

# Avaliação de superpixels para segmentação de imagens

Felipe Augusto Lima Reis<sup>1</sup>

<sup>1</sup>PUC 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 superpixels SLICO e hierarquia de partições para segmentação. Utilizando **redes neurais** para segmentação, os resultados de imagens utilizando superpixels com diferentes níveis de granularidade e imagens sem tratamento foram comparadas em relação ao ground-truth. O artigo também avalia o tempo de treinamento das redes neurais para imagens com superpixels em relação às imagens originais. Para treinamento e avaliação foram utilizadas imagem do Berkeley Segmentation Data Set (BSDS500) [Arbelaez et al. 2011].*

## 1. 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], ilustrada 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.

O presente trabalho tem como objetivo investigar a utilização de superpixels como passo de pré processamento para segmentação de imagens. Esse trabalho investiga se a utilização de métodos de *oversegmentation* podem facilitar ou dificultar o processo de treinamento. O trabalho também tentará identificar se o treinamento utilizando imagens



**Figura 1. Imagem segmentada utilizando superpixel SLIC, com aproximadamente 500 segmentos.**

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, exhibe 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.

## **2. Referencial Teórico**

### **2.1. 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* [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]

### 2.1.1. 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  $3 \times 3$ , 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 5, resume as informações descrita 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  $E \leq \text{threshold}$ 

```

---

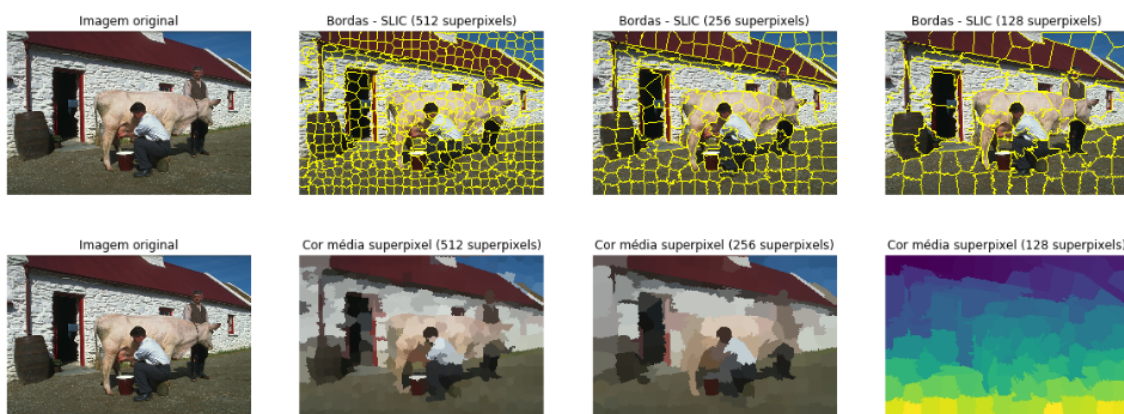
Figura 2. Algoritmo SLIC - Adaptado de [Achanta et al. 2012]

Para o algoritmo descrito na figura 5, 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 5. 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  $D = \sqrt{d_c^2 + (d_s/S)^2 * m^2}$ , 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**

### 2.1.2. Superpixels Felzenszwalb

Os superpixels Felzenszwalb e Huttenlocher 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].

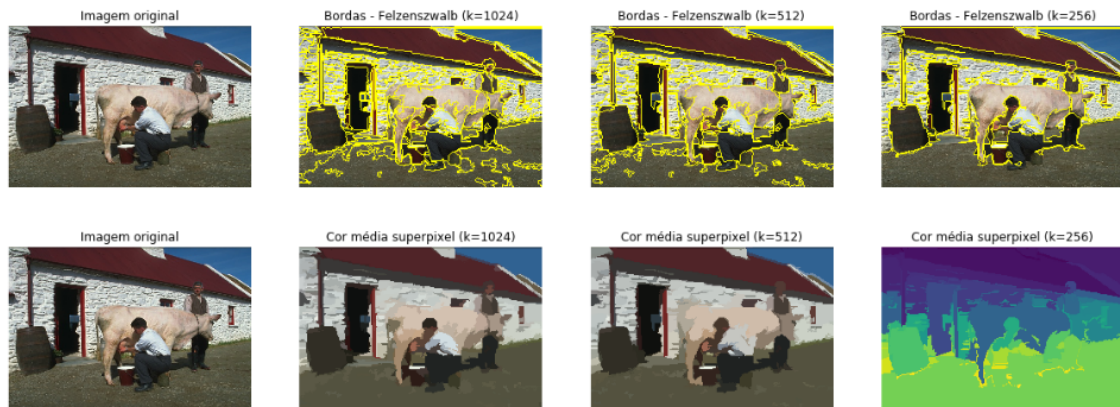
Na abordagem baseada em grafos, a segmentação  $S$  é uma partição dos vértices  $V$  em componentes, no qual cada região  $C \in S$  corresponde a um componente conectado em um grafo  $G' = (V, E')$ , onde  $E' \subseteq E$ , ou seja, a segmentação é induzida por um conjunto de vértices em arestas 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

## 2.2. 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 2018b].

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.

## 2.3. 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]

### 3. Materiais e Métodos

Para realização dos testes foram escolhidos os algoritmos de *oversegmentation* SLIC e Felzenszwalb. 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 Felzenszwalb, produz pixels irregulares, porém é mais aderente aos contornos e as bordas das imagens.

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. Em seguida, foi aplicado às imagens o método de Felzenszwalb.

Após a execução dos algoritmos, foram gerados *clusters* com as correlações entre os grupos, utilizando o algoritmo SLINK (*single-linkage*). Os cortes foram realizados em diversos níveis da hierarquia e os resultados foram avaliados manualmente, a fim de encontrar bons parâmetros para o algoritmo.

O método descrito acima foi aplicado à base de dados BSD500 e comparados com as imagens manualmente segmentadas (*ground-truth*). Para avaliação das imagens, foi utilizada a classificação visual, comparando a qualidade dos resultados e o método de avaliação SEISM, que fornece informações supervisionadas sobre a qualidade da segmentação.

Os algoritmos SLIC e Felzenszwalb utilizados para confecção desse trabalho foram obtidos pela biblioteca Scikit-Image <sup>1</sup>. O algoritmo de geração de cluster SLINK foi obtido na biblioteca Scipy.org <sup>2</sup>. Os códigos fontes gerados estão disponíveis publica-

---

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

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



mente no Github <sup>3</sup>.

## 4. Testes, Resultados e Discussões



**Figura 5. Fronteiras das Hierarquia de partições utilizando superpixel SLICO.**

## 5. Conclusão

### 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., 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.
- 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.
- 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.

<sup>3</sup><https://github.com/falreis/image-segm>

- 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). Hierarchical clustering.
- 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.