

Avaliação de superpixels para segmentação de imagens

Felipe Augusto Lima Reis¹

¹PUC Minas - Pontifícia Universidade Católica de Minas Gerais

`falreis@sga.pucminas.br`

Abstract. *Superpixels are structures that group similar pixels into sets that reflect aspects of the image. This article evaluates the use of SLICO superpixels and partition hierarchy for segmentation. Using **neural networks** for segmentation, the results of superpixels images with different levels of granularity and untreated images were compared to the ground-truth. The article also evaluates the training time of neural networks for superpixels based images. For training and evaluation, the Berkeley Segmentation Data Set (BSDS500) [Arbelaez et al. 2011] image was used.*

Resumo. *Superpixels são estruturas que agrupam pixels semelhantes em conjuntos que refletem aspectos da imagem. Este artigo avalia a utilização de superpixels SLICO e EGB para segmentação. Avalia também os benefícios da combinação dos métodos para produção de segmentação. Por fim, o artigo mostra uma possível implementação do método hierárquico para a segmentação combinada dos métodos SLIC e EGB, que pode ser utilizada para detalhamento e zoom sobre segmentações. Para avaliação foram utilizadas o conjunto de treinamento de imagens do Berkeley Segmentation Data Set (BSDS500) [Arbelaez et al. 2011]. Os resultados obtidos pelos métodos foram comparados em relação ao ground-truth, utilizando o método de precisão e revocação.*

Introdução

A segmentação de imagens consiste em dividir uma imagem em um conjunto de regiões logicamente agrupadas, de modo a reunir áreas que contém informação relevante dentro dos grupos [Domínguez and Morales 2016]. Nessa tarefa, tomamos os *pixels* como unidades básicas de processamento [Wang et al. 2017a]. O agrupamento de pixels em unidades maiores permite um tipo de segmentação chamado de *oversegmentation* [Wang et al. 2017a], ilustrado na figura 1. O uso de superpixels possibilita o aumento da velocidade de processamento posterior, uma vez que a quantidade de pixels diminui consideravelmente em relação a imagem original.

A utilização de superpixels possibilita a redução de itens a serem processados, entretanto pode causar perda de informação importante. No entanto, para alguns casos, a perda de qualidade pode se justificar em relação ao ganho de velocidade obtido utilizando esse tipo de operação. Essa relação consiste então em um *trade-off* entre ambas as características, sendo viáveis em alguns cenários de processamento em tempo real ou para dispositivos com baixo desempenho.

Alguns métodos de geração de superpixels são utilizados para segmentação de imagens e detecção de bordas, como os métodos EGB [Felzenszwalb and Huttenlocher 2004] e SLIC [Achanta et al. 2012]. Esse trabalho

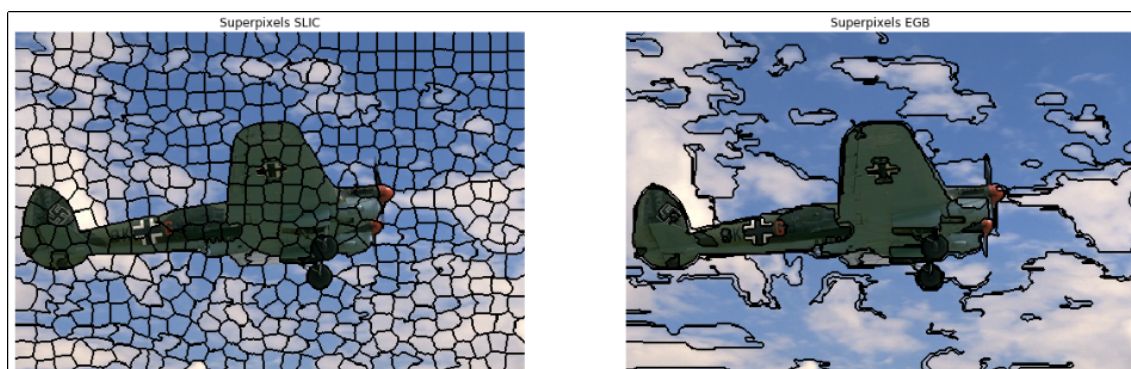


Figura 1. Imagens segmentadas utilizando superpixels SLIC e EGB

investiga se a utilização de métodos segmentação baseados em superpixels, a composição de métodos de *oversegmentation* e a utilização de redes neurais convolucionais para segmentação.

Em relação aos métodos de segmentação utilizando redes neurais, o trabalho também tentará identificar se o treinamento utilizando imagens pré processadas obtêm resultados semelhantes àqueles utilizando imagens originais, na etapa de validação.

O presente trabalho apresenta a seguinte estrutura: a Seção 2 mostra o referencial teórico para construção do trabalho, a Seção 3, exibe os materiais e métodos utilizados nos testes; a Seção 4 mostra os resultados obtidos nos testes realizados e a discussões dos mesmos; a Seção 5 contém a conclusão do artigo, com as considerações finais.

Referencial Teórico

Superpixels

Superpixels são estruturas que agrupam pixels semelhantes em conjuntos. O agrupamento possibilita a redução de complexidade das tarefas de processamento [Achanta et al. 2012], ao reduzir a quantidade de itens a serem processados. Os superpixels são utilizados na área de visão computacional para solução de vasto número de problemas, como detecção de contorno [Wang et al. 2017b], segmentação [Liu et al. 2016] e localização de objetos [Fulkerson et al. 2009].

Superpixels, segundo [Felzenszwalb and Huttenlocher 2004], devem capturar importante grupos ou regiões, refletindo aspectos da imagem. Devem também ser executados em tempo próximo ao linear em relação a quantidade de pixels. Existem diversas abordagens para a geração de superpixels [Achanta et al. 2012]. Dentre elas, podemos classificá-las, segundo o método de agrupamento em:

- *Algoritmos baseados em grafos*: utilizam abordagem baseadas em grafos para correlação entre pixels e criação dos conjuntos. Dentre os algoritmos baseados em grafos podemos citar o *Efficient Graph-Based Image Segmentation* (EGB) [Felzenszwalb and Huttenlocher 2004];
- *Algoritmos baseados em gradiente ascendente*: utilizam métodos de gradiente ascendente iterativamente até que os critérios de convergência correspondam a forma de um superpixel. Nesse conjunto, podemos citar as abordagens *watersheds* [Vincent and Soille 1991] [Achanta et al. 2012]

- *Algoritmos de clusterização iterativo*: utilizam métodos de clusterização, como o *k-means*, para produção de superpixels. Um exemplo desse algoritmo é o SLIC (*Simple Linear Iterative Clustering*) [Achanta et al. 2012]

Superpixels SLIC

O algoritmo SLIC utiliza um único parâmetro k , correspondente a quantidade aproximada de superpixels. A fim de produzir tamanhos semelhantes, o intervalo analisado é $S = \sqrt{N/k}$, onde N é o número de pixels da imagem. Os centros são movidos para o local das sementes correspondentes a posição mais baixa do gradiente em uma vizinhança de 3×3 , evitando que superpixels sejam centrados nas bordas ou em uma posição de ruído [Achanta et al. 2012]. Em seguida, cada pixel é associado com o centro do cluster mais próximo, de modo que a região de busca se sobreponham [Achanta et al. 2012]. A fim de aumentar o desempenho do algoritmo, a região de busca é limitada em 2 vezes o tamanho aproximado do superpixel S , gerando busca em uma área $2S \times 2S$ [Achanta et al. 2012]. Em seguida, um passo de atualiza os centros dos clusters e computa o erro residual E [Achanta et al. 2012]. O algoritmo disponível na figura 2, resume as informações descritas nesse parágrafo.

SLIC superpixel segmentation

```

/* Initialization */
Initialize cluster centers  $C_k = [l_k, a_k, b_k, x_k, y_k]^T$  by
sampling pixels at regular grid steps  $S$ .
Move cluster centers to the lowest gradient position in a
 $3 \times 3$  neighborhood.
Set label  $l(i) = -1$  for each pixel  $i$ .
Set distance  $d(i) = \infty$  for each pixel  $i$ .

repeat
  /* Assignment */
  for each cluster center  $C_k$  do
    for each pixel  $i$  in a  $2S \times 2S$  region around  $C_k$  do
      Compute the distance  $D$  between  $C_k$  and  $i$ .
      if  $D < d(i)$  then
        set  $d(i) = D$ 
        set  $l(i) = k$ 
      end if
    end for
  end for
  /* Update */
  Compute new cluster centers.
  Compute residual error  $E$ .
until  $E \leq \text{threshold}$ 

```

Figura 2. Algoritmo SLIC - Adaptado de [Achanta et al. 2012]

Para o algoritmo descrito na figura 2, é necessário compreender o método para cálculo da medida de distância D entre os conjuntos. Devido ao algoritmo trabalhar no *colorspace* CIELAB, com o espaço-plano *labxy*, a posição do pixel pode assumir um

intervalo de valores de acordo com o tamanho da image 2. Com isso, o cálculo da distância não pode ser feito utilizando uma distância euclidiana, sendo necessária uma prévia normalização da proximidade espacial e de cor. Para isso é utilizado a fórmula $D = \sqrt{d_c^2 + (d_s/S)^2 \cdot m^2}$, onde D corresponde a distância em 5 dimensões do espaço labxy, d_c e d_s correspondem à proximidade de cores e espacial; e m^2 corresponde a distância máxima entre cores no cluster [Achanta et al. 2012]. Para o cálculo da distância em imagens em escala cinza, é utilizada a distância Euclidiana.

Uma etapa extra no processo de pós processamento é a união de *pixels orfãos*. Esses pixels são adicionados ao cluster mais próximo usando o algoritmo de componentes conexos [Achanta et al. 2012].

Devido a limitação do espaço de pesquisa do algoritmo SLIC, a complexidade do algoritmo é $O(n)$, enquanto outros algoritmos que utilizam k-means para segmentação tem custo $O(k^N)$ [Achanta et al. 2012].

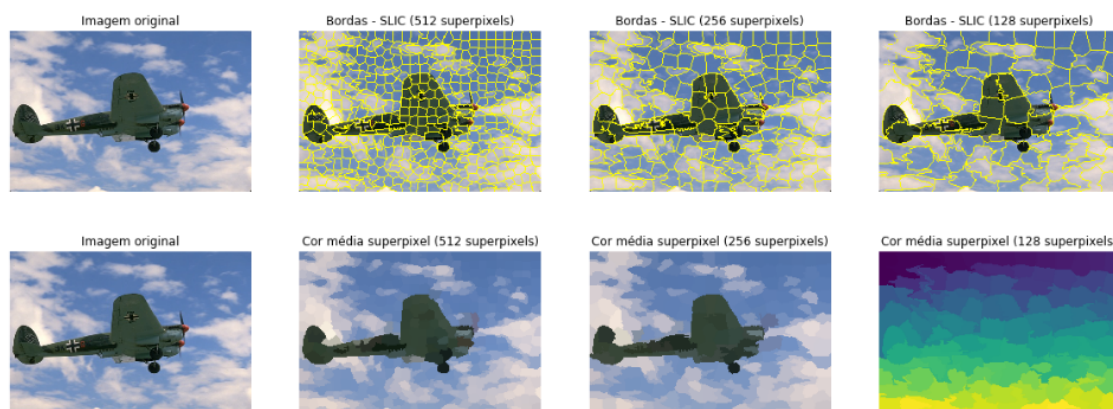


Figura 3. Fronteiras e coloração pelo valor médio dos superpixels SLIC/SLICO, para diferentes quantidades de superpixels

Superpixels EGB

Os superpixels EGB (*Efficient Graph-Based Image Segmentation*) utilizam uma abordagem baseada em grafos não direcionados. Nessa abordagem, cada pixel corresponde a um nó do grafo e a ligação entre eles ocorre por meio de arestas, com pesos não negativos, correspondente a medida de dissimilaridade [Felzenszwalb and Huttenlocher 2004].

Na abordagem baseada em grafos, a segmentação S é uma partição dos vértices V em componentes, no qual cada região $C \in S$ corresponde a um componente conectado em um grafo $G' = (V, E')$, onde $E' \subseteq E$, ou seja, a segmentação é induzida por um conjunto de vértices em arestas E [Felzenszwalb and Huttenlocher 2004].

No algoritmo foi definido um predicado D para avaliação da evidência de bordas entre dois componentes de uma segmentação. O algoritmo avalia a dissimilaridade entre elementos de dois componentes e os compara com elementos vizinhos em um mesmo componente, de modo que o algoritmo possa se adaptar em relação as características dos dados [Felzenszwalb and Huttenlocher 2004].

Para a comparação entre as regiões é utilizada uma função de corte (*threshold*) τ , a fim de medir o grau de diferença entre os componentes. Esse grau deve ser superior a diferença interna mínima, evidenciando uma borda [Felzenszwalb and Huttenlocher 2004]. A função de corte, no algoritmo, é utilizada baseado no tamanho do componente $\tau = k/|C|$, onde $|C|$ corresponde ao tamanho do componente C e k corresponde a um parâmetro do algoritmo [Felzenszwalb and Huttenlocher 2004].

O algoritmo EGB, utilizando pesos inteiros e ordenação por contagem, pode ser executado com custo linear, com complexidade $O(n \log n)$, para qualquer método de ordenação [Felzenszwalb and Huttenlocher 2004].

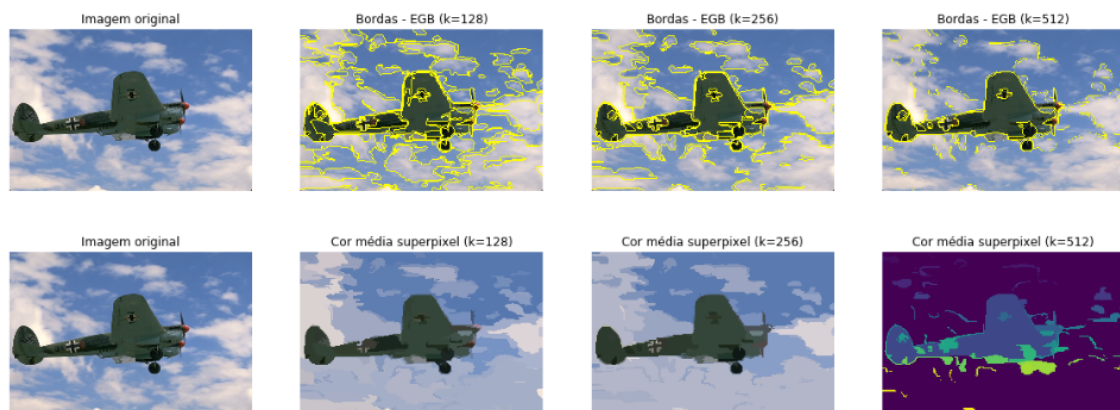


Figura 4. Fronteiras e coloração pelo valor médio dos superpixels EGB, para diferentes quantidades de superpixels

Clusters

Análise de cluster é a tarefa de agrupar conjunto de objetos de forma que um grupo (cluster) tenha características em comuns a outros grupos, ou clusters [Wikipedia 2018a]. Clusters hierárquicos é um método de análise de clusters a fim de buscar hierarquias entre eles. As estratégias para hierarquização de cluster se dividem em dois grupos [Rokach and Maimon 2005]:

- Aglomerativo - abordagem "*bottom up*", em que a observação inicia-se no próprio cluster e os pares de clusters são unidos na medida em que se sobe na hierarquia.
- Divisivo - abordagem "*top down*" onde as observações iniciam-se em um cluster e são realizadas recursivamente a medida em que se desce na hierarquia.

Os algoritmos originais de cluster possuem complexidade $O(n^3)$. Alguns algoritmos, entretanto, como o SLINK, ou *single-linkage*, possuem complexidade $O(n^2)$ [Sibson 1973]. O SLINK utiliza como critério de ligação (*linkage*) entre os clusters a distância mínima. A fórmula $\min\{d(a,b) : a \in A, b \in B\}$, corresponde a distância mínima para dois pares de clusters A e B observados, onde d é a métrica de distância escolhida [Wikipedia 2018c].

A partir da aglomeração de clusters é possível construir uma árvore hierárquica, onde os clusters são agrupados por sua similaridade. Um método de visualização dessas características é o dendrograma.

Segmentação de Imagens

Segmentação de imagens consiste em separar uma imagem em regiões, idealmente correspondente a objetos reais [Zhang et al. 2008]. Esse passo é utilizado como passo de processamento de imagens, vídeos e aplicações de visão computacional. Também consiste em um importante passo na tentativa de explicar uma imagem por meio de algoritmos [Zhang et al. 2008].

Extensiva pesquisa é realizada e muitas abordagens e algoritmos são utilizados, com bons resultados para um conjunto ou classes de imagens [Zhang et al. 2008]. A fim de facilitar a pesquisa, alguns trabalhos foram desenvolvidos para criação de um conjunto de imagens com suas respectivas segmentações manuais. A base BSDS500 (*Berkeley Segmentation Data Set and Benchmarks 500*) provê uma base de imagens para pesquisa em segmentação e detecção de bordas. [Arbelaez et al. 2011]. Essa base de dados é uma extensão da base BSDS300, com 200 novas imagens para avaliação [Arbelaez et al. 2011].

A fim de avaliar a efetividade das segmentações, tradicionalmente são utilizados métodos subjetivos, como a visualização humana, responsável por comparar a qualidade da segmentação ou métodos supervisionados, onde uma segmentação é comparada em relação a uma imagem manualmente segmentada [Zhang et al. 2008]. Projetos de detecção automática, como o SEISM (*Supervised Evaluation of Image Segmentation Methods*) permitem a avaliação da segmentação, usando a recuperação de precisão para bordas e a recuperação de precisão para objetos e peças [Pont-Tuset and Marques 2016]

Materiais e Métodos

Para confecção do trabalho foram escolhidos os algoritmos de *oversegmentation* SLIC e EGB. Os algoritmos possuem características diferentes: o SLIC é capaz de produzir superpixels em formas regulares, porém não é tão preciso ao separar os conjuntos por similaridade; por outro lado, o EGB, produz pixels irregulares, porém é mais aderente às diferenças de cores entre pixels.

Nos testes realizados foi realizada hierarquização dos resultados obtidos após a geração de superpixels SLIC e também sobre a composição de superpixels SLIC+EGB. Os resultados das hierarquizações foram comparados aos melhores resultados de segmentação obtidos pela segmentação SLIC, EGB e SLIC+EGB sem hierarquização. O método de hierarquização está descrito na seção 3.1, enquanto o método de avaliação de resultados está descrito na seção 3.3. Os resultados dos testes estão disponíveis na seção 4.

Hierarquia de Segmentação

Hierarquia Segmentação SLIC

A hierarquização de partições SLIC tem como objetivo possibilitar a utilização do algoritmo para produzir segmentação em diferentes níveis de detalhamento: desde um nível mais detalhado até um nível macro. Os diferentes níveis hierárquicos podem ser combinados em termos do contorno das hierarquias, permitindo a representação das hierarquias indexadas como bordas suaves, chamadas Mapas de Contorno Ultramétrico (UCM - *Ultrametric Contour Map*) [Arbelaez 2006].

Para construção das hierarquias foram gerados *clusters* com as correlações entre os grupos de superpixels, pós segmentação inicial SLIC. Foi utilizado o algoritmo SLINK (*single-linkage*) para agrupamento dos superpixels próximos, gerando uma árvore com as correlações entre os superpixels.

Para verificação da estrutura da árvore foi utilizado um dendrograma, como representado na figura 5. Dendrograma é um tipo de árvore utilizado para ilustração de uma clusterização hierárquica [Wikipedia 2018b]. Frequentemente é utilizado nas áreas de biologia para indicar a mudanças evolucionárias entre ancestrais e descendentes, baseados em suas características [Science-Dictionary 2018].

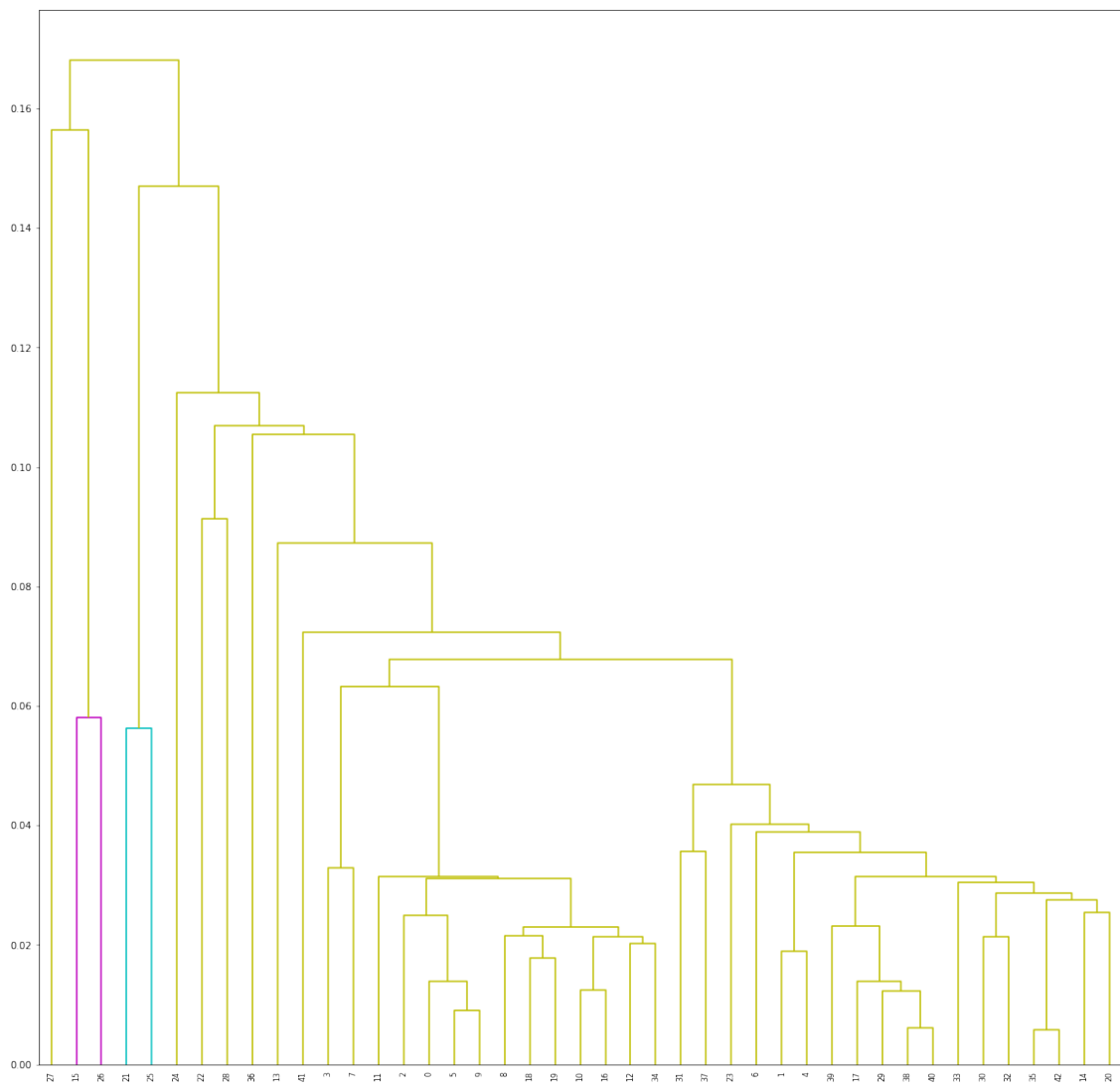


Figura 5. Dendrograma com a correlação entre os superpixels

Em um dendrograma é possível realizar cortes horizontais ou verticais na hierarquia, a fim de obter tipos de agrupamentos diferentes. No trabalho realizado somente foram realizados cortes horizontais. Os cortes foram feitos em diversos níveis da hierarquia.

Devido às características da segmentação, quando comparados ao *groundtruth*,

não foram validados todos os níveis hierárquicos, somente níveis intermediários que apresentaram melhores resultados em testes empíricos. Com isso, foi estabelecido um nível máximo e mínimo para os cortes na hierarquia, de modo que os cortes foram feitos dentro de um intervalo, evitando nós com baixa probabilidade adequação ao *groundtruth*.

Hierarquia Segmentação SLIC+EGB

Uma outra abordagem realizada nesse trabalho foi a segmentação conjunta utilizando o método SLIC e em sequência, o método EGB. A utilização de ambos os algoritmos busca obter um melhor desempenho em relação a utilização dos algoritmos de forma isolada.

O primeiro algoritmo aplicado às imagens foi o algoritmo SLIC, gerando superpixels de formas regulares. Em seguida, foi realizada recoloração da imagem utilizando o valor do pixel médio do superpixel, a fim de tornar a imagem mais simples, com características mais evidentes. Em seguida, foi aplicado às imagens o método EGB.

A partir do resultado do processo conjunto de segmentação, foi realizado o mesmo procedimento de hierarquização utilizado para o algoritmo SLIC. Nele, utilizou-se o algoritmo SLINK (*single-linkage*) para agrupamento dos superpixels próximos. Em seguidas cortes horizontais foram feitos na hierarquia, gerando diferentes níveis de visualização.

Análise Multiescala

Para que os níveis hierárquicos possam garantir a manutenção da informação, agrupando apenas características semelhantes, as hierarquias devem seguir os princípios de análise multiescala [Guimarães and Patrocínio 2013]. Esses princípios asseguram manutenção de duas características principais [Guimarães and Patrocínio 2013]:

- Causalidade - um contorno presente em uma escala k_1 deve estar presente em qualquer escala $k_2 < k_1$ [Guimarães and Patrocínio 2013];
- Localidade - a medida em que o número de regiões diminui, os contornos devem ser estáveis (não devem se mover). A união corresponde a manutenção das bordas dos grupos que se fundiram [Guimarães and Patrocínio 2013].

No método de hierarquização descrito nesse trabalho manteve as características da análise multiescala. A figura 6 mostra os possíveis cortes na hierarquia de uma segmentação SLIC+EGB, sem recoloração da imagem. Nesse modelo, a segmentação é capaz de unir pixels por similaridade.

É importante salientar que os métodos originais (SLIC e EGB), quando apenas variados os parâmetros de criação de superpixels não mantém as características de análise multiescala, conforme podemos ver nas figuras 3 e 4.

Avaliação de Resultados

Para avaliação das imagens, foi primeiramente utilizada a classificação visual, comparando a qualidade dos resultados. Em seguida utilizou-se um método de comparação de precisão e revocação.

O método de precisão é um método de classificação binária, onde a precisão (textitprediction) corresponde à “fração de instâncias recuperadas que são relevantes”, en-

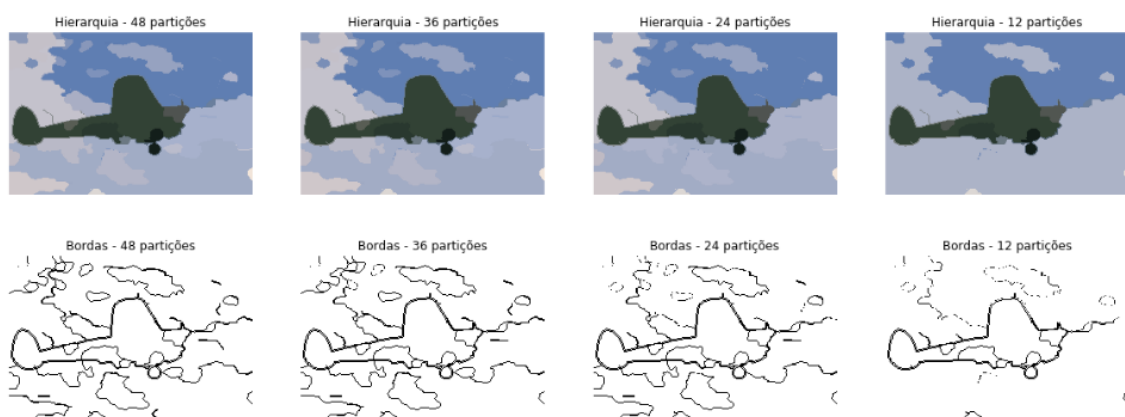


Figura 6. Hierarquia de partições utilizando superpixel SLIC+EGB.

quanto revocação (*recall*), corresponde “a fração de instâncias relevantes que são recuperadas” [Wikipedia 2018d]. Ambos os métodos são avaliados juntos para que possam identificar 4 tipos de valores possíveis:

- Verdadeiros Positivos - correspondem aos valores que foram corretamente classificados como positivos;
- Falsos Negativos - correspondem aos valores que foram classificados incorretamente como negativos;
- Falsos Positivos - correspondem aos valores que foram classificados incorretamente como positivos;
- Verdadeiros Negativos - correspondem aos valores que foram classificados corretamente como negativos [Wikipedia 2018d];

A classificação descrita acima está ilustrada na figura 7.

A classificação de precisão e revocação pode ser dada por uma média harmônica, chamada de *F-measure* ou *F-score* balanceada. Essa medida é dada pela fórmula [Wikipedia 2018d]:

$$F = 2 \cdot \frac{precis \cdot revoc}{precis + revoc} \quad (1)$$

A medida é aproximadamente a média quando seus valores estão próximos, porém o valor é baixo quando as médias estão distantes, favorecendo métodos com baixo número de falsos positivos e verdadeiros negativos [Wikipedia 2018d]

Código Fonte e Bibliotecas Utilizados

Os algoritmos SLIC e EGB utilizados para confecção desse trabalho foram obtidos pela biblioteca Scikit-Image¹. Os algoritmos de geração de cluster SLINK e representação de dendrogramas foram obtido na biblioteca Scipy.org². O algoritmo de avaliação do resultado, utilizando método de previsão e revocação foi obtido no projeto Image-Segmentation, no Github³.

¹<http://scikit-image.org/docs/dev/api/skimimage.segmentation.html>

²<https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.cluster.hierarchy.linkage.html>

³https://github.com/Htiango/Image-Segmentation/blob/master/main/eval_boundary.py

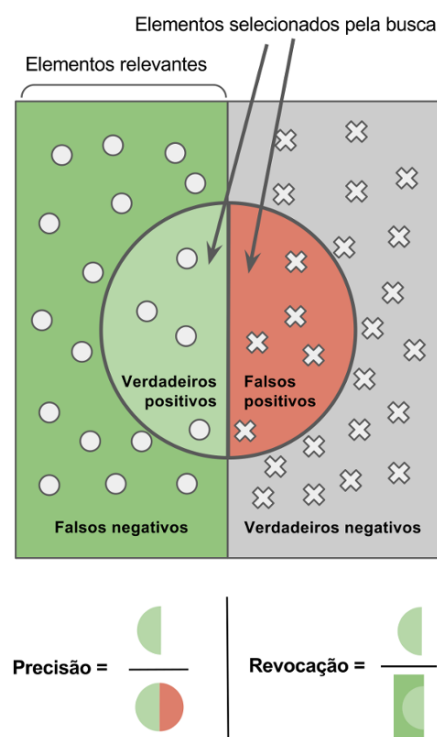


Figura 7. Precisão e revocação. Adaptado de [Wikipedia 2018d]

Os códigos fontes gerados, realizando as segmentações, hierarquias e avaliação dos resultados estão disponíveis publicamente na página pessoal do autor no Github ⁴.

Testes, Resultados e Discussões

Para avaliação do desempenho dos algoritmos foi executada a segmentação das imagens do conjunto de treinamento da base de dados BSD500 (*Berkeley Segmentation Data Set*). O conjunto possui 200 imagens naturais com seus respectivos *ground-truth* feito por anotações humanas [Arbelaez et al. 2011].

Os algoritmos foram aplicados a base de testes e comparados com os *ground-truths* disponíveis na base de dados. A adequação do algoritmo em relação ao *ground-truth* foi feita por meio do mecanismo de precisão e revocação, utilizando a medida *F-measure*, conforme descrito na seção 3.3.

A fim de obter a melhor parametrização possível, foram executados diversos testes para cada algoritmo. Somente o algoritmo EGB não foi extensamente executado a fim de obter a melhor parametrização, uma vez que os valores obtidos pelos autores estão disponíveis no artigo *Efficient Graph-Based Image Segmentation* [Felzenszwalb and Huttenlocher 2004].

Os algoritmos hierárquicos foram executados e a melhor partição obtida para todo o conjunto de imagens (ODS - *Optimal Data Set Scale*) [Arbelaez et al. 2009] foi utilizada. Em seguida os valores foram comparados aos demais algoritmos utilizando com métrica a medida *F-measure*. A eficiência média dos algoritmos está ilustrada na figura 8(a).

⁴<https://github.com/falreis/image-segm>

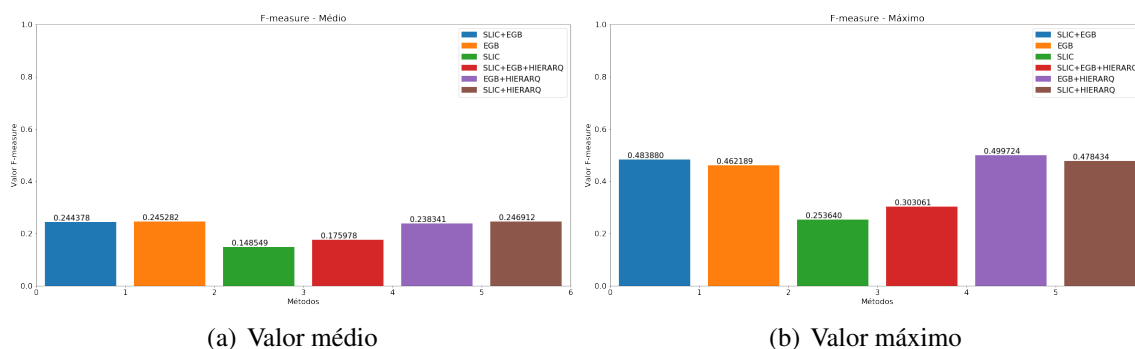


Figura 8. Valor médio e máximo de F -measure para os algoritmos.

Conforme observamos na figura 8(a), o valor médio do algoritmo SLIC+EGB foi ligeiramente superior aos valores obtidos pelo algoritmo EGB. O algoritmo EGB teve valor superior ao algoritmo SLIC executado sem hierarquização.

Os algoritmos hierárquicos, conforme figura 8(a), resultaram em valores superiores aos algoritmos sem hierarquização. O resultado observado é superior àqueles sem hierarquização. Esse resultado é previsível devido a possibilidade de melhor adequação a base de dados.

Outro ponto importante a ser observado na figura 8(a) é o desvio padrão médios das amostras. Observa-se que os resultados dos algoritmo SLIC+EGB foram mais estáveis, variando menos em relação ao algoritmo EGB, quando executado separadamente. Mais uma vez, os algoritmos nas versões com hierarquização tiveram resultados superiores aos algoritmos sem hierarquização.

Uma outra análise possível para os algoritmos são os valores máximos de F -measure. Esses valores estão ilustrados na figura 8(b). A figura mostra mais uma vez que as melhores hierarquias obtiveram resultados superiores às versões sem hierarquia. Por fim, vemos que o resultado máximo do algoritmo EGB é superior ao algoritmo SLIC+EGB, diferindo do valor médio das amostras.

Conclusão

Os algoritmos de geração de superpixels avaliados nesse trabalho tiveram resultados baixos para tarefas de segmentação e detecção de contornos, mesmo quando combinados. As versões hierárquicas dos algoritmos apresentou resultados superiores àqueles obtidos sem hierarquias.

O algoritmo SLIC, apesar das vantagens de produzir superpixels de tamanhos regulares, apresentou a menor eficiência para segmentação de imagens, em relação aos demais algoritmos estudados. A combinação dos algoritmos SLIC+EGB para detecção de contornos não apresentou resultado ligeiramente superior em relação ao algoritmo EGB utilizado separadamente. Além disso, o custo computacional para execução dos dois algoritmos em sequência, bem como os custos para recoloração da imagens mostraram-se injustificáveis frente aos ganhos obtidos.

A utilização de hierarquias após a aplicação dos métodos de geração de *superpixels* aumentou consideravelmente os resultados dos métodos. Apesar da utilização não

resultar em alto grau de desempenho na detecção de bordas e segmentação, justifica-se a utilização das versões hierárquicas em relação às versões originais dos algoritmos quando houver necessidade de *oversegmentation*. Os parâmetros descritos nesse trabalho podem ser utilizados como passo inicial para refinamento dos parâmetros em trabalhos futuros.

Os métodos hierárquicos apresentados no trabalho também permitem a utilização em diferentes cenários, principalmente para redução e agrupamento de informação relevante em imagens, atendendo aos preceitos de análise multiescala.

Como trabalho futuro sugere-se a utilização dos algoritmos mostrados nesse trabalho como passo de pré processamento de outras técnicas de segmentação, como redes neurais convolucionais. Esses passos de segmentação iniciais podem permitir treinamento mais rápido e menor quantidade de informação a ser processadas, além de identificação mais rápida de bordas e contornos.

Referências

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.
- Arbelaez, P. (2006). Boundary extraction in natural images using ultrametric contour maps. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 182–182.
- Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2009). From contours to regions: An empirical evaluation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2294–2301.
- Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916.
- Domínguez, D. and Morales, R. R. (2016). *Image Segmentation: Advances*, volume 1.
- Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.
- Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *2009 IEEE 12th International Conference on Computer Vision*, pages 670–677.
- Guimarães, S. J. F. and Patrocínio, Z. K. G. (2013). A graph-based hierarchical image segmentation method based on a statistical merging predicate. In Petrosino, A., editor, *Image Analysis and Processing – ICIAP 2013*, pages 11–20, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Liu, T., Seyedhosseini, M., and Tasdizen, T. (2016). Image segmentation using hierarchical merge tree. *IEEE Transactions on Image Processing*, 25(10):4596–4607.
- Pont-Tuset, J. and Marques, F. (2016). Supervised evaluation of image segmentation and object proposal techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(7):1465–1478.
- Rokach, L. and Maimon, O. (2005). *Clustering Methods*, pages 321–352. Springer US, Boston, MA.

Science-Dictionary (2018). What is dendrogram?

Sibson, R. (1973). Slink: An optimally efficient algorithm for the single-link cluster method. *Comput. J.*, 16:30–34.

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598.

Wang, M., Liu, X., Gao, Y., Ma, X., and Soomro, N. Q. (2017a). Superpixel segmentation: A benchmark. *Signal Processing: Image Communication*, 56:28 – 39.

Wang, X.-Y., Wu, C.-W., Xiang, K., and Chen, W. (2017b). Efficient local and global contour detection based on superpixels. *Journal of Visual Communication and Image Representation*, 48:77 – 87.

Wikipedia (2018a). Cluster analysis.

Wikipedia (2018b). Dendrogram.

Wikipedia (2018c). Hierarchical clustering.

Wikipedia (2018d). Precision and recall.

Zhang, H., Fritts, J. E., and Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260 – 280.