

Novo descritor de imagens baseado em árvore geradora mínima

New image descriptor based on minimal spanning tree

RESUMO

Felipe Seolin Bento
bentof@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Pedro Henrique Bugatti
pbugatti@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Há uma grande quantidade de algoritmos que buscam realizar a descrição de imagens da maneira mais precisa e eficaz possível, isto porque esta é uma etapa muito importante no processamento de imagens digitais. Este trabalho traz a abordagem de um novo descritor baseado em árvores geradoras mínimas, o qual consiste em ligar os *pixels* com todos seus vizinhos (vizinhança oito), calcular um peso para cada aresta, composto pela diferença dos valores dos *pixels*, encontrar a MST, e, por fim, realizar algumas medidas estatísticas com os valores obtidos. Sendo assim, o algoritmo proposto quando comparado a outros descritores presentes na literatura se mostrou eficiente, obtendo resultados de acurácia e matriz de confusão similares.

PALAVRAS-CHAVE: Processamento de Imagens. Teoria dos grafos. Árvores (Teoria dos grafos).

ABSTRACT

There are a large number of algorithms that seek to carry out of description in the most accurate and effective way possible, because it is a very important step on digital image processing. This work brings the approach of a new descriptor based on minimum spanning trees, which consists of connecting the pixels with all their neighbors (neighborhood eight), calculating a weight for each edge, composed of the difference of the pixels values, finding the MST and finally perform some statistical measurements with the values obtained. Thus, the proposed algorithm when compared to other descriptors in the literature proved to be efficient, obtaining results of similar accuracy and confusion matrix.

KEYWORDS: Image processing. Graph theory. Trees (Graph theory).

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

O mundo está repleto de dados, que podem ser encontrados e processados de formas distintas. Isto não é diferente com imagens digitais, pois estas representam uma quantidade significativa das medidas realizadas. Porém, para imagens de ressonância magnéticas, tomografias, observações astronômicas e outras fazerem sentido é preciso extrair informações, sendo possível devido ao processamento de imagens digitais (Walt et al., 2014).

Gonzalez e Woods (2009) definem o processamento digital de imagens não apenas como a entrada de uma imagem que é processada e resulta em outra, mas também como o reconhecimento de determinadas áreas, atributos, cores e texturas, por exemplo.

Junior, Backes e Cortez (2013) ainda dizem que tal área possui grande importância na computação visual, sendo que a textura é caracterizada como um dos elementos principais e essenciais para alguns campos, entre estes: detecção de objetos, recuperação de imagens, sensoriamento remoto e outros.

Na literatura não há uma definição exata sobre textura, sendo que alguns caracterizam como a repetição de um modelo diversas vezes na imagem com pequenas variações. Outros, no entanto, dizem que até mesmo a ausência de padrões já é capaz de caracterizar uma textura. Uma das definições que é bem aceita, diz que textura corresponde a padrões visuais complexos que são formados por entidades ou sub-padrões, com características específicas como: tamanho, brilho e outras (JUNIOR; CORTEZ; BACKES, 2014).

Atualmente, a textura vêm sendo um atributo de descrição muito estudado e fonte de diversas pesquisas, apesar de já existirem métodos consolidados, os estudos sempre buscam abordagens diferentes, seja por matrizes de co-ocorrência, análise espectral ou outros (JUNIOR; BACKES; CORTEZ, 2013).

Este trabalho teve como objetivo desenvolver uma nova abordagem para a descrição de texturas em imagens, sendo baseada em árvores geradoras mínimas, também conhecidas como *minimum spanning trees* (MST).

MATERIAL E MÉTODOS

Pensando-se nos *pixels* e na estrutura de imagens, muitos a definem como um plano cartesiano. No entanto, uma imagem corresponde a uma função bidimensional, em que os eixos x e y representam o plano espacial, enquanto f corresponde aos valores finitos dos graus de cinza. Os *pixels*, por sua vez, são definidos como elementos que realizam a composição da imagem, sendo que cada um desses possui características diferentes devido a suas posições e valores (GONZALEZ; WOODS, 2009).

Para o desenvolvimento do algoritmo e para os testes, foi definida como linguagem de programação a ser utilizada o *Python*, pois esta possui várias vantagens, tais como: a possibilidade da escrita de códigos legíveis, ser *open-source* com uma comunidade presente na evolução da linguagem e a vasta coleção de bibliotecas, sendo várias de cunho científico.

A biblioteca *scikit-image* foi de extrema importância para o desenvolvimento, pois esta lidou com todo tratamento das imagens, desde a importação dos arquivos, até sua conversão para escala de cinza e comunicação com os outros módulos.

Essa biblioteca possui várias funções que podem ser facilmente utilizadas, fornecendo rotinas versáteis e intuitivas de processamento digital de imagens, provendo submódulos dedicados a: cor, desenho, exposição, análise de texturas e bordas, grafos, entrada e saída, entre tantos outros que indiscutivelmente podem auxiliar a muitos na construção de um software destinado a processamento de imagens (WALT et al., 2014).

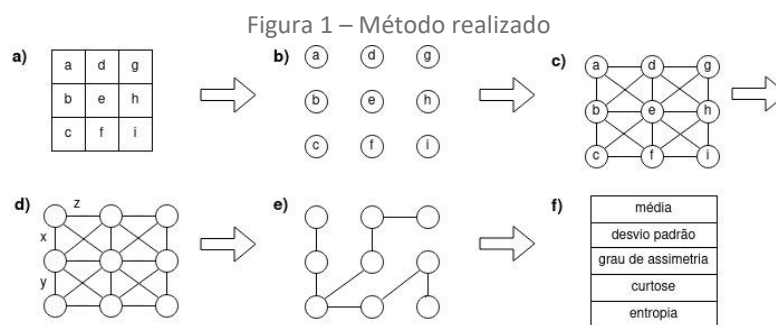
Outra biblioteca utilizada foi a *igraph*, que auxiliou a construir os grafos para uma determinada imagem, atribuir os pesos aos vértices e também encontrar a árvore geradora mínima.

Vale destacar o uso de outras bibliotecas, entre elas: *pandas*, *numpy*, *scipy*, *matplotlib*, *ipython*, *cython*, *pycairo* e *scikit-learn*, sendo que esta última foi importante durante a condução dos testes, validações e comparações com outros algoritmos de texturas já existentes na literatura.

Desta forma, o método realizado para a extração de características de textura de uma determinada imagem, através de árvores geradoras mínimas, é composto através dos seguintes passos:

- a) selecionar uma imagem;
- b) converter a imagem em um grafo sem arestas, em que cada *pixel* representa um vértice e armazena o seu valor em *f*;
- c) ligar todos os *pixels* com seus vizinhos, caracterizando vizinhança oito;
- d) atribuir valores (pesos) a cada aresta do grafo formado, de modo que este seja composto pelo módulo da diferença entre os valores dos *pixels* (vértices);
- e) encontrar e selecionar uma árvore geradora mínima do grafo, utilizando a biblioteca *igraph* e o algoritmo de Prim;
- f) extrair medidas estatísticas dos valores das arestas, entre essas: média, desvio padrão, grau de assimetria, curtose e entropia.

A Figura 1 mostra o método realizado e descrito, em que cada passo realizado é destacado.



Fonte: Autoria própria (2020).

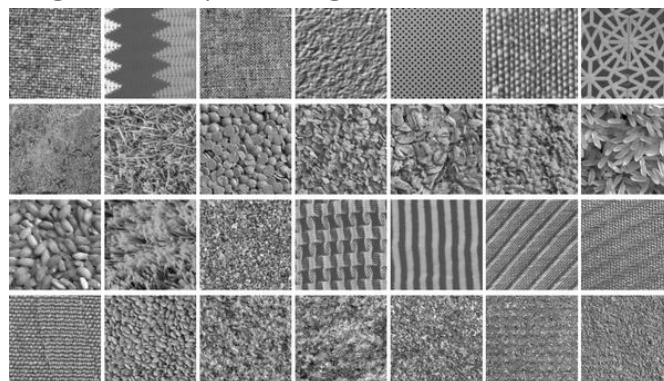
Vale destacar que uma MST é um grafo acíclico que busca percorrer o menor caminho possível e que está contido em um grafo. O termo “árvore geradora mínima” é uma miniatura de sentença de “árvore geradora de peso mínimo”. Tal explicação se faz necessária, pois existe a possibilidade de que o termo leve a uma má interpretação, pois pode-se pensar que se está diminuindo o número de arestas em uma árvore geradora, o que não corresponde à verdade (CORMEN et. al, 2012).

Por fim, vale ressaltar que o método só abrange um único canal da imagem, e que todos os testes foram realizados com imagens em níveis de cinza. Assim, ainda não foram considerados os canais RGB e como seriam as interações entre eles.

RESULTADOS E DISCUSSÃO

A fim de comprovar a eficácia do algoritmo, foram realizados alguns testes com o conjunto de imagens de texturas de Kylberg (2011). Tal *dataset* é composto por 28 classes de texturas (Figura 2), 160 imagens originais em cada classe, resolução de 576x576 *pixels* e disponibilizadas em PNG.

Figura 2 – Exemplos de imagens de cada uma das classes



Fonte: Kylberg (2011).

Também para a realização dos testes, a fim de comparação, foram escolhidos alguns algoritmos para extração de características de texturas já presentes na literatura, entres estes: Gabor (Zhang et al., 2000), Haralick (Haralick et al., 1973), LBP (Guo et al., 2010) e Tamura (Tamura et al., 1978).

Deste modo, para cada um dos algoritmos foram realizados quatro testes com uso de aprendizado supervisionado, sendo os classificadores utilizados: *decision tree*, *k-nearest neighbors* (KNN), *linear support vector* (*linear SVC*) e *c-support vector* (SVC).

Antecedendo a realização dos testes foram extraídas as características do *dataset* para os outros métodos de descrição, sendo disponibilizados arquivos do tipo *Attribute-Relation File Format* (ARFF) para cada extrator. Para o algoritmo tratado neste documento, foi disponibilizado um arquivo na formatação *Comma-Separated Values* (CSV).

Além disso, foram definidas algumas premissas para todos os testes, sendo que: o conjunto de testes corresponde a 20% das imagens, o conjunto de treino

corresponde a 80%, a semente para valores aleatórios igual a 5 e que a divisão das imagens entre treino e teste deveriam ser a mesma para todos os testes.

Na Tabela 1, é possível observar a relação obtida das acurácias para cada extrator e classificador, sendo o método proposto descrito como MST.

Tabela 1 – Resultados das acurácias obtidas

	Gabor (%)	Haralick (%)	LBP (%)	Tamura (%)	MST (%)
Decision Tree	19,98	88,28	93,30	73,10	88,06
KNN	18,64	46,21	96,21	59,38	79,46
Linear SVC	9,71	5,69	99,33	26,34	68,75
SVC	8,82	29,58	94,08	64,17	55,92

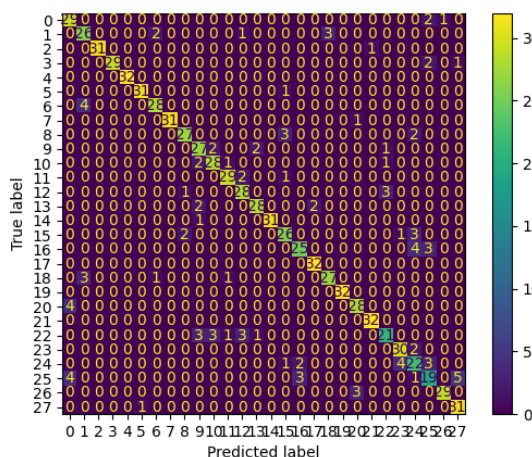
Fonte: Autoria Própria (2020).

Outra medida observada foi a matriz de confusão dos algoritmos, a qual fornece uma visão geral das medidas em forma de imagem e também o detalhamento para cada classe. Através da matriz de confusão, pode-se observar a quantidade de falsos negativos, falsos positivos, verdadeiros positivos e verdadeiros negativos.

Também foram obtidos os valores: precisão, *recall* e *f1-score*, que assim como a acurácia são obtidos através da matriz de confusão. No entanto, os valores apresentados se encontram iguais ou próximos, devido às 28 classes presentes no *dataset*. Desta forma, o cálculo é realizado de forma *macro*, o que leva em consideração métricas para cada classe e encontra a média sem pesos. Porém, a melhor forma é observar tais resultados classe por classe, para isto foi disponibilizado em: <https://bit.ly/sicite-resultados>.

O algoritmo proposto teve como resultado uma precisão de 88%, *recall* de 88% e *f1-score* de 88%, com a utilização do classificador árvore de decisão. A Figura 3 retrata esta matriz de confusão.

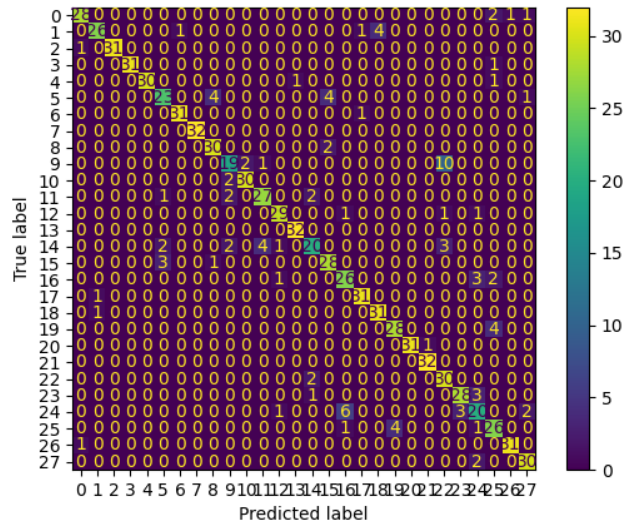
Figura 3 – Matriz de confusão do algoritmo proposto com uso de árvore de decisão



Fonte: Autoria Própria (2020).

Já o extrator Haralick, utilizando árvore de decisão, teve como resultado uma precisão de 89%, *recall* de 88% e *f1-score* de 88%. A Figura 4 retrata a matriz de confusão obtida.

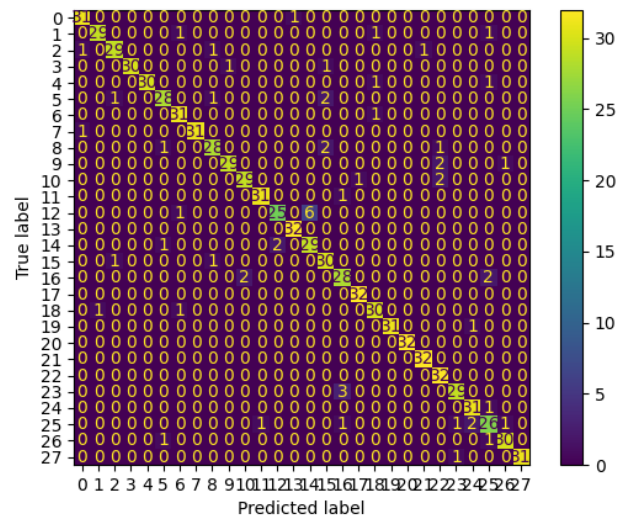
Figura 4 – Matriz de confusão do extrator Haralick com uso de árvore de decisão



Fonte: Autoria Própria (2020).

Também vale ressaltar o extrator LBP, o qual obteve as maiores acurácias. Com o uso de árvore de decisão, resultou em uma precisão de 93%, *recall* de 93% e *f1-score* de 93%. A Figura 5 retrata a matriz de confusão obtida.

Figura 5 – Matriz de confusão do extrator LBP com uso de árvore de decisão



Fonte: Autoria Própria (2020).

CONCLUSÃO

A partir dos resultados obtidos é possível afirmar que o método de extração de características com árvore geradora mínima se mostra eficiente e pode ser utilizado para a caracterização de texturas de imagens.

Mesmo alguns algoritmos obtendo resultados de acurácia e matriz de confusão melhores, este extrator é capaz de coletar as informações necessárias para o reconhecimento de uma textura de uma imagem com um número reduzido de características e com eficácia. A Tabela 2, exibe uma relação entre os extratores e número de características que cada um utiliza.

Tabela 2 – Extratores e quantidade de características extraídas

Extrator	Quantidade de Características
Gabor	60
Haralick	14
LBP	256
Tamura	18
MST (método proposto)	5

Fonte: Autoria Própria (2020).

É importante destacar que medidas de custo computacional e tempo de execução não foram levadas em consideração neste trabalho, pois o processo de extração de características de uma imagem (descrição) é, em geral, um processo custoso e demorado, sendo sempre realizado de forma prévia.

No entanto, é importante ressaltar que o algoritmo proposto apresentou um tempo relativamente alto, quando comparado aos outros algoritmos, o que faz com que ele precise ser aperfeiçoado e otimizado. Isto pode ser feito reduzindo o uso de algumas bibliotecas que fornecem objetos e funções não necessárias, mas mesmo assim o processamento é executado. Como exemplo, a biblioteca *igraph* a qual retorna sempre um objeto *Graph* como resultado, sendo que só é necessário um *array* com os valores das arestas da MST.

Por fim, vale ressaltar que o uso de grafos e de árvore geradora mínima, que já são soluções utilizadas por diversos problemas na computação, mostra-se mais uma vez eficaz.

AGRADECIMENTOS

À Universidade Tecnológica Federal do Paraná, especialmente ao campus Cornélio Procópio, que proporcionou e incentivou esta iniciação científica voluntária.

REFERÊNCIAS

CORMEN, T. H. et al. **Algoritmos: teoria e prática**. [S.l.]: Elsevier, 2012. ISBN9788535236996.

GONZALEZ, R. C.; WOODS, R. **Processamento de Imagens Digitais**. [S.l.]: Edgard Blucher, 2000. ISBN 8521202644.

GONZALEZ, R. C.; WOODS, R. C. **Processamento Digital de Imagens**. 3. ed. [S.l.]: Pearson, 2009. ISBN 8576054019.

GUO, Z., ZHANG, L., and ZHANG, D. Rotation invariant texture classification using LBP variance (LBPV) with global matching. **Pattern recognition**, Elsevier, v.43, n. 3, p. 706–719, 2010.

HARALICK, R. M., SHANMUGAM, K., and DINSTEN, I. Textural features for image classification. **IEEE Transactions on Systems, Man, and Cybernetics**, IEEE, n; 6, p. 610–621, 1973.

JUNIOR, J. J. d. M. S.; BACKES, A. R.; CORTEZ, P. C. Texture analysis and classification using shortest paths in graphs. **Pattern Recognition Letters**, Elsevier, v. 34, n. 11, p. 1314–1319, 2013.

JUNIOR, J. J. d. M. S.; CORTEZ, P. C.; BACKES, A. R. Color texture classification using shortest paths in graphs. **IEEE Transactions on Image Processing**, IEEE, v. 23, n. 9, p.3751–3761, 2014.

KYLBERG, G. **The Kylberg Texture Dataset v. 1.0**. [S.l.], 2011. Disponível em: <http://www.cb.uu.se/~gustaf/texture/>. Acesso em: 30 ago. 2020.

TAMURA, H., MORI, S., and YAMAWAKI, T. (1978). Textural features corresponding to visual perception. **IEEE Transactions on Systems, man, and cybernetics**, 8(6): 460–473.

WALT, S. van der et al. Scikit-Image: Image Processing in Python. **PeerJ**, v. 2, p. e453, 62014. ISSN 2167-8359. Disponível em: <https://doi.org/10.7717/peerj.453>. Acesso em: 30 ago. 2020.

ZHANG, D., Wong, A., INDRAWAN, M., and LU, G. (2000). Content-based image retrieval using gabor texture features. **IEEE Transactions PAMI**, p. 13–15.