho. Ao final são apresentadas as referências bibliográficas consultadas para a fundamentação teórica do estudo.

2 REFERENCIAL TEÓRICO

Este capítulo fundamenta o estudo realizado através de referências bibliográficas. Serão apresentados aspectos relevantes para a compreensão deste trabalho, como Visão Computacional, Processamento de Imagem, Análise de Textura e GLCM, que é o método proposto para desenvolvimento desta monografia.

2.1 Visão Computacional

Se dá pelo nome de Visão Computacional a tecnologia e ciência das máquinas que enxergam. Ela prioriza o desenvolvimento teórico e tecnológico com a finalidade da construção de sistemas artificiais que colhem informação de imagens ou dados multidimensionais (ANTONELLO, 2014).

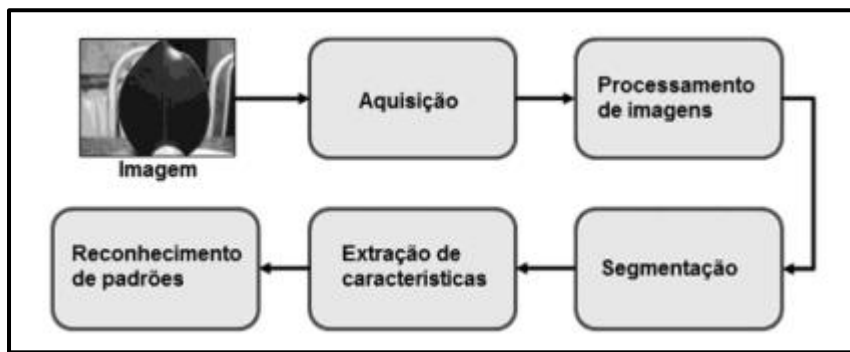
A Visão Computacional pode ser definida como a área de estudo que busca repassar para máquinas a capacidade de visão (BACKES e JUNIOR, 2019). Segundo estes autores, a visão consiste na captação de imagens, além da possibilidade de aperfeiçoamento, como por exemplo, na retirada de ruídos e no aumento de contraste. Também engloba a separação de regiões ou objetos de interesse, a extração de informações relevantes da imagem tais como cor, textura, e por fim, associar imagens com outras catalogadas previamente.

Segundo os autores Backes e Junior (2019), um sistema de Visão Computacional é constituído de várias fases, sendo elas:

- **Aquisição:** fase responsável por captar imagens. Dispositivos como *scanners*, filmadoras e máquinas fotográficas simulam as funções dos olhos.
- **Processamento:** responsável por “melhorar” a imagem, no caso, o saneamento de possíveis problemas existentes, como por exemplo, retirar ruídos, salientar bordas e suavizar a imagem. Essa fase de processamento pode ser um fim em si mesma ou ter continuidade ao fornecer uma imagem mais adequada para futuras fases. É uma etapa que abrange “pré-processamento”, como a rotação da imagem, até processamentos complexos como filtragens, por exemplo.
- **Segmentação:** fase responsável por particionar a imagem em regiões de interesse. Por exemplo, uma imagem de paisagem pode possuir áreas de possível interesse como céu, vegetação, ou apenas um objeto específico existente na imagem, que são separados automaticamente.
- **Extração de características / análise de imagens:** é a fase responsável em extrair um conjunto de características do objeto de interesse.
- **Reconhecimento de padrões:** fase responsável pela classificação ou agrupamento das imagens com base em seus conjuntos de características. Por exemplo, ao ver uma foto de uma laranja, poderá indicar que o objeto contido na imagem pertence à classe “laranja”, com base em alguns atributos como cor, formato e tamanho.

Tal como enfatizado por Backes e Junior (2019), não existe uma descrição única para cada uma das fases de um sistema de Visão Computacional. Por exemplo, a denominação que alguns autores utilizam para “pré-processamento” utilizaremos como “processamento de imagem”, e chamam de “processamento de imagem” como um somatório de todas as cinco fases apresentadas. O número de fases também é variável de acordo com autores. Além de que certas fases podem ser anuladas dependendo do problema. A Figura 1 apresenta um esquema simples de um sistema de Visão Computacional.

Figura 1: Esquema de um sistema de Visão Computacional.



Fonte: Backes e Junior (2019).

A área de estudos sobre Visão Computacional está sempre em evolução. Com base neste ponto, é relevante destacar que, além de fazer máquinas enxergarem, através do reconhecimento de objetos, paisagens, gestos, faces e padrões, por exemplo, tais algoritmos possuem capacidade de serem utilizados em reconhecimento de padrões em bases de dados, não obrigatoriamente feitos de imagens (ANTONELLO, 2014).

2.2 Processamento de Imagens Digitais

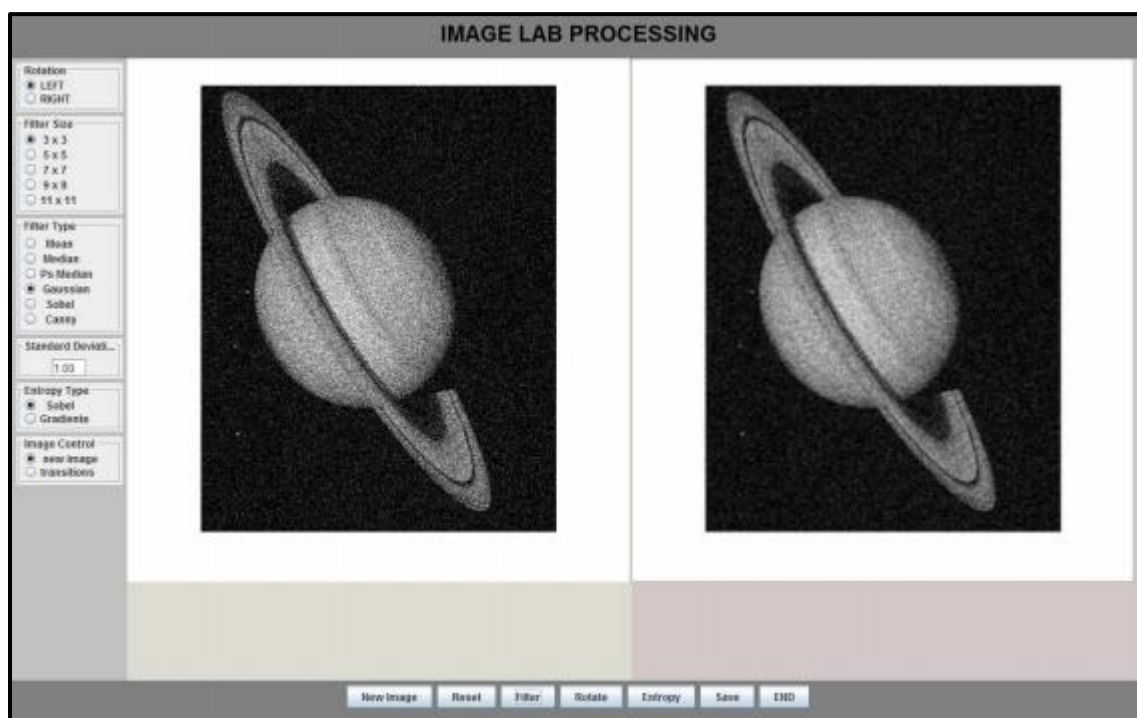
Uma imagem digital é representada usando uma matriz bidimensional, que contém números codificados, o que permite seu armazenamento, transferência, impressão ou até mesmo reprodução, e por fim seu processamento em meios eletrônicos (JESUS e COSTA, 2015).

O termo Processamento de Imagens Digitais vem diretamente da utilização de um computador para realizar o processamento de imagens. Historicamente, os primeiros usos de aplicações de imagens digitais datam do início de 1920 na imprensa escrita, na época transmitidas entre Nova York e Londres por um cabo subaquático (VALENÇA, 2011).

Na década de 30 do século XX, foram elaboradas mais aplicações e experiências nos processos desenvolvidos. Contudo, ainda que envolvam imagens digitais, conforme a definição apresentada, não existe processamento digital sem a utilização de computadores. O desenvolvimento do computador digital e das

linguagens de programação estão profundamente ligados ao processamento digital de imagens (VALENÇA, 2011). Visto a necessidade de cálculo e capacidade de armazenamento, a evolução do processamento digital de imagens tem estado vinculada ao desenvolvimento dos computadores. Para mais, a urgência em apoiar tecnologias que incluam visualização, armazenamento e transmissão de dados também impulsionou a sua evolução. Devido às aplicações espaciais, as técnicas de processamento de imagens foram muito estimuladas até que, posteriormente, levassem ao início da sua utilização na Medicina, entre o final de 1960 e início de 1970 (VALENÇA, 2011).

Figura 2: exemplo de processamento de imagem no programa IMAGE LAB.



Fonte: Jesus e Costa (2015).

O processamento de imagens tem uma enorme variedade de aplicações, como pode ser visto na Figura 2. em quase todas as áreas da ciência e tecnologia. Geralmente, o processamento de imagens digitais requer um extenso trabalho experimental envolvendo software de simulação e teste de inúmeras imagens (ROCHA *et al.*, 2013). Por exemplo, o processamento digital de imagens permite detectar descontinuidades na imagem, i.e., pontos onde ocorrem mudanças bruscas no nível da intensidade dos *pixels* (VALENÇA, 2011).

Frequentemente, as técnicas de pré-processamento de imagens possuem o intuito de corrigir, ou ao menos reduzir distorções geométricas e radiométricas da imagem quando introduzida durante sua aquisição, juntamente com a eliminação de ruído, permitindo o destaque de objetos ou características (VALENÇA, 2011). O propósito destas operações está diretamente ligado com o aprimoramento da percepção de todos os dados considerados relevantes na imagem. Nesta etapa do processamento, os filtros/transformações se encontram agrupados em dois tipos: alteração do histograma da imagem e alteração da intensidade dos *pixels*. No caso da alteração do histograma da imagem, as intensidades da imagem de entrada serão mapeadas em valores na imagem de saída, de maneira que uma determinada porcentagem de dados fique saturada em faixas de acentuação extremas (altas e baixas), o resultado dessas operações permite o aumento do contraste da imagem, assim como o realce de objetos existentes na imagem. No caso da alteração da intensidade dos *pixels*, é realizado a aplicação de um determinado fator, onde ocorre a alteração uniforme de intensidades, levando a uma diminuição do contraste da imagem de saída (VALENÇA, 2011).

Por outro lado, as técnicas de pós-processamento de imagem procuram abranger a seleção de objetos com características pré-definidas. Um exemplo é a aplicação de técnicas de morfologia matemática (MM), efetuada sobre imagens binárias e com um vasto campo de aplicação.

2.3 Análise de Textura

A análise de textura é uma das formas de considerar o contexto no qual cada pixel é inserido em uma imagem. Com o uso de métodos que realçam a textura, as feições da textura resultantes podem ser acrescentadas ao processo de classificação, aumentando assim a precisão. Uma das ferramentas que permite aumentar ainda mais a precisão da classificação que usa unicamente parâmetros espectrais é a imagem de alta resolução (HAY, 1993 apud SETTE *et al.* 2011).

Para Emerson, Lam e Quattrochi (1999), a aplicação de um método de análise de textura gera índices que possuem relação com determinados fenômenos

geográficos presentes numa imagem digital. O método de análise de textura opera baseado na variabilidade de mudança geográfica, assimilando a resolução e escala da imagem.

Tradicionalmente, quando ocorre uma classificação de uma imagem, exclusivamente os tons de cinza são analisados e não usufrui do conteúdo contextual das imagens como seria feito numa fotointerpretação, este é um dos comportamentos do descritor GLCM que será elucidado na seção 2.4 (HAY, 1993 apud SETTE *et al.* 2011).

Um método simples de extração de características de imagens é o LBP (*Local Binary Pattern*). Considerado um método convalidado, é considerado uma das melhores formas de extrair características relacionadas a textura. Suas vantagens em relação a outras aplicações vêm da sua não influência por variações de iluminação, habilidade de codificar detalhes e possibilitar análise local das informações (SILVA *et al.* 2015).

De acordo com Ribeiro, Paiva e Jorge (2014) LBP nada mais é do que um descritor de textura com a função de analisar a imagem em comparação com as intensidades relativas entre os pixels em uma parcela da imagem. Uma imagem em tons de cinza ao ser analisada pelo LBP é feita uma delimitação espacial de 3x3 ao redor do pixel central em foco. Essa delimitação espacial acaba por ser limiarizada, transformação de uma imagem de níveis de cinza numa imagem definida a preto e branco, pelo valor do pixel central gerando assim, uma matriz com valores binários.

Outra técnica diversa de descritor textural seria o LFP (*Local Fuzzy Pattern*). Originalmente o LFP foi proposto como um descritor focado em análise de micro padrões de imagens digitais. A funcionalidade anormal deste descritor vem da sua forma de tratar um determinado padrão da imagem, onde são utilizados conjuntos *fuzzy* para tratar uma delimitação espacial de pixels e um número *fuzzy* para abordar o nível de cinza de cada pixel, ao invés de apenas utilizar um conjunto de regras *fuzzy* para se obter um código final que descreva um certo padrão da imagem (RIBEIRO, PAIVA e JORGE, 2014).

Segundo Musci *et al.* (2011), o método de LPQ (*Local Phase Quantization*) foi idealizado com o intuito de superar o LBP, proporcionando soluções e ajustes

em aplicações onde as imagens estão afetadas por iluminação não uniforme ou borradas. Essencialmente similar ao LBP, o LPQ computa um código para cada pixel determinado com o objetivo de representar a textura de pixel numa vizinhança. O LPQ utiliza da Transformação de Fourier para fazer a quantização de fase, considerando exclusivamente o sinal das componentes real e imaginária da transformada de Fourier em ao menos quatro valores computados próximos ao pixel de origem.

2.4 GLCM

A definição dada para matriz de co-ocorrência de níveis de cinza (*Gray Level Co-occurrence Matrix* – GLCM) é a de um método estatístico para descrição de texturas. A GLCM tem a característica de armazenar a probabilidade de que dois valores de intensidade de cinza estejam envolvidos por uma definida relação espacial. A partir desta matriz de probabilidades diversas medidas estatísticas são retiradas, a fim de caracterizar a textura existente em uma imagem. Entre estas medidas estão contraste, energia, entropia, correlação, homogeneidade, momento de terceira ordem e probabilidade máxima. Partindo da matriz de co-ocorrência, é possível extrair características de textura usando estas medidas. Tais características podem ser vistas através de equações na Figura 3 (MARTINS *et al.*, 2011).

Segundo Martins *et al.* (2011), a capacidade de armazenar a probabilidade de que dois valores de intensidade de cinza estejam envolvidos por uma relação espacial definida permite que uma GLCM corresponda a uma matriz de ordem $N_g \times N_g$. Diante disso, alguns parâmetros como a distância d entre os pixels e o ângulo θ definido pela orientação da reta que transcorre pelos pixels são relevantes na caracterização da relação espacial. De forma que $p(i,j)$ é a probabilidade de ocorrência das intensidades de cinza i e j , considerando a distância $d=1$ e com determinado ângulo θ .

Figura 3: Características encontradas em imagem derivadas por equações.

$$\begin{aligned}
 (1) \quad & \text{Contraste} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i-j)^2 p(i,j) \\
 (2) \quad & \text{Energia} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (p(i,j))^2 \\
 (3) \quad & \text{Entropia} = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \log p(i,j) \\
 (4) \quad & \text{Homogeneidade} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{p(i,j)}{1 + (i-j)^2} \\
 (5) \quad & \text{Momento de terceira ordem} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)(i-j)^3 \\
 (6) \quad & \text{Probabilidade máxima} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \max p(i,j) \\
 (7) \quad & \text{Correlação} = \frac{p(i,j) - \mu_x \mu_y}{\sigma_x^2 \sigma_y^2}
 \end{aligned}$$

onde $\mu_x = \sum_{i=1}^{N_g} i \times p_x(i)$, $p_x(i) = \sum_{j=1}^{N_g} p(i,j)$, $\sigma_x^2 = \sum_{i=1}^{N_g} (i - \mu_x)^2 p_x(i)$,
 $\mu_y = \sum_{j=1}^{N_g} j \times p_y(j)$, $p_y(j) = \sum_{i=1}^{N_g} p(i,j)$ e $\sigma_y^2 = \sum_{j=1}^{N_g} (j - \mu_y)^2 p_y(j)$.

Fonte: Martins *et al.* (2011).

Já a definição de GLCM por Jucá e Mello (2011) indica uma matriz quadrada $C \times C$, onde C é a resolução de cor da imagem. O autor relata um exemplo usando imagens que possuem 256 tons de cinza, onde o GLCM tem uma dimensão de 256 x 256. Com isso, a célula (i, j) demonstra quantas vezes a cor j mostra-se longe da cor i em uma distância (dx, dy) determinadas anteriormente e de maneira heurística de acordo com a aplicação.

Para Silva *et al.* (2015), o conceito para *Gray Level Co-occurrence Matrix* (GLCM) se baseia em um descritor para análise de textura em imagens, na qual expõe em forma de matriz quadrada $L \times L$ uma medida probabilística de ocorrência de cores separadas uma dada distância numa certa direção. Com isso, é possível mapear o número de transições de intensidade entre *pixels* de uma imagem I , conforme este critério de vizinhança.

Em estudos presentes na literatura é observado que muitas das características extraídas da GLCM são correlacionadas e seu uso simultâneo não

traz benefícios na maioria dos casos (MARTINS *et al.* 2011). O presente trabalho pretende testar e apontar as características extraídas pela GLCM de maior relevância na comparação, suas condições de manuseio e seu comportamento em relação às outras características. A seguir serão apresentados trabalhos relacionados aos conceitos elucidados neste capítulo, exemplificando as demais possibilidades de desenvolvimento e metodologia, que servem de embasamento para o desenvolvimento deste trabalho.

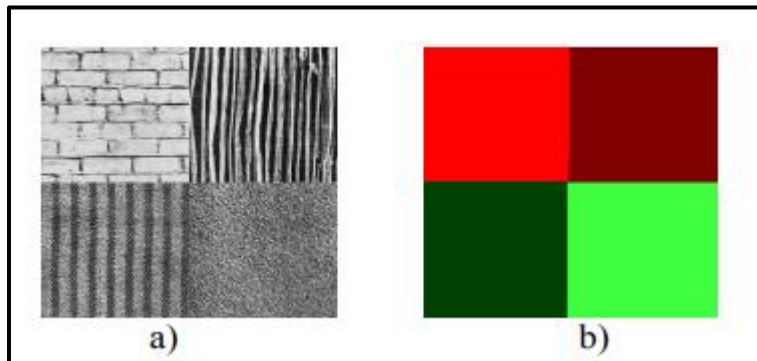
3 TRABALHOS RELACIONADOS

Este capítulo apresenta um estudo sobre métodos e trabalhos relacionados com a área de estudo do presente trabalho. Eles servem de apoio para o desenvolvimento do método proposto. Com o objetivo de esclarecer a pesquisa, a última seção traz um comparativo entre o presente trabalho e os trabalhos estudados, de forma a validar o método proposto.

3.1 Texture-Based Segmentation Using Image Fidelity Indexes

O trabalho realizado por Jucá e Mello (2011), busca uma forma diferente de comparar a semelhança entre texturas pelo meio de índices de fidelidade. Utiliza descritores de textura como GLCM (Matriz de Co-ocorrência de Tons de Cinza – *Grey Level Co-occurrence Matrix*), SDH (Histogramas da Soma e da Diferença – *Sum and Difference Histograms*) e o LBP (*Local Binary Pattern*). O trabalho faz com que uma imagem possa ser analisada em janelas “deslizantes”, com o intuito de identificar texturas com características parecidas, de maneira a ajudar a determinar regiões com diversas texturas. O resultado esperado do processo de segmentação, descrito pelos autores, pode ser visto na Figura 4.

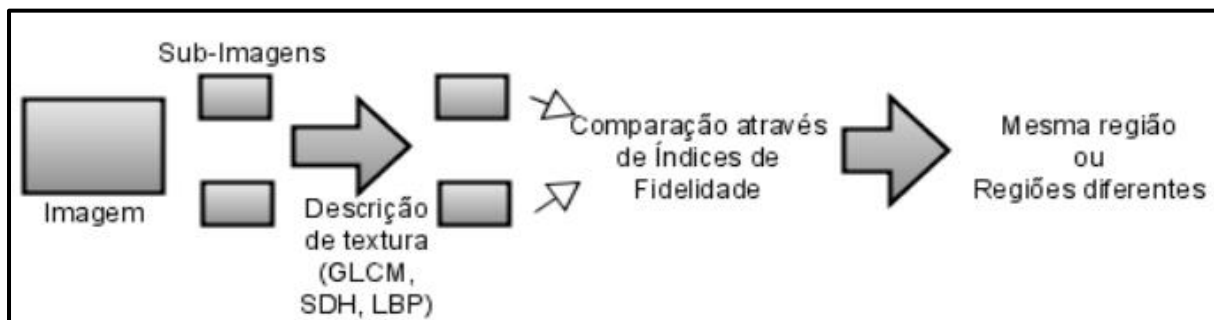
Figura 4: Onde a) Imagem original com 4 texturas diferentes e b) resultado esperado da segmentação.



Fonte: Adaptado pelo autor com base em Jucá e Mello (2011).

A maneira pela qual a técnica funciona é apresentada na Figura 5. Primeiramente, a imagem deve ser dividida em janelas. A partir disso, cada janela deve ser retratada por um descritor de textura. Logo, cada janela é então comparada entre si utilizando um índice de fidelidade. Caso ocorra de que o valor encontrado supere um limiar pré-determinado, as janelas são julgadas como partes de uma mesma região. O processo tem continuidade até as janelas seguintes e encontra seu fim quando todas as janelas já tenham sido comparadas entre si.

Figura 5: Diagrama do processo utilizado para segmentação de texturas.



Fonte: Adaptado pelo autor com base em Jucá e Mello (2011).

No respectivo trabalho, são utilizados três descritores de textura estatísticos: GLCM como já descrito no referencial teórico, LBP e SDH. De modo geral, os métodos estatísticos analisam a distribuição espacial dos tons de cinza, calculando características locais em cada ponto da imagem e derivando um conjunto de estatísticas a partir dessas distribuições, segundo os autores.

Com o propósito de auxiliar nas comparações, foi criado um índice de qualidade universal chamado Índice Q. Esse índice é independente às condições das imagens que estão sendo testadas, a partir das condições de visualização ou de cada um dos observadores, obtido pelo cálculo (1):

$$Q = \frac{4\sigma_{xy} \overline{xy}}{\sigma_x^2 + \sigma_y^2 + (\bar{x})^2 + (\bar{y})^2} \quad (1)$$

Uma das avaliações de similaridade usando descritores de textura é a do GLCM, onde pode ser visto na Tabela 1 uma comparação das texturas usando o índice Q aplicado à GLCM das imagens. Os autores frisam que é possível observar que os valores das comparações entre classes diferentes foram baixos o bastante para possibilitar que seja traçado um ponto de corte que separa as classes perfeitamente, apesar de que os valores nas comparações entre partições de mesma classe não se aproximem do valor máximo ($Q = 1$).

Tabela 1: Comparação Inter-Classes utilizando o índice Q sobre a GLCM.

	Comparação Inter-Classes					
	A1>A2	A1>A3	A1>A4	A2>A3	A2>A4	A3>A4
Classe1	0,6099	0,5115	0,6053	0,4925	0,6194	0,5073
Classe2	0,6337	0,6238	0,6221	0,6220	0,6503	0,6190
Classe3	0,5091	0,5545	0,4928	0,4846	0,5796	0,5074
Classe4	0,5337	0,6194	0,6036	0,5363	0,5910	0,6154

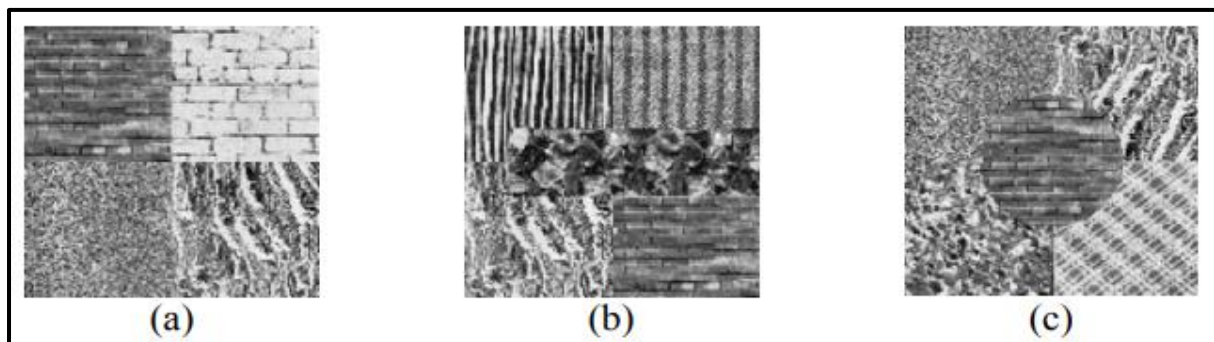
Fonte: Adaptado pelo autor com base em Jucá e Mello (2011).

As avaliações foram efetuadas também com as técnicas SDH e LBP. A técnica LBP apresentou ótimo comportamento na comparação inter-classes, atingindo valores elevados enquanto na comparação entre-classes os valores foram menores. Já a SDH não apresentou resultados satisfatórios, onde apenas o histograma da soma foi usado. No processo ocorreram problemas na classificação de algumas amostras.

No trabalho ocorreram outras avaliações, entre elas: a recuperação de textura baseada em conteúdo, segmentação de imagens utilizando decomposição *Quadtree* e segmentação de imagens baseada em textura. No primeiro caso, foi proposta uma nova métrica com intuito de avaliar a semelhança estrutural, através

da comparação de padrões locais de *pixels* que foram normalizados para luminância e contraste. No segundo, a imagem era segmentada em suas texturas utilizando o conceito de *split-and-merge*, onde primeiramente a imagem é dividida em regiões em acordo com algum critério (*split*), seguido do agrupamento dessas regiões (*merge*) conforme outro critério, a imagem dividida em regiões pode ser vista na Figura 6.

Figura 6: Exemplos de imagens geradas para testes de segmentação: a) imagem com 4 texturas (Tipo A); b) imagem com 5 texturas (Tipo B); e c) imagem com região circular (Tipo C).



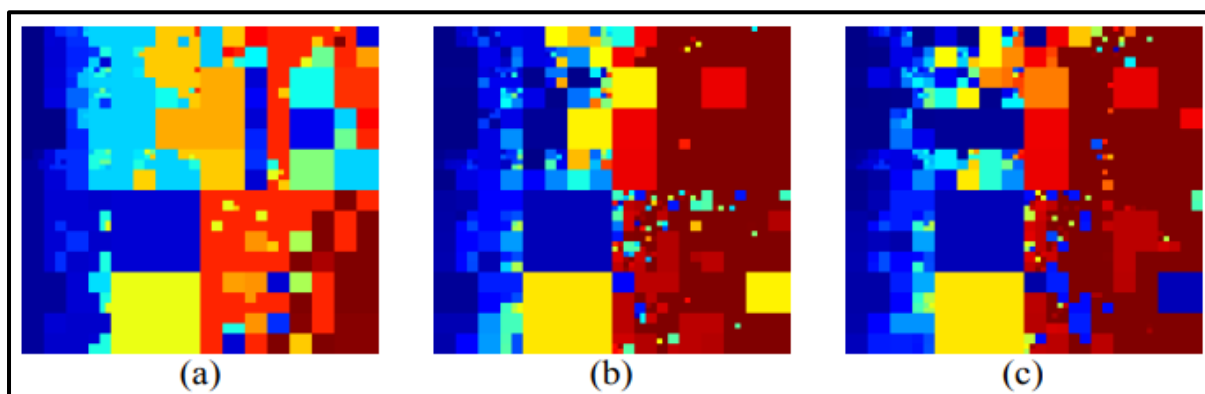
Fonte: Adaptado pelo autor com base em Jucá e Mello (2011).

No terceiro caso, o qual é mais relevante para o estudo deste trabalho de conclusão, refere-se diretamente à análise de textura. Para realizar este experimento, os autores geraram 18 imagens utilizando texturas da base de dados Brodatz. O processo seguiu da seguinte maneira:

1. Um mapa é gerado incluindo o valor de cada descritor a partir da análise da imagem completa.
2. Foi usado um ponto de corte definido para caracterizar como a janela deve ser classificada posteriormente. Empregando o índice de fidelidade Q, todas as janelas devem ser comparadas entre si, de forma que se o resultado for superior que o ponto de corte, a janela é classificada como pertencente à mesma classe das anteriores. Senão, uma nova região é considerada encontrada.

3. Assim que todas as janelas forem classificadas, gera-se uma imagem contendo as regiões que foram encontradas, como pode ser visto na Figura 7.

Figura 7: Resultado da segmentação utilizando decomposição Quadtree e o índice Q para comparar as janelas usando descritores de textura: a) GLCM (35 regiões); b) LBP (142 regiões) e c) SDH (152 regiões).



Fonte: Adaptado pelo autor com base em Jucá e Mello (2011).

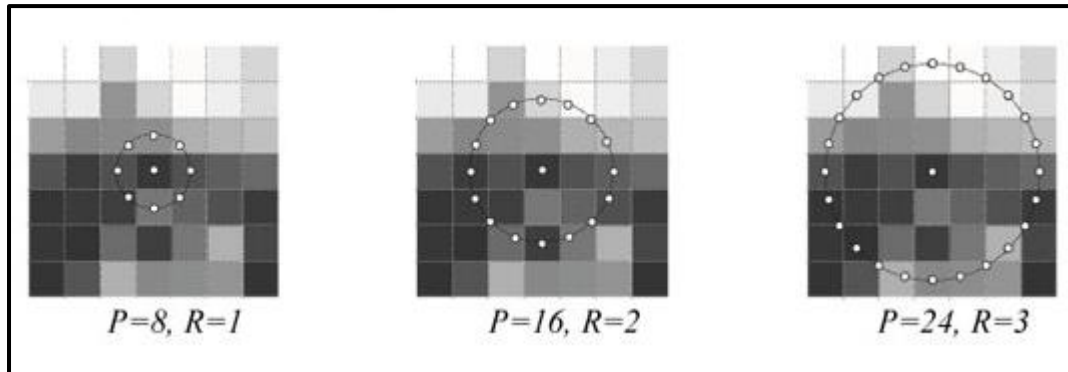
As conclusões dos autores indicam que mesmo com os avanços na área de Visão Computacional, até agora não é possível simular os demais estágios da visão humana. O trabalho descrito propôs um conceito novo, o uso de índices de fidelidade de imagens apoiado na comparação de janelas de textura na mesma imagem. O LBP expressou bons resultados com a finalidade de descrever e comparar imagens através do índice Q.

3.2 Uma avaliação de descritores de textura baseados em códigos binários locais para classificação de imagens de sensoriamento remoto

O trabalho realizado pelos autores Musci *et al.* (2011), publicado pela PUC-Rio no Boletim de Ciências Geodésicas, busca avaliar descritores texturais multi-escalares invariantes à rotação, baseados em Padrões Binários Locais (LBP) e também em Quantização de Fase Local (LPQ) visando a classificação de uso e cobertura do solo em uma imagem IKONOS-2 e uma imagem Quickbird-2.

Dentro da codificação de textura via LBP está P e R , no qual o LBP relacionado a um pixel na posição $w = (x,y)$ é calculado com base em um conjunto de P amostras espaçadas com igualdade e distribuídas sobre uma circunferência de raio R com o centro sobre o pixel, conforme a Figura 8.

Figura 8: Representação do LBP para diferentes valores de P e R .

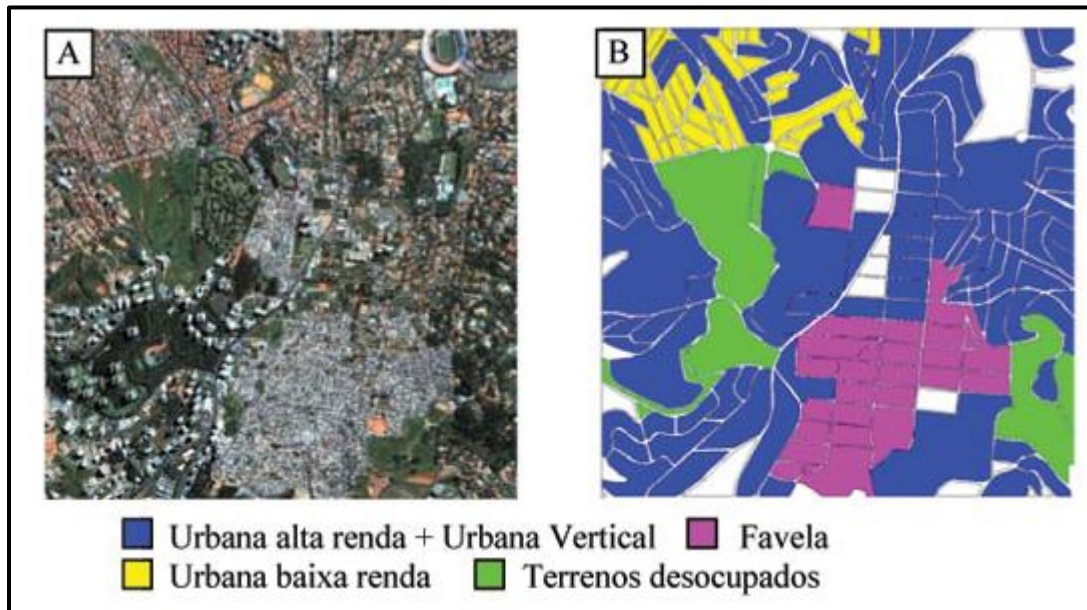


Fonte: Adaptado pelo autor com base em Musci *et al.* (2011).

O objetivo do trabalho está relacionado a uma avaliação do desempenho dos algoritmos LBP e LPQ como descritores de textura em aplicações de análise de imagens oriundas de Sensoriamento Remoto, no qual foi seguida uma abordagem baseada em objetos. Foram avaliadas no estudo diversas configurações multi-escalares dos algoritmos LBP e LPQ invariantes à rotação de imagens VHR. Como referência foram utilizados descritores baseados em GLCM.

Um exemplo de uma das áreas de estudo que foram utilizadas no trabalho se localiza na cidade de São Paulo. A imagem capturada pelo sensor Quickbird-2 permite ao estudo realizar uma classificação de uso do solo, buscando-se a classificação de objetos como quadras, cujos limites foram extraídos de dados cadastrais da Secretaria Municipal de Planejamento Urbano de São Paulo (PMSP, 2009). A Figura 9 mostra a composição da imagem utilizada.

Figura 9: Imagem Quickbird-2 utilizada no trabalho. Classificação de Uso (A); Mapa de uso do solo (B).



Fonte: Adaptado pelo autor com base em Musci *et al.* (2011).

O objetivo principal do trabalho não era maximizar a precisão da classificação, e sim comparar o desempenho associado a cada descritor. Com isso, os experimentos foram realizados com intuito de classificar as imagens de entrada puramente baseado em atributos texturais. Os aplicativos usados nos experimentos para o cálculo de LBP, LPQ e VAR (variância local) foram desenvolvidos pela *University of Oulu* (OULU, 2011). A implementação do SVM foi obtida de (CHANG; LIN, 2001). Já para o cálculo de atributos extraídos da GLCM, foram usadas funções disponíveis no MATLAB (MATHWORKS, 2009).

Serão descritos alguns experimentos realizados no trabalho, como por exemplo, a avaliação do impacto da informação de contraste, no qual foram avaliados quatro descritores de textura:

- a) apenas pelo histograma $LBP_{P,R}$,
- b) apenas pelo histograma $VAR_{P,R}$,
- c) pelo histograma resultante da concatenação dos histogramas individuais de $LBP_{P,R} + VAR_{P,R}$, e
- d) pelo histograma bidimensional que representa a distribuição conjunta textura $LBP_{P,R} / VAR_{P,R}$.

O experimento resultou na Tabela 2, onde são vistos os valores do índice Kappa para ambas as aplicações e áreas de estudo, gerando uma das conclusões de que os valores ótimos de P e R baseiam-se da aplicação alvo e o ajustes dos parâmetros possuem grande impacto sobre o desempenho.

Tabela 2: Resultados da classificação para descritores de textura mono-escalares formados por diferentes combinações de $LBP_{P,R}$ e $VAR_{P,R}$.

P,R	índice Kappa								#bins dos histogramas			
	Área de Estudo 1				Área de Estudo 2							
	LBP	VAR	$LBP+VAR$	LBP/VAR	LBP	VAR	$LBP+VAR$	LBP/VAR	LBP	VAR	$LBP+VAR$	LBP/VAR
8,1	0,69	0,81	0,81	0,83	0,78	0,72	0,83	0,83	10	8	18	80
8,2	0,76	0,79	0,90	0,89	0,79	0,77	0,85	0,86	10	8	18	80
8,3	0,79	0,77	0,88	0,90	0,65	0,82	0,86	0,86	10	8	18	80
16,2	0,81	0,80	0,84	0,86	0,78	0,77	0,91	0,87	18	8	26	144
16,3	0,81	0,77	0,87	0,88	0,78	0,84	0,86	0,86	18	8	26	144
24,3	0,81	0,76	0,88	0,86	0,76	0,77	0,82	0,83	26	8	34	208
24,5	0,78	0,68	0,86	0,85	0,53	0,77	0,74	0,77	26	8	34	208

Fonte: Adaptado pelo autor com base em Musci *et al.* (2011).

Já no segundo experimento, de avaliação de versões multi-escalares, o objetivo era de avaliar as configurações multi-escalares e comparar com as versões mono-escalares que as compõem, onde os resultados são aparentes na Tabela 3.

Tabela 3: Resultados da classificação para descritores de textura multi-escalares formados por diferentes combinações de $LBP_{P,R}$ e $VAR_{P,R}$.

P,R	índice Kappa								#bins dos histogramas			
	Área de Estudo 1				Área de Estudo 2							
	LBP	VAR	$LBP+VAR$	LBP/VAR	LBP	VAR	$LBP+VAR$	LBP/VAR	LBP	VAR	$LBP+VAR$	LBP/VAR
8,1+8,3	0,81	0,82	0,88	0,88	0,84	0,79	0,86	0,86	20	16	36	160
8,2+8,3	0,78	0,79	0,89	0,87	0,77	0,82	0,89	0,89	20	16	36	160
8,2+16,3	0,80	0,78	0,89	0,87	0,82	0,80	0,85	0,86	28	16	44	224
16,2+24,5	0,80	0,78	0,88	0,86	0,79	0,78	0,82	0,80	44	16	60	352
8,2+16,3+24,5	0,79	0,80	0,88	0,86	0,79	0,77	0,82	0,82	56	24	78	432
8,1+16,2+24,3	0,84	0,82	0,88	0,87	0,84	0,76	0,91	0,86	56	24	78	432
8,1+8,2+8,3	0,81	0,81	0,89	0,86	0,84	0,79	0,89	0,89	30	24	54	240

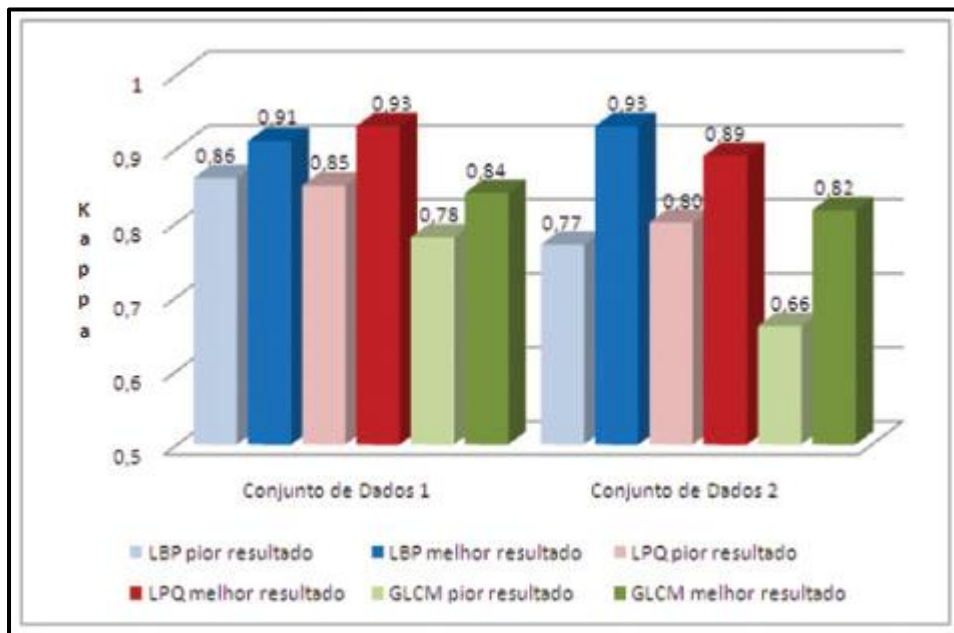
Fonte: Adaptado pelo autor com base em Musci *et al.* (2011).

Através da comparação das Tabelas 2 e 3, os autores concluíram que em cerca de 2/3 dos casos a versão multi-escalar entregou desempenho abaixo ou igual ao de ao menos um dos descritores mono-escalares que a compõem.

O trabalho inteiro contou com cinco experimentos no total, onde a Figura 8 mostra os melhores e piores resultados medidos para cada descritor.

Na Figura 10 é visível a superioridade dos resultados vindos do LBP e LPQ em comparação aos descritores GLCM, segundo os autores. A capacidade de discriminação de cada um dos descritores pode ser avaliada na comparação dos valores máximos do índice Kappa incluídos em cada caso.

Figura 10: Comparação de resultados da classificação com os descritores LBP, LPQ e GLCM das áreas de estudo.



Fonte: Adaptado pelo autor com base em Musci *et al.* (2011).

Segundo os autores, os experimentos realizados no trabalho asseguram que o poder de discriminação do LBP e LPQ apresenta uma significativa melhora quando combinados com a informação de contraste. Também vale frisar que não foi observado superioridade considerável das versões multi-escalares dos descritores LBP e LPQ sobre as versões mono-escalares. Concluindo, o presente trabalho também realizou comparações entre os descritores propostos com os descritores derivados da GLCM apresentados por Haralick, onde os experimentos apresentaram uma relevante superioridade dos descritores baseados em códigos binários locais sobre os descritores de Haralick.

3.3 Uso de descritores de textura extraídos de GLCM para o reconhecimento de padrões em diferentes domínios de aplicação

No trabalho de Martins *et al.* (2011), os autores tiveram o intuito de aplicar o uso de descritores de textura, extraídos via GLCM, em aplicações de reconhecimento de padrões em diferentes domínios: classificação de espécies florestais; classificação de gêneros musicais; e verificação de autoria de textos.

Nos experimentos foi utilizado o classificador *Support Vector Machine* (SVM). Foi necessária uma normalização de dados, onde os valores permanecem em uma escala que varia entre -1 e 1.

Dos três experimentos realizados, apenas um será descrito com base na sua relação com a análise textural. No experimento de classificação de espécies florestais, os autores afirmam que a maneira de simplificar a tarefa de classificar as espécies florestais se baseia na identificação de classes ou famílias taxonômicas. No contexto do experimento, podem ser tanto angiospermas como gimnospermas.

A base de dados utilizada no experimento é constituída por 600 imagens de 15 diferentes espécies florestais, fornecidas pelo Laboratório de Anatomia da Madeira da Universidade Federal do Paraná (UFPR). As espécies estão representadas na Tabela 4.

Tabela 4: Espécies que compõem a Base de Dados.

Gimnospermas	Angiospermas
<i>Araucaria Angustifolia</i>	<i>Dinizia excelsa</i>
<i>Pinus elliottii</i>	<i>Cedrela sp</i>
<i>Pinus maximinoi</i>	<i>Dalbergia nigra</i>
<i>Pinus taeda</i>	<i>Eucalyptus grandis</i>
<i>Podocarpus lambertii</i>	<i>Grevillea robusta</i>
	<i>Tabebuia sp</i>
	<i>Hymenaea sp</i>
	<i>Nectandra sp</i>
	<i>Porcelia macrocarpa</i>
	<i>Swietenia macrophylla</i>

Fonte: Adaptado pelo autor com base em Martins *et al.* (2011)

No desenvolvimento do experimento, os autores utilizaram seis descritores extraídos da GLCM: energia, contraste, entropia, homogeneidade, probabilidade máxima e momento de terceira ordem. Os testes que melhor apresentaram resultados foram os obtidos com distância $d = 1$ e ângulo $\theta = [0, 45, 90, 135]$, considerando que os seis descritores foram calculados para os quatros sentidos e concatenados, foi gerado um vetor contendo características com 24 itens.

Os resultados encontrados pelos autores são derivados das 24 características extraídas da GLCM aplicadas aos 3 classificadores. Os resultados indicam que todos os classificadores atingiram resultados semelhantes com uma pequena vantagem aproximada em 2% para SVM, como pode ser visto na Tabela 5.

Tabela 5: Taxa de reconhecimento por classificador.

Classificador	LDA	kNN($k = 4$)	SVM
Taxa de reconhecimento (%)	95.06	95.55	97.43
σ	0.18	0.87	0.59
Distância	4	5	4

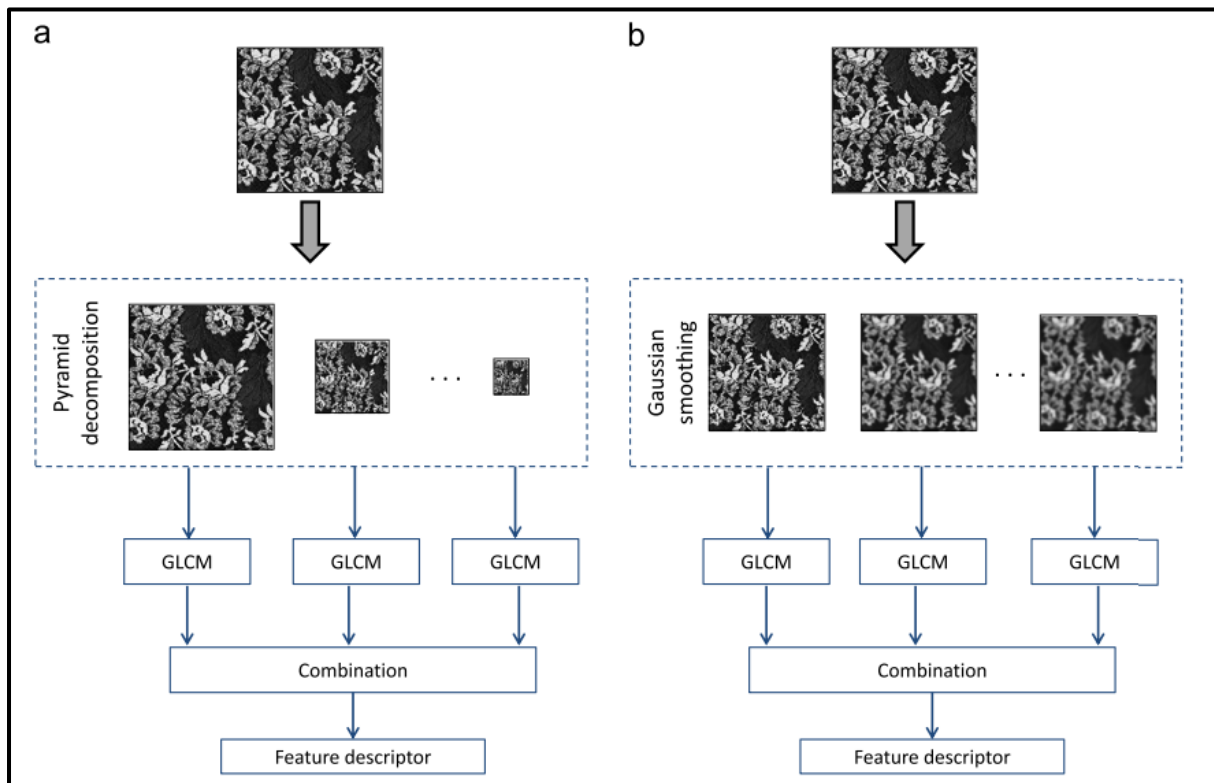
Fonte: Adaptado pelo autor com base em Martins *et al.* (2011)

A conclusão dos autores se dá na efetividade dos descritores empregados, sendo que os resultados foram altamente satisfatórios uma vez que foram utilizados em domínios de aplicação extremamente diversos entre si. Os resultados encontrados em reconhecimento de espécies florestais se mostraram animadores segundo os autores, com uma taxa de 97,43% ($\sigma = 0,59$), instigando que a investigação seja apropriada.

3.4 Multi-scale gray level co-occurrence matrices for texture description

O trabalho realizado por Siqueira *et al.* (2013) consiste em um novo esquema de extração de descritores da GLCM que é mais robusto sobre variação de escala. Dois representantes de multi-escala são usados na extensão do descritor. A performance da proposta do trabalho é avaliada através da aplicação de descritores de multi-escala sobre cinco bases de dados de textura de referência (Figura 12), sendo seus resultados comparados com outros descritores de textura. O esquema de abordagem de multi-escala utilizando GLCM pode ser visto na Figura 11:

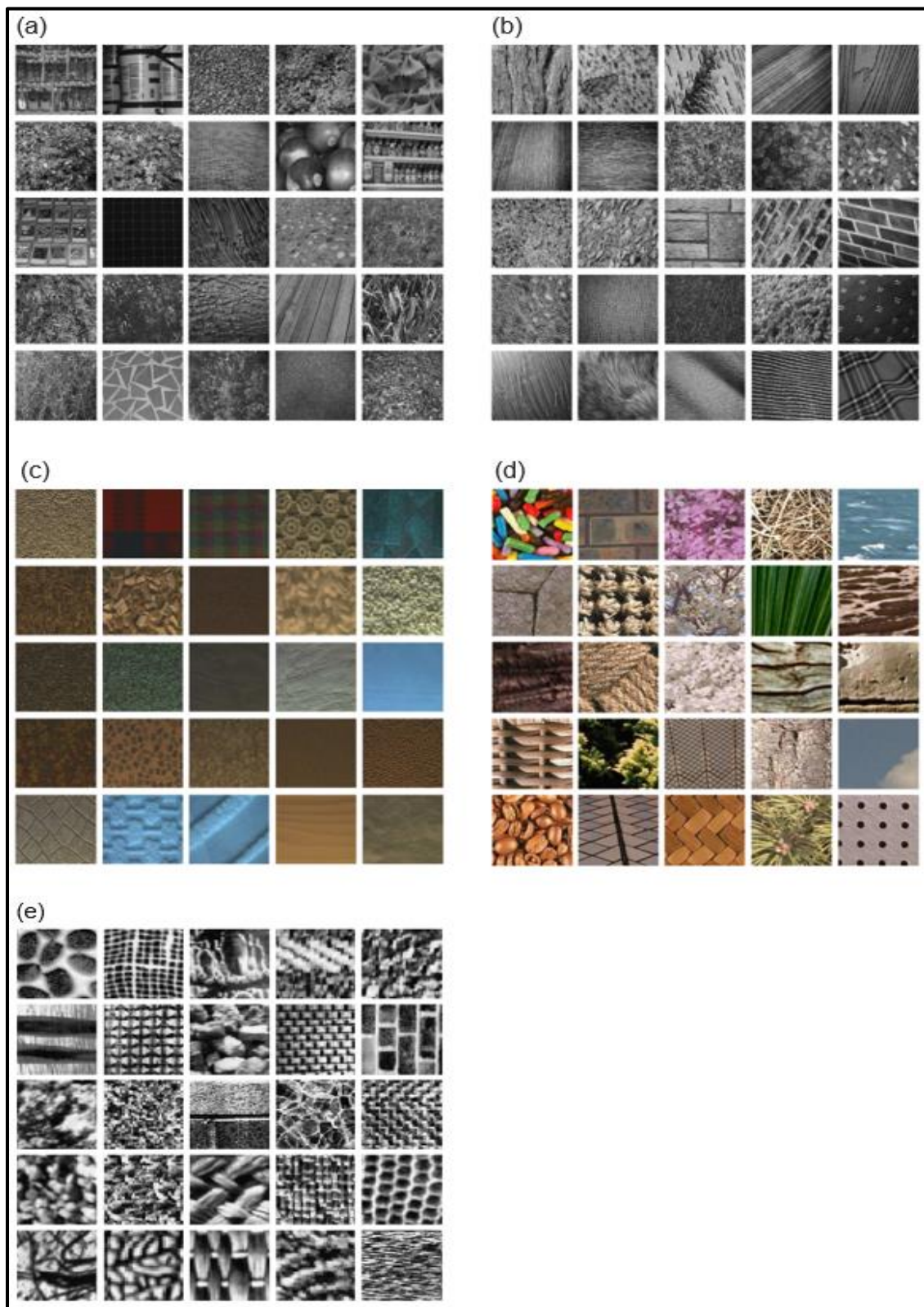
Figura 11: Abordagem de multi-escala em GLCM. (a) Decomposição em pirâmide;
(b) Suavização Gaussiana.



Fonte: Adaptado pelo autor com base em Siqueira *et al.* (2013).

Os dois métodos de multi-escala mais comuns são a suavização Gaussiana e decomposição em pirâmide. A suavização Gaussiana (*Gaussian Smoothing*) é a representação de uma escala espacial, onde uma sequência de imagens em diferentes níveis de escala espacial é montada sobre tamanhos variantes de *kernel*. Já na decomposição em pirâmide, a imagem é gradualmente reduzida em cada nível da pirâmide. Imagens em pirâmide podem ser geradas através da aplicação da suavização Gaussiana. Segundo os autores, o uso de várias escalas pode aprimorar o poder de distinção dos descritores de textura, visto que algumas informações estão presentes em apenas certas escalas.

Figura 12: Exemplificação das amostras de texturas utilizadas no trabalho extraídas dos respectivos conjuntos de dados. (a) extraídos por UMD; (b) extraídos por UIUC; (c) extraídos por OuTex; (d) extraídos por VisTex; (e) extraídos por Brodatz.



Fonte: Adaptado pelo autor com base em Siqueira *et al.* (2013)

De acordo com os autores, os resultados expressos nas Tabelas 6 e 7 descrevem que a escala de decomposição baseada em suavização Gaussiana concedeu uma taxa de classificação muito superior para todos os conjuntos de dados utilizados. A maior diferença está presente nos dados vindos do conjunto OuTex (84.61% com decomposição em pirâmide e 89.50% com suavização Gaussiana). Essa diferença se deve ao fato da pequena quantidade de amostras deste conjunto de dados associado à decomposição em pirâmide, resultado em matrizes esparsas.

Tabela 6: Taxas de classificação (%) adquiridas quando decomposição em pirâmide é considerada na GLCM e diversas abordagens para combinação de descritores são empregados.

Data set	Scales	Combination approach				
		Local	L+G	Corresponding	Weighted	Concatenation
UMD	2	90.84	88.27	92.81	91.67	93.03
	3	88.26	88.14	95.28	91.74	94.07
	4	89.88	89.88	94.65	91.77	94.52
UIUC	2	71.82	68.91	68.35	70.60	75.38
	3	72.18	72.18	71.93	70.99	79.02
	4	73.19	73.99	71.07	72.12	80.37
Brodatz	2	68.27	68.27	67.45	67.80	68.03
	3	65.77	65.77	64.10	66.72	61.81
	4	58.13	58.13	52.08	60.56	53.18
VisTex	2	90.28	90.28	86.11	93.98	94.68
	3	90.05	90.05	80.79	93.06	94.91
	4	87.04	87.04	68.98	93.75	92.59
OuTex	2	84.61	84.61	73.79	83.26	84.29
	3	82.28	82.28	70.94	82.22	82.44
	4	76.66	76.66	67.47	82.01	74.33

Fonte: Adaptado pelo autor com base em Siqueira *et al.* (2013).

Tabela 7: Taxas de classificação (%) adquiridas quando suavização Gaussiana é considerada na GLCM e diversas abordagens para combinação de descritores são empregados.

Data set	Scales	Combination approach				
		Local	L+G	Corresponding	Weighted	Concatenation
UMD	2	92.40	88.33	90.59	91.99	93.25
	3	88.25	88.25	93.89	92.29	94.28
	4	88.46	88.46	95.28	92.42	94.94
UIUC	2	80.08	80.08	81.88	81.74	81.02
	3	80.13	80.13	81.79	81.88	81.85
	4	81.88	81.88	81.74	81.88	81.79
Brodatz	2	80.82	80.82	86.37	73.50	81.82
	3	81.21	81.21	87.07	76.45	83.48
	4	81.29	81.29	87.19	77.71	83.05
VisTex	2	94.44	91.20	94.21	94.21	96.06
	3	91.90	91.90	93.98	93.52	95.14
	4	91.90	91.90	96.06	93.06	95.37
OuTex	2	88.71	88.71	84.98	84.56	88.30
	3	89.50	89.50	87.09	84.94	88.75
	4	89.33	89.31	87.36	84.65	88.38

Fonte: Adaptado pelo autor com base em Siqueira *et al.* (2013).

Tabela 8: Taxas de classificação (%) adquiridas por diversos descritores texturais em diferentes conjuntos de dados. A segunda coluna indica a quantidade de amostras utilizadas.

Data set	Samples	Feature extraction method					
		LBP	ILBP	Gabor	Granulometry	GLCM	
						Original	Multi-scale
UMD	5	72.5	73.3	75.7	75.4	83.1	85.2
	10	81.5	81.3	82.3	83.8	88.7	91.2
	15	85.4	85.0	84.9	88.0	91.1	93.5
	20	88.1	87.4	88.1	90.6	92.7	95.2
UIUC	5	41.7	44.8	51.0	48.3	58.9	65.1
	10	44.6	51.7	59.4	58.9	68.5	74.4
	15	49.2	54.7	65.1	64.5	72.7	78.9
	20	53.4	61.3	69.9	68.6	75.6	81.7
Brodatz	–	87.1	89.4	89.4	88.8	78.3	87.2
VisTex	–	96.5	97.9	92.6	94.4	94.4	96.0
OuTex	–	89.0	82.0	84.9	89.2	83.0	89.4

Fonte: Adaptado pelo autor com base em Siqueira *et al.* (2013).

Os resultados encontrados na Tabela 8 indicam uma melhoria significativa em todos os conjuntos de dados quando é aplicado a GLCM de multi-escala. Além disso, considerando os métodos já existentes na literatura, a proposta deste trabalho atingiu uma classificação melhor em três dos cinco conjuntos de dados testados. É importante ressaltar que a melhoria mais significativa ocorreu nos conjuntos de dados com a maior quantidade de amostras (UMD e UIUC).

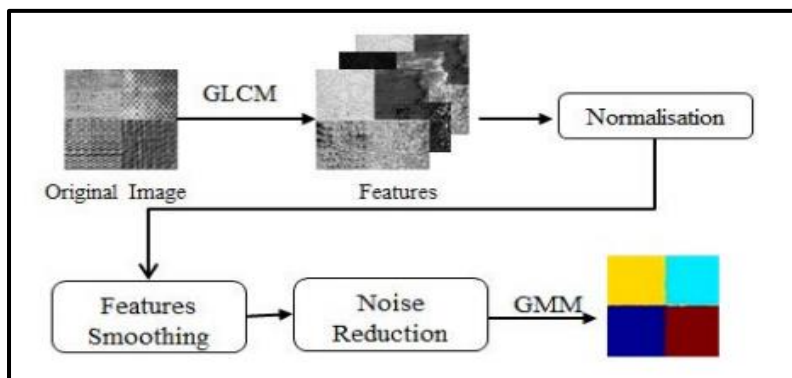
3.5 Texture Segmentation Using Different Orientations of GLCM Features

O trabalho realizado por Rampun *et al.* (2013) busca o desenvolvimento de um novo algoritmo baseado na segmentação de texturas a partir do GLCM. O método proposto pelos autores segmenta diferentes texturas baseado em técnicas de redução de ruído, as quais se mostram eficientes descritores de textura.

No respectivo trabalho, foi proposto que fosse realizado um processo adicional para cada recurso extraído antes de ser agrupado nas classificações, uma vez que os recursos extraídos individualmente da GLCM não são tão efetivos quando utilizados para discriminar diferentes texturas. O objetivo principal é testar diferentes técnicas de suavização antes que os recursos extraídos da GLCM sejam utilizados para agrupamento.

A metodologia utilizada naquele trabalho pode ser vista na Figura 13:

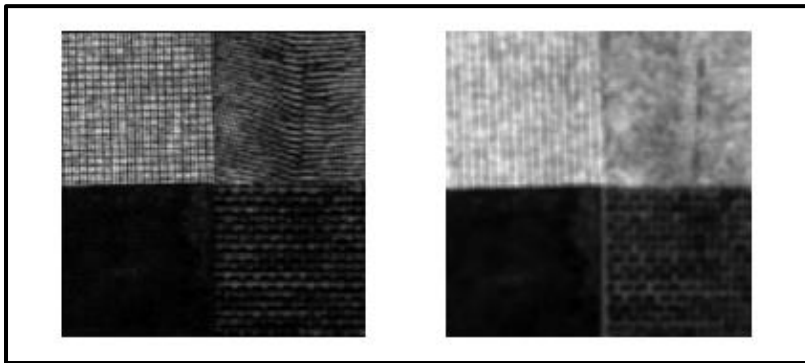
Figura 13: Visão geral da metodologia proposta. Consiste em seis etapas: extração, normalização, suavização, remoção de ruído, agrupamento e segmentação.



Fonte: Adaptado pelo autor com base em Rampun *et al.* (2013).

Para a extração de recursos foram selecionadas oito características de Haralick: contraste, correlação, energia, média da soma, variância da soma, entropia da soma, variância da diferença e entropia da diferença. Além disso, mais seis características sugeridas por Soh e Tsatsoulis (1999): proeminência de cluster, sombra de cluster, dissimilaridade, entropia, homogeneidade e probabilidade máxima. E por fim, mais duas características sugeridas por Clausi (2002): diferença inversa e momento de diferença inversa. Após a extração, os valores foram normalizados entre 0 e 1.

Figura 14: Extração original da GLCM (esquerda) e imagem processada pela equação (2) (direita).



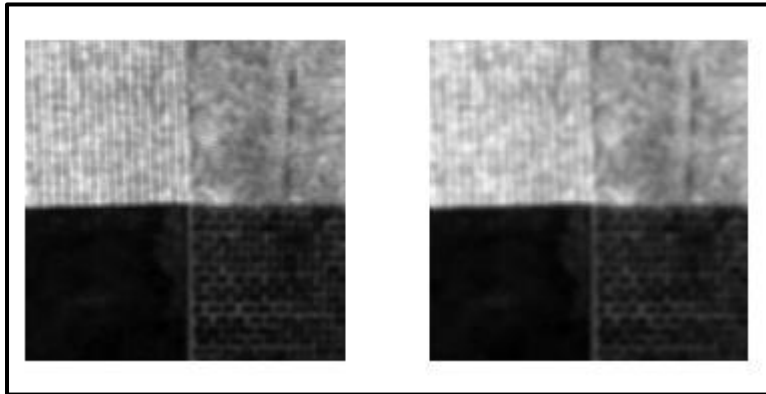
Fonte: Adaptado pelo autor com base em Rampun *et al.* (2013).

A Figura 14 exemplifica um método de suavização representado pela equação (2):

$$f_r(p, q) = \frac{\sqrt{2f_1^2} + \sqrt{2f_2^2}}{2} \quad (2)$$

A Figura 15 exemplifica a redução de ruído atingido após o processamento da imagem.

Figura 15: Exemplo de processamento de redução de ruído. Imagem pré-processamento (esquerda) e imagem pós-processamento (direita).

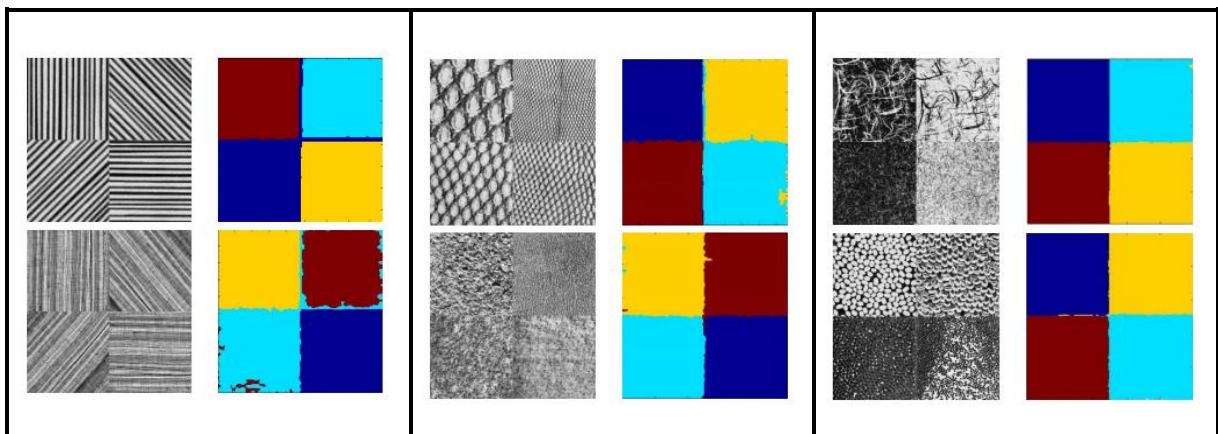


Fonte: Adaptado pelo autor com base em Rampun *et al.* (2013).

Nos experimentos realizados foram utilizadas 80 diferentes imagens com o tamanho de 256 x 256 pixels do álbum de Brodatz, cada imagem possui de duas a cinco texturas diferentes.

Os resultados dos testes de segmentação (Figura 16) sugerem que o algoritmo funciona corretamente independentemente a variação de orientação, e independente da variação de tamanho da textura, provando que o método sugerido é capaz de discriminar texturas em escalas diferentes. O método também se provou efetivo ao discriminar texturas com intensidades diferentes. O bom resultado foi obtido em regiões e fronteiras das imagens, com uma porcentagem de erro de aproximadamente 0.73% nos primeiros testes. O método de segmentação proposto pelos autores se mostrou menos eficiente em regiões com formatos diferentes.

Figura 16: Amostras de figuras antes (esquerda) e depois (direita) da segmentação.



Fonte: Adaptado pelo autor com base em Rampun *et al.* (2013).

Em conclusão, o método utiliza 16 diferentes recursos extraídos da GLCM para captação de informações texturais, cada extração foi realizada em duas orientações diferentes para maximizar a informação capturada. Após a normalização de valores, todos os recursos foram processados baseados na equação (2). Antes que a segmentação fosse feita, ocorreu a suavização de textura. O método foi testado em diversas texturas, tamanhos e formatos. Com uma porcentagem de aproximadamente de 3.65% de erro, o método demonstrou uma alta precisão em regiões e fronteiras de imagens. Os autores finalizam afirmando que o método é capaz de discriminar texturas independente de orientações, intensidade e tamanhos diferentes.

3.6 Comparativo com o trabalho proposto

É perceptível pelos trabalhos elucidados que existe uma grande variedade de técnicas de análise de textura, assim como suas vantagens para determinados campos de pesquisa na área de Visão Computacional. Este capítulo busca realizar um comparativo assertivo do presente trabalho com os trabalhos relacionados.

O trabalho realizado por Jucá e Mello (2011), diferindo deste, utilizou além da GLCM, outras técnicas de descrição textural como SDH e LBP. A metodologia utilizada pelo autor leva em consideração a análise de janelas “deslizantes”, onde imagens são divididas em sub imagens e cada parte dividida é tratada por um descritor textural, por fim sendo comparadas por um índice de fidelidade pré-definido. Em divergência com o atual trabalho, onde foram utilizadas imagens estáticas e a comparação dos resultados é realizada entre as amostras de imagens, não por um índice. Contudo, a semelhança entre os trabalhos está no modo de gerar os resultados, de maneira que foram utilizadas matrizes de dados para fazer a análise das imagens.

Já no trabalho feito por Musci *et al.* (2011), uma das diferenças encontradas é que a GLCM foi utilizada apenas como um índice comparativo, pois os descritores trabalhados foram LBP e LPQ. Outra diferença está na metodologia que busca comparar o desempenho de cada descritor, e não maximizar a precisão da

classificação. No entanto, o comparativo de resultados se assemelha no uso de matrizes e tabelas para a análise de resultados.

No trabalho elaborado por Martins *et al.* (2011), assim como o trabalho de conclusão que está sendo produzido, também foi feita uma tentativa de testar e validar o uso da técnica GLCM através do uso de descritores texturais, porém, o trabalho abrangeu três diferentes domínios de aplicação, colocando a prova a extensão de possibilidades do uso da técnica. As configurações da GLCM utilizadas por Martins *et al.* (2011) serviram como base para os inícios dos testes deste trabalho.

A obra escrita por Siqueira *et al.* (2013) buscou uma nova maneira de extrair informações da GLCM, utilizando representantes de multi-escala, se diferenciando desde a maneira de aplicar a GLCM até na questão de avaliação de resultados. O trabalho dividiu o desenvolvimento e comparação em dois testes isolados, e por fim os comparou entre si, sendo um dos testes baseado em Decomposição em pirâmide e outro em Suavização Gaussiana. O trabalho de Siqueira *et al.* (2013) serviu de inspiração na pesquisa literária de técnicas adversas e compatíveis à GLCM.

O trabalho escrito por Rampun *et al.* (2013) foi utilizado como referência de metodologia para o presente trabalho, usufruindo de três etapas das seis totais utilizadas pelo autor, que seriam, extração, normalização, suavização, remoção de ruído, agrupamento e segmentação, de modo que no atual trabalho foram utilizadas a extração, processamento de imagem e normalização. A Obra de Rampun *et al.* (2013) segmentou o processamento de imagem com intuito de expandir as possibilidades da remoção de ruído.

De maneira geral, todos os trabalhos deixaram heranças para serem utilizadas em pesquisas futuras, de maneira que foram aproveitadas neste trabalho de conclusão. A metodologia de Rampun *et al.* (2013), a forma avaliativa de Siqueira *et al.* (2013), as propriedades da GLCM providas por Martins *et al.* (2011), os objetivos de Musci *et al.* (2011) e as referências tecnológicas e teóricas absorvidas de Jucá e Mello (2011) resultaram no desenvolvimento deste trabalho.

4 MATERIAIS E MÉTODOS

Este capítulo apresenta as tecnologias utilizadas no desenvolvimento do trabalho, assim como os métodos empregados em sua elaboração.

Considerando o objetivo do trabalho de tornar mais explícitas as dificuldades da análise textural e o aprimoramento das técnicas, a metodologia de pesquisa do trabalho se caracteriza como exploratória que, segundo Gil (2002), permite elucidar o problema e construir hipóteses, nas quais se tem como principal objetivo o aprimoramento de ideias ou até a descoberta de intuições.

Levando em consideração o foco na realização de um experimento, a pesquisa também pode ser considerada experimental, pois, de acordo com Gil (2002), consiste em determinar um objeto de estudo, selecionar as variáveis de maior influência no objeto e definir formas de controle e observação dos efeitos que a variável produz sobre o objeto.

A pesquisa realizada no trabalho pode ser considerada como quantitativa. Pois, para Prodanov e Freitas (2013), essa abordagem considera tudo que pode ser quantificável, de maneira a transformar em números as demais opiniões e informações para analisá-las e classificá-las. A pesquisa quantitativa usa recursos e técnicas estatísticas para expressar os resultados, como porcentagem e média. A abordagem quantitativa se enquadra à essa pesquisa pelo fato de utilizar métricas

comparativas sobre resultados coletados por uma ferramenta, expressando os resultados em porcentagens de sucesso e falha.

4.1 Tecnologias

Diversas ferramentas e tecnologias serão utilizadas para o desenvolvimento deste trabalho. As principais delas serão descritas nas próximas seções.

4.1.1 Python

Segundo Borges (2014), Python é considerada uma linguagem de alto nível orientada a objetos, de tipagem dinâmica e forte, interativa e interpretada. A linguagem possui uma sintaxe clara e sucinta que beneficia a legibilidade do código-fonte, favorecendo a produtividade da linguagem.

A linguagem conta com diversas estruturas de alto nível já inclusas, como listas, data/hora, dicionários e números complexos. *Frameworks* de terceiros também podem ser incorporados. Recursos encontrados em outras linguagens como persistência, geradores, introspecção, unidades de teste e metaclasses também se encontram no Python. A interpretação da linguagem é feita por meio de *bytecode* na máquina virtual Python. Devido a isso, torna-se possível a compilação de aplicações em uma plataforma e rodar em outros sistemas, ou até mesmo executar diretamente o código-fonte (BORGES, 2014).

Borges (2014) explica que em Python existem blocos de códigos que são demarcados pelo uso de indentação. São consideradas boas práticas para manter e melhorar a consistência de um projeto, evitando que ocorra mistura de espaços e tabulações.

4.1.2 Jupyter

Segundo Lima (2017), o Jupyter Notebook é um ambiente que permite a edição e execução de documentos através de um *browser*. O Jupyter não requer acesso à Internet, pois ele exibe um painel de controle que mostra os locais de arquivos, possibilitando a execução local e interação via navegadores *web*.

4.1.3 Scikit-Image

Segundo Walt (2014), *Scikit-Image* é uma biblioteca de processamento de imagens que implementa algoritmos e utilitários para uso em pesquisas, educação e aplicações comerciais. Implementado na linguagem de programação Python por uma comunidade ativa de voluntários, é de código aberto e disponível através da licença *Open Source BSD*.

Esta biblioteca incorpora uma implementação da técnica GLCM, em que os cálculos de propriedades de textura são feitos através do módulo *Greycoprops* (SCIKIT-IMAGE DOCUMENTATION, 2020). As fórmulas utilizadas nos descritores podem ser vistas na Figura 17.

Figura 17: Características utilizadas neste trabalho para descrição de textura.

- 'contrast': $\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$
- 'dissimilarity': $\sum_{i,j=0}^{levels-1} P_{i,j}|i-j|$
- 'homogeneity': $\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$
- 'ASM': $\sum_{i,j=0}^{levels-1} P_{i,j}^2$
- 'energy': \sqrt{ASM}
- 'correlation':

$$\sum_{i,j=0}^{levels-1} P_{i,j} \left[\frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right]$$

Fonte: Scikit-Image Documentation (2020).

4.1.4 Matplotlib

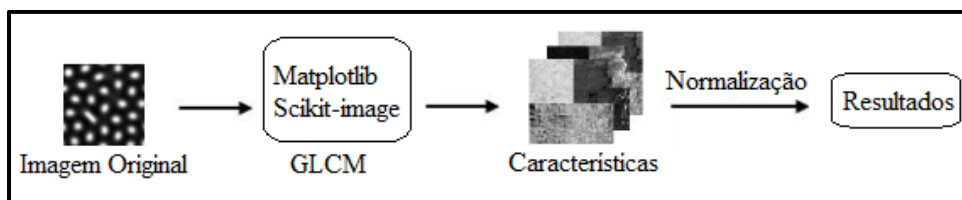
Matplotlib é uma biblioteca que auxilia a criação de gráficos, animações e interações visuais em Python. De forma prática, o Matplotlib representa os dados em figuras (MATPLOTLIB DOCUMENTATION, 2020).

4.2 Desenvolvimento

O desenvolvimento do trabalho tem continuidade através da sistematização de testes usando o GLCM utilizando as tecnologias mencionadas no começo no capítulo 4.

Citada no referencial teórico e nos trabalhos relacionados, é vasta a literatura sobre técnicas de análise de textura, a partir disto, é almejável a aplicação de diferentes técnicas e métricas sobre os descritores da GLCM. O algoritmo proposto utiliza a separação por painéis, característica do Jupyter, para organizar e sequenciar as operações, desde a importação e atribuição de bibliotecas, definição de amostras e métricas e por fim a apresentação de resultados.

Figura 18: Importação da biblioteca Matplotlib.



Fonte: Do autor (2020).

A metodologia almejada, que pode ser vista na Figura 18, se baseia na mesclagem de duas bibliotecas de algoritmos em Python para formar a GLCM, possibilitando extrair características do banco de imagens e processar tais características em resultados comparativos. As bibliotecas utilizadas, Matplotlib e Scikit-Image podem ser vistas sendo importadas na Figura 19.

Figura 19: Importação das bibliotecas utilizadas, Matplotlib e Scikit-Image.

```
import matplotlib.pyplot as plt
%matplotlib inline

import numpy as np

from skimage.feature import greycomatrix, greycoprops
from skimage.io import imread
```

Fonte: Do autor (2020).

O algoritmo modelo utilizado nos testes preliminares, Figura 20, foi usado como base para todas as futuras alterações das métricas utilizadas e amostras analisadas.

Figura 20: Modelo utilizado nos testes preliminares.

```
for category in ['04', '05', '08', '12']: # separados em 4 categorias
    for number in range(10):
        name = 'ratio-{}-{}.png'.format(category, number)

        a = imread(name, as_gray=True) # Leitura em tons de cinza [0.0, 1.1]
        b = (a - a.min()) / (a.max() - a.min()) * 255
        c = b.astype(int) # imagem [0, 255]

        glcm = greycomatrix(c, distances=[5], angles=[0], levels=256, symmetric=True, normed=True)

        co = greycoprops(glcm, 'contrast')[0, 0]
        cr = greycoprops(glcm, 'correlation')[0, 0]
        di = greycoprops(glcm, 'dissimilarity')[0, 0]
        en = greycoprops(glcm, 'energy')[0, 0]
        ho = greycoprops(glcm, 'homogeneity')[0, 0]

        print('{} co {:.2f} cr {:.2f} di {:.2f} en {:.4f} ho {:.4f}'.format(name, co, cr, di, en, ho))
```

Fonte: Do autor (2020).

Na Figura 21, pode ser visto como é feita a seleção das amostras das imagens. Separadas por quatro categorias, o objetivo é a comparação das imagens dentro das suas categorias até que seja feita uma relação aproximada de seus resultados.

Figura 21: Seleção das categorias das imagens a serem analisadas.

```
for category in ['04', '05', '08', '12']: # separados em 4 categorias
    for number in range(10):
        name = 'ratio-{}-{}.png'.format(category, number)
```

Fonte: Do autor (2020).

Figura 22: Seção da leitura das imagens em tons de cinza.

```
a = imread(name, as_gray=True) # leitura em tons de cinza [0.0, 1.1]
b = (a - a.min()) / (a.max() - a.min()) * 255
c = b.astype(int) # imagem [0, 255]
```

Fonte: Do autor (2020).

A seção do algoritmo que utiliza as bibliotecas para fazer a leitura das imagens em tons de cinza reconhecíveis para a GLCM está na Figura 22, onde ocorre a leitura e normalização dos valores provenientes das imagens entre 0.0 e 1.1.

A Figura 23 exhibe como é feita a escolha das propriedades a serem extraídas, como contraste, correlação, energia, homogeneidade e suas configurações, variáveis apontadas respectivamente pela GLCM: lista de deslocamentos da distância de par de pixels; lista de ângulos de par de pixels em radianos; número de níveis de cinza contados; definição se a matriz de saída será simétrica; definição se cada matriz será normalizada pela divisão do número total de ocorrências.

Figura 23: Seleção das características e métricas a serem extraídas da GLCM.

```
glcm = greycomatrix(c, distances=[5], angles=[0], levels=256, symmetric=True, normed=True)
co = greycoprops(glcm, 'contrast')[0, 0]
cr = greycoprops(glcm, 'correlation')[0, 0]
di = greycoprops(glcm, 'dissimilarity')[0, 0]
en = greycoprops(glcm, 'energy')[0, 0]
ho = greycoprops(glcm, 'homogeneity')[0, 0]
```

Fonte: Do autor (2020).

Os resultados obtidos são apresentados pela seção exibida na Figura 24, a partir da exibição podem ser comparadas e relacionados entre si e cada *ratio* de imagem.

Figura 24: Exibição dos resultados obtidos.

```
print('{} co {:.2f} cr {:.2f} di {:.2f} en {:.4f} ho {:.4f}'.format(name,co,cr,di,en,ho))
```

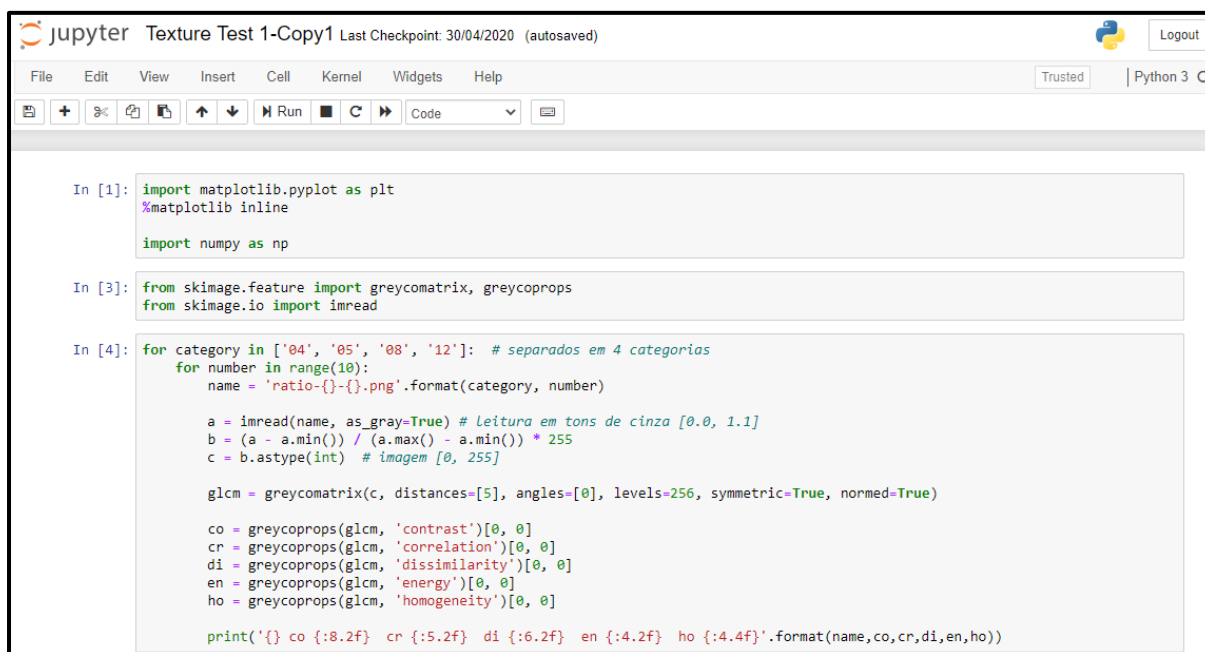
Fonte: Do autor (2020).

No capítulo a seguir serão detalhados os testes e resultados obtidos a partir da ferramenta desenvolvida, aplicando a metodologia de uso da GLCM sobre imagens digitais para extrair suas características em forma de dados, e a partir da normalização de dados, representar as possíveis relações existentes entre os *ratios* de imagem.

5 TESTES E RESULTADOS

Os testes iniciais ocorreram na plataforma Jupyter Notebook (Figura 25). O objetivo principal dos experimentos era encontrar um padrão nos valores obtidos pela aplicação da GLCM em cima das amostras de imagens, com o princípio de achar uma relação de discriminação entre os valores e as imagens.

Figura 25: Algoritmo utilizado para realização dos experimentos.

The image shows a Jupyter Notebook window titled "Texture Test 1-Copy1". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and output viewing. The code is written in Python and is organized into three input cells. The first cell imports matplotlib.pyplot as plt, sets matplotlib to inline, and imports numpy as np. The second cell imports skimage.feature.greycoprops, skimage.io.imread, and skimage.feature.greymatrix. The third cell is a loop that iterates over four categories ('04', '05', '08', '12') and ten numbers (0-9). For each iteration, it reads an image, normalizes it to grayscale, and calculates GLCM features: contrast, correlation, dissimilarity, energy, and homogeneity. The results are printed in a formatted string.

```
In [1]: import matplotlib.pyplot as plt
        %matplotlib inline
        import numpy as np

In [3]: from skimage.feature import greycoprops
        from skimage.io import imread

In [4]: for category in ['04', '05', '08', '12']: # separados em 4 categorias
        for number in range(10):
            name = 'ratio-{}-{}.png'.format(category, number)

            a = imread(name, as_gray=True) # leitura em tons de cinza [0.0, 1.1]
            b = (a - a.min()) / (a.max() - a.min()) * 255
            c = b.astype(int) # imagem [0, 255]

            glcm = greycoprops(c, distances=[5], angles=[0], levels=256, symmetric=True, normed=True)

            co = greycoprops(glcm, 'contrast')[0, 0]
            cr = greycoprops(glcm, 'correlation')[0, 0]
            di = greycoprops(glcm, 'dissimilarity')[0, 0]
            en = greycoprops(glcm, 'energy')[0, 0]
            ho = greycoprops(glcm, 'homogeneity')[0, 0]

            print('{} co {:.2f} cr {:.2f} di {:.2f} en {:.2f} ho {:.2f}'.format(name, co, cr, di, en, ho))
```

Fonte: Do autor (2020).

O método utiliza cinco descritores texturais extraídos da GLCM, sejam eles: contraste, correlação, dissimilaridade, energia e homogeneidade. Os conjuntos de amostra de imagens podem ser vistos na Figura 26.

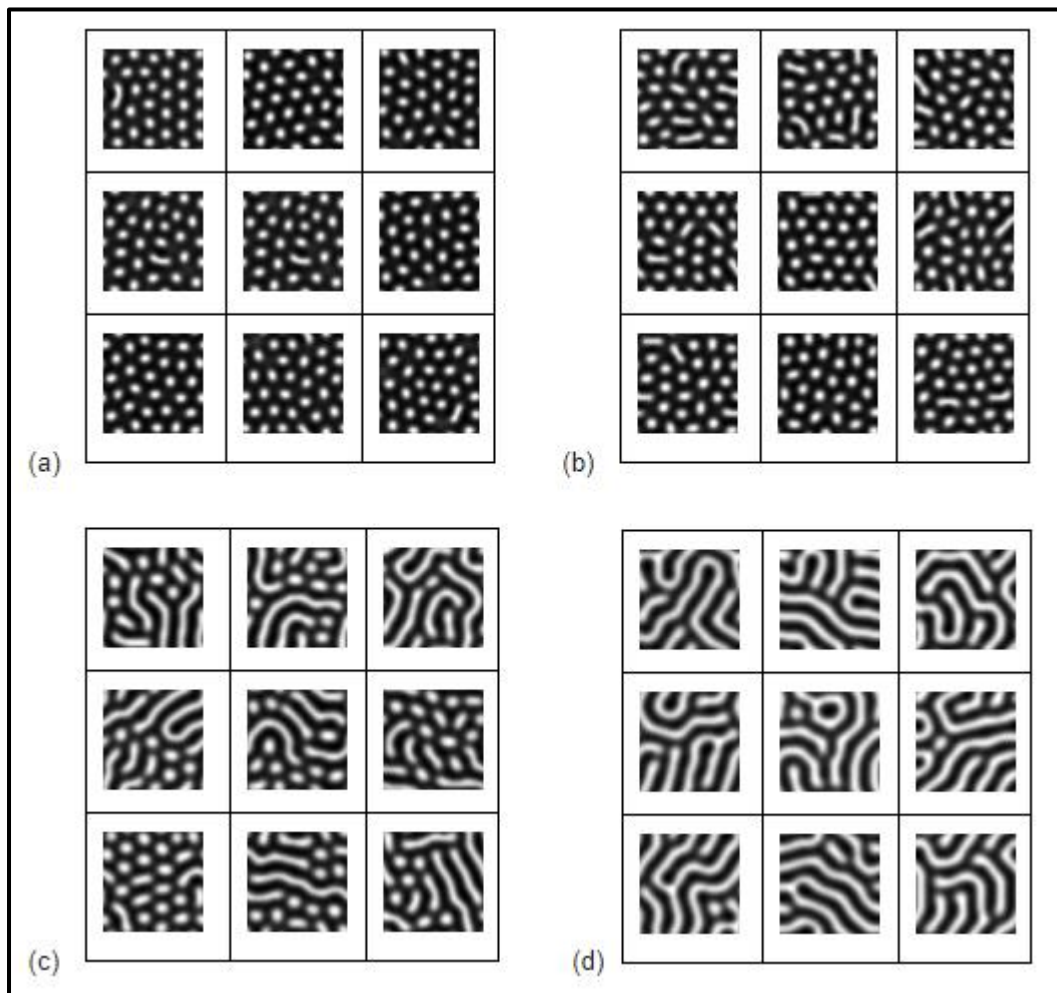
O conjunto de amostras de imagens utilizado nos experimentos se baseiam em imagens sintéticas criadas pelo processo de reação-difusão (FENSTERSEIFER, 2018). As imagens são classificadas em 4 tipos diferentes, com base no parâmetro *ratio* de geração delas.

Duas importações de bibliotecas são feitas para funcionamento do algoritmo, Matplotlib e Scikit-Image. As importações acrescentam ao algoritmo ferramentas de tratamento de imagem e propriedades matemáticas.

O algoritmo pode ser dividido em diversas etapas:

1. Importação das bibliotecas;
2. Seleção das categorias de imagens a serem utilizadas;
3. Seleção da quantidade de amostras a serem utilizadas;
4. Função de tratamento de imagem para leitura em tons de cinza;
5. Configuração da GLCM (conjunto de imagens; lista de deslocamentos da distância de par de pixels; lista de ângulos de par de pixels em radianos; número de níveis de cinza contados; definição se a matriz de saída será simétrica; definição se cada matriz será normalizada pela divisão do número total de ocorrências);
6. Definição dos descritores a serem utilizados;
7. Impressão dos resultados obtidos;

Figura 26: Conjunto de amostras de imagens sintéticas classificadas por *ratio*. (a) *Ratio* 4; (b) *Ratio* 5; (c) *Ratio* 8; (d) *Ratio* 12.



Fonte: Do autor (2020).

É possível ver alguns padrões nos resultados listados na Figura 27. Um dos padrões percebidos, é que imagens de *ratio* 12 possuem um valor de homogeneidade normalmente abaixo de 0,02, com apenas algumas breves exceções. Outro padrão observado indica que imagens de *ratio* 4 possuem valores abaixo de 80 em sua dissimilaridade.

Figura 27: Resultados encontrados a partir do primeiro conjunto de amostras.

ratio-04-0.png	co	11381.78	cr	-0.40	di	79.44	en	0.0205	ho	0.0292
ratio-04-1.png	co	9776.79	cr	-0.31	di	71.76	en	0.0198	ho	0.0353
ratio-04-2.png	co	10823.36	cr	-0.33	di	77.26	en	0.0198	ho	0.0320
ratio-04-3.png	co	11406.74	cr	-0.42	di	79.65	en	0.0200	ho	0.0331
ratio-04-4.png	co	11649.63	cr	-0.41	di	79.86	en	0.0195	ho	0.0293
ratio-04-5.png	co	10592.85	cr	-0.30	di	74.67	en	0.0199	ho	0.0379
ratio-04-6.png	co	10798.35	cr	-0.35	di	76.25	en	0.0206	ho	0.0352
ratio-04-7.png	co	11129.63	cr	-0.39	di	78.45	en	0.0199	ho	0.0294
ratio-04-8.png	co	11316.07	cr	-0.37	di	78.50	en	0.0197	ho	0.0228
ratio-04-9.png	co	11098.68	cr	-0.36	di	78.00	en	0.0194	ho	0.0300
ratio-05-0.png	co	11022.35	cr	-0.35	di	80.29	en	0.0202	ho	0.0356
ratio-05-1.png	co	11061.13	cr	-0.34	di	79.43	en	0.0189	ho	0.0288
ratio-05-2.png	co	11409.45	cr	-0.38	di	80.82	en	0.0190	ho	0.0273
ratio-05-3.png	co	11569.60	cr	-0.35	di	82.38	en	0.0188	ho	0.0237
ratio-05-4.png	co	11323.02	cr	-0.40	di	81.16	en	0.0197	ho	0.0240
ratio-05-5.png	co	11289.42	cr	-0.33	di	80.09	en	0.0194	ho	0.0304
ratio-05-6.png	co	11160.19	cr	-0.34	di	80.29	en	0.0193	ho	0.0271
ratio-05-7.png	co	11696.56	cr	-0.42	di	83.11	en	0.0190	ho	0.0273
ratio-05-8.png	co	11400.86	cr	-0.32	di	80.46	en	0.0197	ho	0.0292
ratio-05-9.png	co	11335.72	cr	-0.39	di	80.31	en	0.0205	ho	0.0281
ratio-08-0.png	co	9912.35	cr	-0.06	di	79.07	en	0.0174	ho	0.0209
ratio-08-1.png	co	12204.08	cr	-0.33	di	88.94	en	0.0172	ho	0.0231
ratio-08-2.png	co	9554.39	cr	0.03	di	75.03	en	0.0174	ho	0.0266
ratio-08-3.png	co	12203.61	cr	-0.28	di	88.89	en	0.0172	ho	0.0172
ratio-08-4.png	co	12102.60	cr	-0.24	di	89.56	en	0.0170	ho	0.0193
ratio-08-5.png	co	9136.05	cr	0.04	di	74.15	en	0.0174	ho	0.0227
ratio-08-6.png	co	12508.81	cr	-0.30	di	92.64	en	0.0168	ho	0.0153
ratio-08-7.png	co	19754.55	cr	-0.73	di	121.08	en	0.0194	ho	0.0106
ratio-08-8.png	co	12242.96	cr	-0.30	di	88.57	en	0.0174	ho	0.0220
ratio-08-9.png	co	13864.74	cr	-0.33	di	96.24	en	0.0169	ho	0.0145
ratio-12-0.png	co	10882.76	cr	0.05	di	84.69	en	0.0166	ho	0.0152
ratio-12-1.png	co	11616.60	cr	-0.12	di	87.93	en	0.0170	ho	0.0176
ratio-12-2.png	co	9103.00	cr	0.09	di	76.23	en	0.0168	ho	0.0177
ratio-12-3.png	co	13856.08	cr	-0.34	di	98.45	en	0.0177	ho	0.0122
ratio-12-4.png	co	10643.15	cr	0.05	di	79.88	en	0.0187	ho	0.0406
ratio-12-5.png	co	7098.99	cr	0.30	di	64.64	en	0.0173	ho	0.0280
ratio-12-6.png	co	14692.03	cr	-0.48	di	102.50	en	0.0170	ho	0.0132
ratio-12-7.png	co	18806.56	cr	-0.75	di	119.98	en	0.0199	ho	0.0084
ratio-12-8.png	co	14544.69	cr	-0.32	di	101.01	en	0.0169	ho	0.0112
ratio-12-9.png	co	15354.58	cr	-0.40	di	104.02	en	0.0180	ho	0.0115

Fonte: Do autor (2020).

Essa foi uma análise preliminar, uma vez que se almeja sistematizar o processo de discriminação de texturas. As inconsistências precisam ser trabalhadas para minimizar as variações e facilitar a análise.

5.1 Rodadas de teste

Com a obtenção de um novo conjunto de métricas devido a aplicação de novas configurações, é fundamental uma comparação entre as estatísticas iniciais e as novas. Com isso, torna-se necessário uma seleção das métricas mais significativas para classificação automática dos conjuntos de imagens. Também serão realizados testes adicionais com outras classes de texturas, para verificar se os critérios de classificação são mais amplos. Uma exemplificação dos resultados a serem obtidos pode ser visto na Figura 28.

Figura 28: Exemplificação dos dados extraídos pela GLCM.

ratio-05-95.png	co	11525.04	cr	-0.4018	di	82.41	en	0.0201	ho	0.0210
ratio-05-96.png	co	12577.22	cr	-0.4566	di	86.98	en	0.0200	ho	0.0204
ratio-05-97.png	co	10856.27	cr	-0.3743	di	79.72	en	0.0195	ho	0.0216
ratio-05-98.png	co	11565.14	cr	-0.2858	di	81.88	en	0.0188	ho	0.0241
ratio-05-99.png	co	10922.49	cr	-0.3210	di	79.38	en	0.0200	ho	0.0285
ratio-08-0.png	co	13595.16	cr	-0.4653	di	96.28	en	0.0175	ho	0.0116
ratio-08-1.png	co	11540.64	cr	-0.2663	di	86.43	en	0.0176	ho	0.0178
ratio-08-2.png	co	12940.69	cr	-0.3102	di	91.86	en	0.0174	ho	0.0174
ratio-08-3.png	co	12837.76	cr	-0.3459	di	92.00	en	0.0175	ho	0.0179
ratio-08-4.png	co	12629.42	cr	-0.2974	di	91.47	en	0.0172	ho	0.0176

Fonte: Do autor (2020).

Os testes foram avaliados conforme Siqueira *et al.* (2013), de maneira onde seus resultados são comparados com testes onde suas propriedades e métricas diferem, almejando encontrar a melhor configuração de extração da GLCM.

A seleção das características contraste, correlação, dissimilaridade e energia deve-se a análise preliminar decorrida dos testes iniciais, onde foi observado uma resposta mais eficiente de tais características. As demais características que podem ser extraídas pela GLCM, como entropia e sombra, foram deixadas de lado, pois não apresentaram uma reação satisfatória em relação as alterações testadas nas configurações da ferramenta.

Os testes preliminares resultaram em um comparativo de igualdade aceitável entre imagens de mesmo *ratio*, mas foi insuficiente na diferenciação de imagens de *ratios* diferentes. As propriedades usadas nos testes iniciais foram contraste,

correlação, dissimilaridade, energia e homogeneidade, e suas métricas foram distância (5), ângulo (0), nível (256), simetria (Verdadeiro) e normalizado (Verdadeiro).

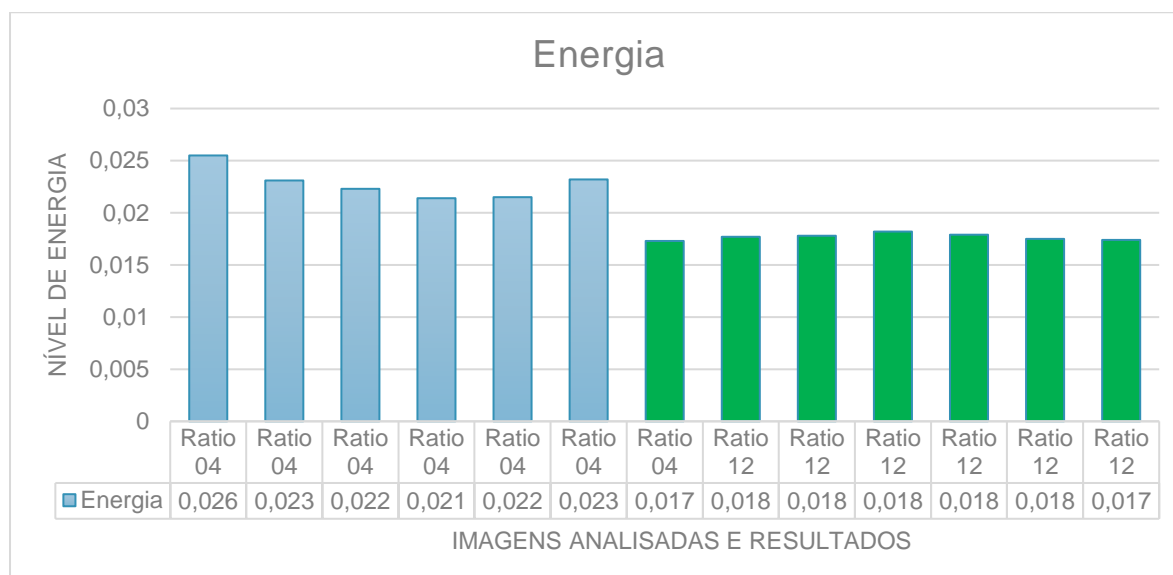
Durante as realizações dos testes iniciais foi realizada uma análise da reação dos resultados em relação as alterações das métricas que eram aplicadas sobre as imagens, desenvolvendo assim um parâmetro de possíveis combinações que seriam suscetíveis aos testes.

As rodadas de testes foram conduzidas a partir de alterações e experimentos sobre o modelo inicial, de maneira que cada rodada de teste se refere a um experimento com propriedades e métricas variantes às anteriores. Nas seguintes rodadas de testes, os resultados em porcentagem foram obtidos a partir de 100 amostras de cada *ratio*.

5.1.1 Rodada de teste 1

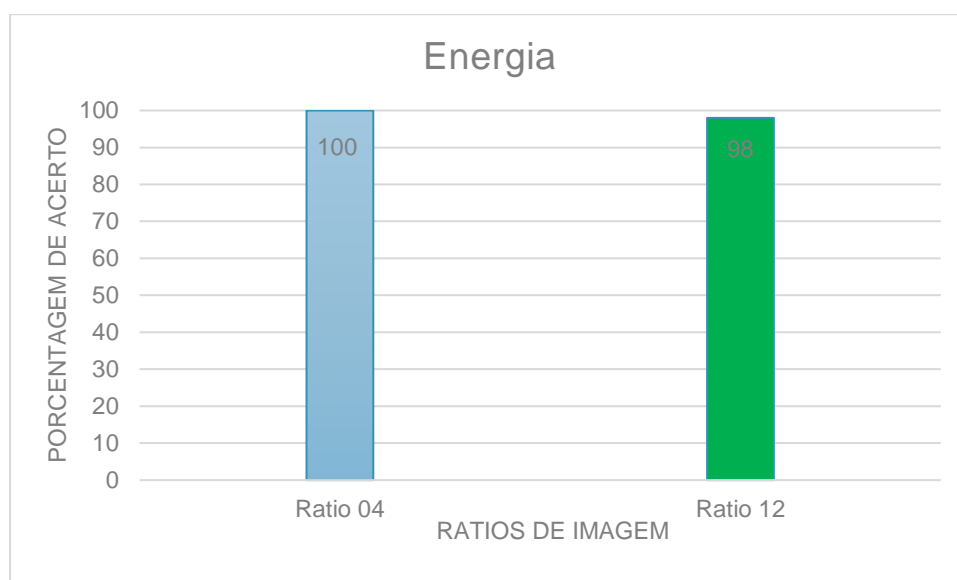
A primeira rodada de teste pretendia encontrar uma diferença nítida entre os dois *ratios* de imagem mais distintos, que seriam 04 e 12. Entre as alterações realizadas para encontrar essa distinção, foram testadas diversas possibilidades de configurações, mas a alteração mais satisfatória foi na distância da análise dos deslocamentos de pares de pixels, onde seu valor foi ajustado de 5 para 10. No Gráfico 1 é possível visualizar a diferença existente entre amostras de *ratio* 04 e 12, onde é observado uma separação dos dois tipos de amostras a partir do valor do nível de energia de 0,02, na qual imagens de *ratio* 04 possuem valores superiores à 0,02 enquanto imagens de *ratio* 12 possuem valores inferiores à 0,02.

Gráfico 1: Representação dos resultados da primeira rodada de teste.



Fonte: Do autor (2020).

Gráfico 2: Resultado em porcentagem da primeira rodada de teste.



Fonte: Do autor (2020).

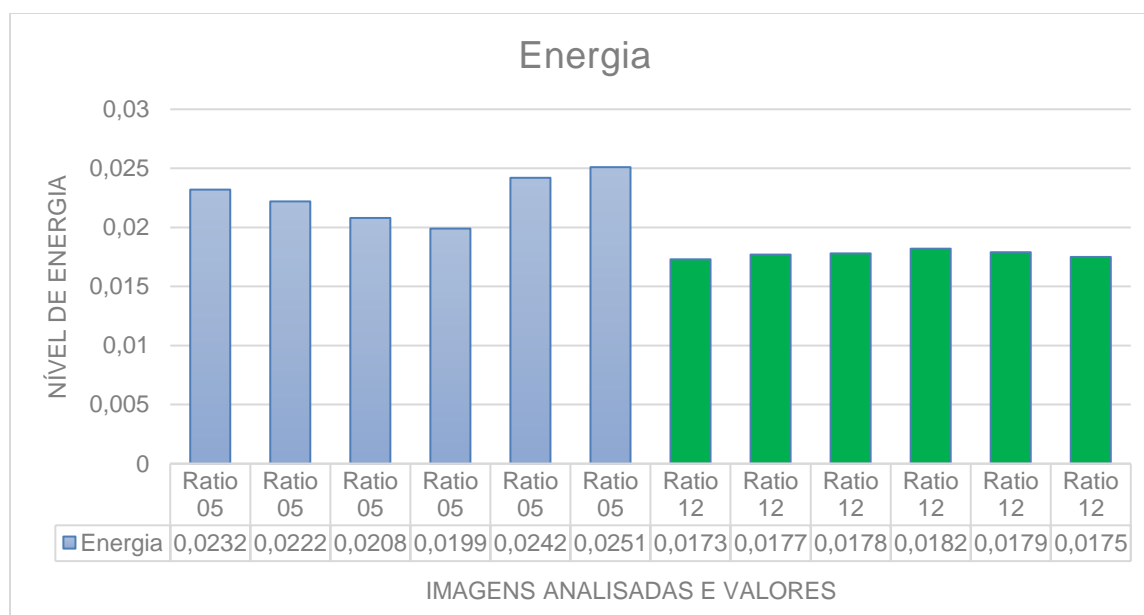
A primeira rodada de testes permitiu a visualização da característica mais propícia à exploração, o nível de energia. Porém esse resultado extremamente satisfatório, manifestado no Gráfico 2, é visto na comparação entre os *ratios* 04 e 12 devido às estruturas contidas nas imagens, na qual as imagens de *ratio* 04 consistem em pontos isolados e as imagens de *ratio* 12 são constituídas de estruturas alongadas e contínuas.

A partir dos resultados obtidos na primeira rodada, a GLCM se mostrou promissora devido ao seu comportamento favorável a alterações de parâmetros, com uma porcentagem de 100% de acerto na análise de imagens de *ratio* 04 e 98% de acerto para as imagens de *ratio* 12. A característica nível de energia encontrada nesta rodada foi usada como referência nos testes subsequentes.

5.1.2 Rodada de teste 2

Com embasamento na primeira rodada de teste, foi realizado um comparativo das imagens de *ratio* 05 e 12, detalhado no Gráfico 3, utilizando as mesmas configurações, foi possível observar que os valores de energia possuíam a mesma relação, imagens de *ratio* 05 possuem valores próximos ou superiores à 0,02 enquanto imagens de *ratio* 12 possuem valores próximos ou inferiores à 0,02.

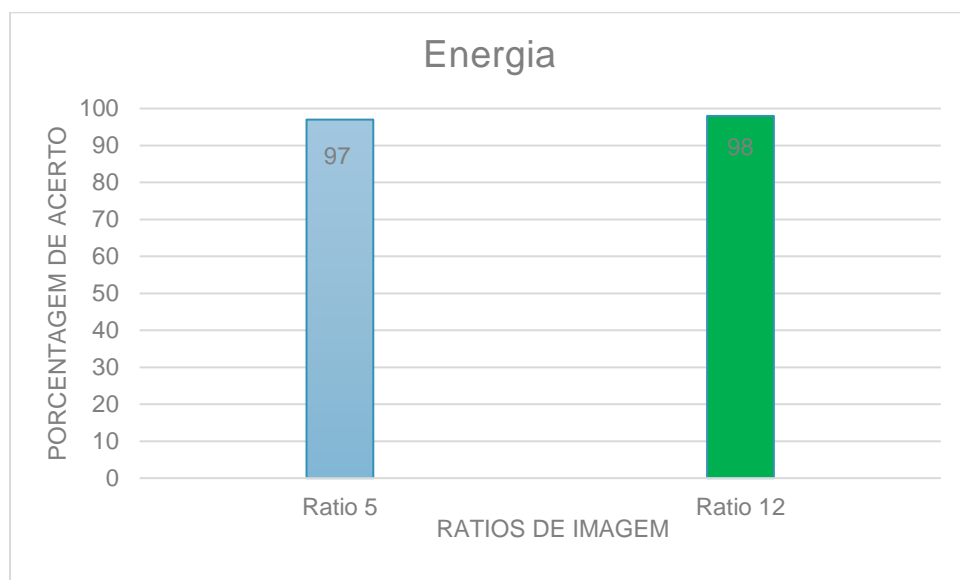
Gráfico 3: Representação dos resultados da segunda rodada de teste.



Fonte: Do autor (2020).

O resultado observado no Gráfico 4 confirma a existência da disparidade dos valores de energia entre os *ratios* trabalhos na primeira e segunda rodada de testes. Com uma porcentagem de acerto de 97% e 98% para imagens de *ratio* 05 e 12 respectivamente, o teste se mostrou extremamente satisfatório ao diferenciar com consistência imagens distintas.

Gráfico 4: Resultado em porcentagem da segunda rodada de teste.

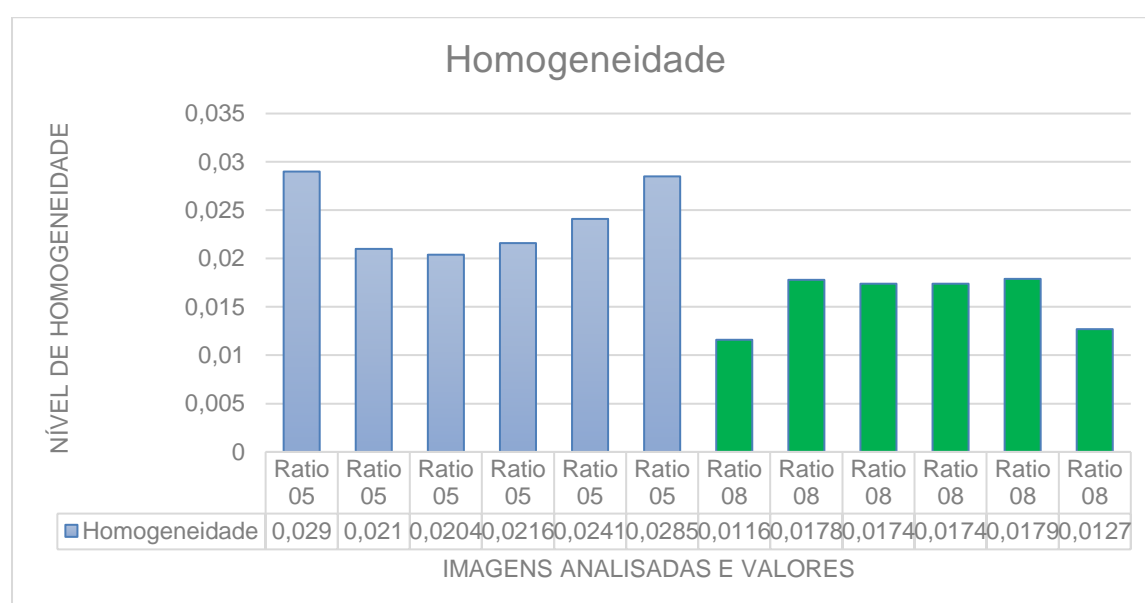


Fonte: Do autor (2020).

5.1.3 Rodada de teste 3

A terceira rodada de teste objetivou encontrar uma métrica eficaz para comparar dois *ratios* similares, que seriam 05 e 08. Durante essa rodada de teste foram aplicadas diferentes alterações, isoladas e em conjunto, e diferenciando-se dos testes antecessores, a alteração de melhor rendimento foi sobre a característica de homogeneidade. Com a alteração da angulação dos pares de pixels de 0 para 180, foi possível obter uma relação sólida, onde os valores de homogeneidade de imagens de *ratio* 05 se mantiveram acima de 0,02 e as imagens de *ratio* 08 abaixo de 0,02. O Gráfico 5 expressa uma representação da diferença existente entre as amostras de imagens.

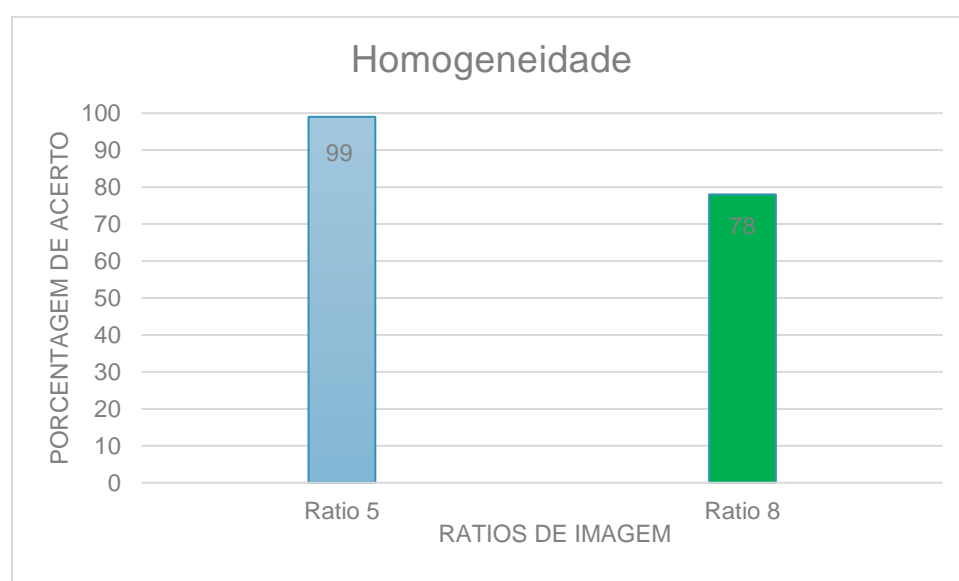
Gráfico 5: Representação dos resultados da terceira rodada de teste.



Fonte: Do autor (2020).

O resultado que pode ser observado no Gráfico 6, demonstra uma queda na taxa de acerto nas comparações, com uma porcentagem de acerto de 99% e 78% para imagens de *ratio* 05 e 08 respectivamente, o teste se demonstrou satisfatório, mas evidencia deficiências ao comparar imagens similares.

Gráfico 6: Resultado em porcentagem da terceira rodada de teste.



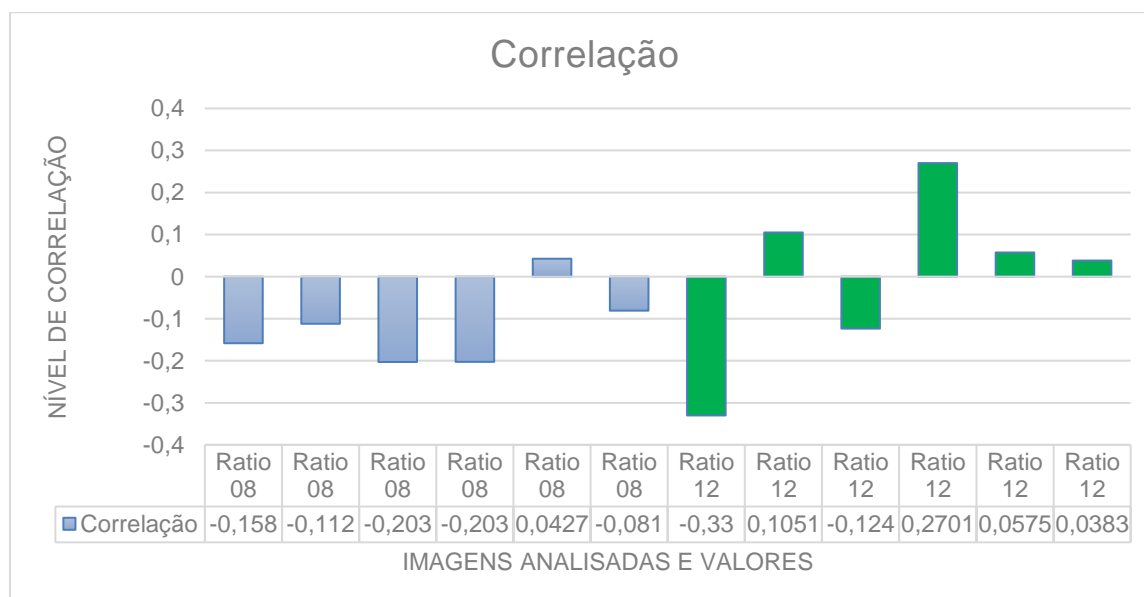
Fonte: Do autor (2020).

5.1.4 Rodada de teste 4

Já a quarta rodada de teste buscava comparar os dois *ratios* mais complexos em questão de estrutura, tanto as imagens de *ratio* 08 quanto as de *ratio* 12 possuem estruturas isoladas e contínuas, de maneira a serem o maior desafio das análises. Essa rodada conta com a maior quantidade de alterações realizadas, sendo o maior teste deste trabalho, sendo necessário alterações em todas as propriedades de extração, a configuração com melhor resultado seria distância (15), ângulo (180), nível (256), simetria (Falso) e normalizado (Falso).

A característica correlação foi que apresentou os melhores resultados, porém não satisfatórios quando comparada com os demais testes. O motivo das dificuldades encontradas está na similaridade das imagens, que seria a sua inconsistência de padrões, dificultando a extração e gerando um resultado variante além do estimado.

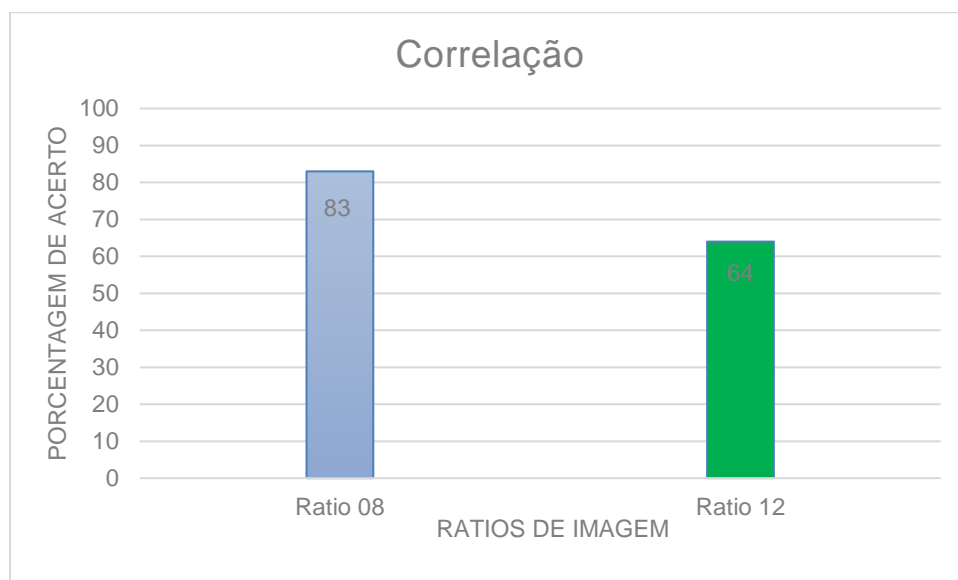
Gráfico 7: Representação dos resultados da quarta rodada de teste.



Fonte: Do autor (2020).

Infelizmente, o comportamento da GLCM sobre imagens inconsistentes teve uma efetividade abaixo do esperado. Conforme é possível visualizar no Gráfico 7, as imagens apresentavam, mas não com a uniformidade visualizada em outros testes.

Gráfico 8: Resultado em porcentagem da quarta rodada de teste.



Fonte: Do autor (2020).

O resultado exposto no Gráfico 8, demonstra uma queda ainda maior na taxa de acerto nas comparações em relação aos testes anteriores, com uma porcentagem de acerto de 83% e 64% para imagens de *ratio* 08 e 12. É importante ressaltar que a quarta rodada de teste comparou imagens praticamente semelhantes em questão de estrutura e os resultados encontrados se enquadram dentro do esperado.

5.2 Síntese dos testes

A partir dos testes realizados confirmou-se a efetividade da GLCM na classificação de imagens sintéticas a partir de tons de cinza. Como observado na primeira e segunda rodada de teste, a comparação de imagens que possuem estruturas distintas se mostrou extremamente satisfatória, expressando resultados próximos a 98%% de acerto. Já a comparação de imagens com estruturas complexas e irregulares obteve um rendimento de acerto entre 80% e 60%.

Conforme detalhado por Martins *et al.* (2011), foi observado que na grande maioria dos testes o uso simultâneo das características extraídas da GLCM não trouxe benefícios para a avaliação das imagens. Muito pelo contrário, as

configurações e cenários de testes demonstraram uma variação acima do esperado em relação ao comportamento das demais características deixadas de fora da avaliação.

Referente às características extraídas da GLCM analisadas neste trabalho, a que mais se destacou foi a energia, de forma a concretizar a disparidade entre os *ratios* de imagens mais distintos. A homogeneidade e correlação foram mais efetivas na comparação de imagens complexas.

6 CONCLUSÃO

Com base no estudo realizado, que envolveu conceitos de Visão Computacional, análise de textura e classificadores de imagem, procurou-se propor a validação da técnica GLCM como um descritor textural, a fim de classificar imagens a partir de suas características e aprimorar a eficácia da ferramenta a partir de seleção de características mais eficazes no âmbito de análise.

Considerando as demais técnicas de extração de características presentes na literatura, a GLCM mostrou-se ideal para a classificação de imagens digitais sintéticas, suas propriedades analíticas sobre tons de cinza e disparidade de pixels manifestam bons resultados ao serem configurados exclusivamente para o ambiente a ser trabalhado, direcionando o foco da análise para as características mais relevantes.

Em relação ao desenvolvimento da ferramenta, tomou-se como a melhor linguagem para testes classificatórios e de validação o Python, a sintaxe clara e sucinta do Python beneficia a legibilidade do código-fonte, favorecendo a implementação de algoritmos exploratórios. No ambiente de desenvolvimento Jupyter, as bibliotecas Matplotlib e Scikit-Image foram manipuladas para formar a GLCM, para que assim, o banco de imagens selecionado possa ser processado e comparado.

Com intuito de validar a proposta e selecionar as melhores características providas pela GLCM, testes preliminares foram realizados envolvendo o uso de GLCM sobre imagens sintéticas, onde o descritor apresentou resultados promissores, pois mesmo em um conjunto de amostras pequeno foi possível reconhecer padrões em seus resultados de descrição. Nos testes preliminares, notou-se uma boa resposta das características de Energia, Correlação e Homogeneidade, que acabaram se tornando o enfoque dos testes definitivos.

Com a realização dos testes definitivos, foi obtido um padrão mais definido de classificação textural a partir da GLCM. As rodadas de testes colocaram à prova a eficiência da GLCM como descritor textural, assim como a definição de configurações mais apropriadas para cada situação.

No decorrer dos testes, a característica de melhor resultado foi a Energia, tendo sua aparição na classificação de imagens praticamente todos os testes, com a eficácia de análise comparativa próxima a 98%, apenas ocorrendo uma queda de rendimento ao ser utilizada para classificar imagens muito semelhantes. Com um declínio no desempenho da classificação, foram necessárias várias tentativas de encontrar a configuração ideal para a classificação de imagens semelhantes e que possuem estruturas texturais complexas.

Na tentativa de encontrar uma relação comparativa aceitável entre as imagens de *ratio* 05, 08 e 12, as características de Correlação e Homogeneidade foram mais eficazes. Com um rendimento variável entre 80% e 60%, a GLCM mostrou resultados satisfatórios que indicam que ainda é possível melhorar as configurações de análise para se obter resultados mais definitivos.

Porém, como visto em trabalhos relacionados, a GLCM necessita de técnicas de processamento de imagem para que haja uma consistência em seus resultados. Tais técnicas podem ser substituídas por uma configuração adequada da ferramenta, como foi comprovado nos testes.

Os resultados obtidos neste trabalho mostraram-se satisfatórios quando comparados com trabalhos de validação e aprimoramento presentes na literatura de análise de imagens. De forma a comprovar a afirmação satisfatória, pode ser levado em consideração os resultados dos testes de validação, onde é observado

uma porcentagem de eficácia muito alta quando a GLCM está configurada para analisar o ambiente desejado.

Entre as possibilidades de trabalhos futuros, é possível elencar o uso de *Machine Learning* para a dedução da melhor configuração para a situação de interesse, consistindo na viabilidade de ter uma ferramenta independente de manuseio humano em relação a sua configuração. Onde seria analisado e tomado nota das consistências através de testes, pois com a grande variação das possíveis configurações da GLCM, uma definição estática de avaliação não tem o desempenho que teria uma avaliação com enfoque na situação alvo.

A continuidade também pode ser exercida através do incremento de deduções obtidas por técnicas como LBP e LPQ na análise de resultados, possibilitando um embasamento mais concreto da efetividade das técnicas mais antigas e recentes.

Por fim, é possível considerar que a ferramenta desenvolvida em Python, no ambiente Jupyter onde é simulada a GLCM, é capaz de classificar imagens e ainda aprimorar a classificação através da seleção de parâmetros. Estima-se que com a implementação de possíveis melhorias e do *Machine Learning*, a GLCM possa se tornar uma ferramenta descritiva independente.

REFERÊNCIAS BIBLIOGRÁFICAS

ANTONELLO, Ricardo. **Introdução a Visão Computacional com Python e OpenCV. Engenharia de Controle e Automação do Instituto Federal Catarinense–IFC, campus Luzerna**, 2014. Acesso em: 15 de mar. 2020.

BACKES, André Ricardo; JUNIOR, Jarbas Joaci de Mesquita Sá. **Introdução à Visão Computacional usando Matlab**. Alta Books Editora, 2019. Acesso em: 17 de mar. 2020.

BORGES, Luiz Eduardo. **Python para desenvolvedores: aborda Python 3.3**. Novatec Editora, 2014. Acesso em: 11 de abr. 2020.

CHANG, C. C; LIN, C. J. (2001). **LIBSVM: a library for support vector machines**. Disponível em: <<http://www.csie.ntu.edu.tw/~cjlin/libsvm>>. Acesso em: 13 de abr. 2020.

Clausi, D. A. **An analysis of co-occurrence texture statistics as a function of grey level quantization**. Can. J. Remote Sensing, 28(1):45–62, 2002. Acesso em: 13 de abr. 2020.

LIMA, Antonio Wagner Rodrigues. **Análise de Dados com Python: Avaliação do Potencial Geotérmico do Estado da Paraíba**. Acesso em: 25 de abr. 2020.

SILVA, Wallinson et al. **Detecção Automática do Glaucoma pelo uso de Descritores Locais, Matriz GLCM e Aprendizado de Máquina**. Acesso em: 26 de abr. 2020.

EMERSON, C. W., LAM, N. S., QUATTROCHI, D. A. **Multi-Scale fractal analysis of image texture and pattern. Photogrammetric Engineering & Remote Sensing**, v. 65, n. 1, p. 51-61, 1999. Acesso em: 01 de jun. 2020.

FENSTERSEIFER, Henrique. **Simulação de padrões biológicos através de reação-difusão em superfícies submetidas ao crescimento**. 2018. Monografia (Graduação em Engenharia da Computação) – Universidade do Vale do Taquari -

Univates, Lajeado, 06 dez. 2018. Disponível em: <<http://hdl.handle.net/10737/2327>>. Acesso em: 26 de abr. 2020.

GIL, Antonio Carlos et al. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002. Acesso em: 03 de jun. 2020.

JESUS, Edison O.; COSTA, Roberto. A utilização de filtros gaussianos na análise de imagens digitais. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 3, n. 1, 2015. Acesso em: 18 de abr. 2020.

JUCÁ, Vinicius; MELLO, Carlos. Texture-based segmentation using image fidelity indexes. **IEEE Latin America Transactions**, v. 9, n. 3, p. 423-428, 2011. Acesso em: 04 de abr. 2020.

SOH, L; TSATSOULIS C. **Texture analysis of sar sea ice imagery using gray level co-occurrence matrices**. Geoscience and Remote Sensing, IEEE Transactions on, 37(2):780–795, 1999. Acesso em: 13 de abr. 2020.

MARTINS, Jefferson Gustavo et al. Uso de descritores de textura extraídos de glcm para o reconhecimento de padrões em diferentes domínios de aplicação. In: **XXXVII Conferência Latinoamericana de Informática**. 2011. p. 637-652. Acesso em: 04 de abr. 2020.

MATHWORKS, MATLAB. **Image Processing Toolbox**, version 6.3, 2009. Acesso em: 07 de jun. 2020.

MATPLOTLIB DOCUMENTATION. Disponível em <<https://matplotlib.org/Matplotlib.pdf>>. Acesso em: 07 mar. 2020.

MUSCI, Marcelo et al. Uma avaliação de descritores de textura baseados em códigos binários locais para classificação de imagens de sensoriamento remoto. **Boletim de Ciências Geodésicas**, v. 17, n. 4, p. 549-570, 2011. Acesso em: 05 de abr. 2020.

OULU. University of Oulu - **Department of Computer Science and Engineering, LBP** Disponível em: <<http://www.cse.oulu.fi/MVG/Downloads/LBP Matlab>>. LPQ Disponível em: <<http://www.cse.oulu.fi/Downloads/LPQ Matlab>>. Acesso em: 13 de abr. 2020.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição**. Editora Feevale, 2013. Acesso em: 02 de out. 2020.

RAMPUN, Andrik; STRANGE, Harry; ZWIGGELAAR, Reyer. (2013). **Texture Segmentation Using Different Orientations of GLCM Features**. ACM International Conference Proceeding Series. 10.1145/2466715.2466720. Acesso em: 05 de abr. 2020.

RIBEIRO, Patricia Pedroso Estevam; PAIVA, Maria Stela Veludo; JORGE, Lúcio André de Castro. Application of texture analysis for differentiation of the greening from others pests. In: **Embrapa Instrumentação-Artigo em anais de congresso**

(ALICE). In: WORKSHOP DE VISÃO COMPUTACIONAL-WVC, 10., 2014, Uberlândia. Anais... Uberlândia: UFU, 2014. p. 202-207., 2014. Acesso em: 14 de ago. 2020.

SIQUEIRA, F. Roberti de; SCHWARTZ, W. Robson; PEDRINI, H. (2013). **Multi-scale gray level co-occurrence matrices for texture description**. *Neurocomputing*, 120, 336–345. doi:10.1016/j.neucom.2012.09.042. Acesso em: 05 de abr. 2020.

ROCHA, Ana Maria AC et al. Processamento de imagem digital com MatLab: **Uma aplicação em ambiente industrial**. 2013. Acesso em: 19 de abr. 2020.

SCIKIT-IMAGE DOCUMENTATION. Disponível em <<https://scikit-image.org/docs/dev/api/skimimage.feature.html#skimimage.feature.greycomatrix>>. Acesso em: 07 mar. 2020.

SETTE, Priscila Gonçalves Couto; MAILLARD, Philippe. Análise de textura de imagem de alta resolução para aprimorar a acurácia da classificação da mata atlântica no sul da Bahia. **Anais do XV Simpósio Brasileiro de Sensoriamento Remoto**, p. 2020-2026, 2011. Acesso em: 22 de mar. 2020.

VALENÇA, Jónatas Miguel de Almeida. **Monitorização do estado de conservação de estruturas de betão por processamento de imagem e análise multi-espectral**. 2011. Tese de Doutorado. Acesso em: 22 de abr. 2020.

VIEIRA, R. T.; CHIERICI, C. E. O.; FERRAZ, C. T.; GONZAGA, A. Local Fuzzy Pattern: **A New Way for Micro-pattern Analysis**, IDEAL 2012, H. Yin et al. (Eds.): LNCS 7435, Springer-Verlag Berlin Heidelberg 2012, pp. 602–611, 2012. Acesso em: 08 de out. 2020.

WALT, S. van der et al. Scikit-image: **image processing in Python**. *PeerJ*, v. 2, p. e453, 6 2014. ISSN 2167-8359. Acesso em: 02 de jun. 2020.