Segmentação Semântica

Abdullah Zaiter
15/0089392
abdullah.zaiter@gmail.com
Ian Moura Alexandre
15/0129661
ianzeba@gmail.com

Departamento de Ciência da Comptutação
Universidade de Brasília
Campus Darcy Ribeiro, Asa Norte
Brasília-DF, CEP 70910-900, Brazil,

Abstract

[1] [2] Este documento apresenta os fundamentos teóricos, a metologia e os resultados obtidos para a implementação de dois algoritmos de segmentação semântica para a resolução do desafio PASCAL VOC2007 Visual Segmentation Taster, sendo uma implementação utilizando uma CNN do tipo Xception65 do deeplabv3, e a outra utilizando-se de matrizes de coocorrência e variáveis de textura.

1 Introdução

A segmentação é a divisão de uma imagem em diferentes regiões e categorias à partir de uma semelhança de características entre seus píxels[10], tendo como objetivo sua decomposição e mudar sua forma de representação, de maneira a organizar as informações desejadas para análise. As imagens possuem diversos tipos de informações, sendo elas cores, iluminação, descritores de textura(energia, ASM, homogeneidade), dentre outras, das quais podem ser possíveis traçar similaridades entre os píxels vizinhos, separando-os em um limitado número de categorias. Existem várias formas de segmentação de imagens, sendo as principais categorias: por **thresholding**, **métodos por detecção de bordas** e **método baseado em regiões**, podendo ser eles supervisionados ou totalmente automático.Na imagem 1, pode-se ver quatro tipos de segmentações feitas em uma imagem, utilizando os métodos de Felzenszwalb[7], SLIC (*Simple Linear Iterative Clustering*)[9], Quickshift e Compact Watershed[7], como mostrado no tutorial em [6].

No algoritmo realizado, utilizou-se o método SLIC, que utliza-se de K-means em um espaço 5-D com informações de cor e região para poder dividir a imagem. Obtidos os superpíxels, para se obter as características necessárias da imagem, utilizou as descrições de textura. A textura pode ser observada como um padrão estatístico observado sobre uma determinada região de píxels. Uma das maneiras de se trabalhar com texturas é utlizando-se de matrizes de co-ocorrência (GLCM)[8], que descrevem a estrutura e os padrões observados de brilho de píxel da imagem em escala de cinza. Obtida a GLCM e realizando-se a normalização, dividindo cada elemento pela soma de toda a matriz, pode-se obter os descritores de textura, como **energia, contraste, dissimilaridade, homogeneidade, correlação** e **entropia**,

[1] Abdullah: Desenvolveu a parte de segmentação semântica utilizando a deeplab e Xception65. No relatório, participou da escrita da introdução metodologia, dos resultados e da conclusão.

[2] Ian: Desenvolveu o método de segmentação semântica por análise de textura. No relatório, participou da escrita do abstract, da introdução e parte da metodologia, resultados e conclusão.

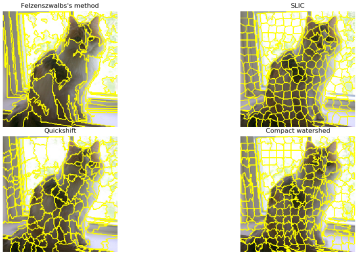

Figure 1: Quatro métodos de segmentações utilizados em uma imagem

mostrados em (1), (2) e (3). Com as informações obtidas de cada superpíxel de várias imagens de seus diferentes labels, torna-se possível classificar outras imagens, criando-se suas regiões segmentadas semanticamente.

$$ENE = \sum_{i,j=0}^{G-1} [Pd, \theta(i,j)]^2 \qquad CON = \sum_{i,j=0}^{G-1} (I-J)^2 Pd, \theta(i,j) \tag{1}$$

$$DIS = \sum_{i,j=0}^{G-1} |i-j| Pd, \theta(i,j) \qquad HOM = \sum_{i,j=0}^{G-1} \frac{[Pd, \theta(i,j)]}{1+(i-j)^2} \tag{2}$$

$$COR = \frac{\sum_{i,j=0}^{G-1} (i-\mu_x)(j-\mu)[P_d, \theta(i,j)]}{\sigma_x \sigma_y} \qquad ENT = -\sum_{i,j=0}^{G-1} P_{d,\theta(i,j)Ln}\left[P_{d,\theta(i,j)}\right] \tag{3}$$

A outra implementação utilizada neste artigo utiliza-se da arquitetura de rede neural convolucional Xception65[4], sendo o estado da arte para o desafio PASCAL VOC 2012. A arquitetura apresentada utiliza a DeepLabV3+[3] como um módulo de encoder e um módulo simples de decodificação para obter melhores as segmentações.

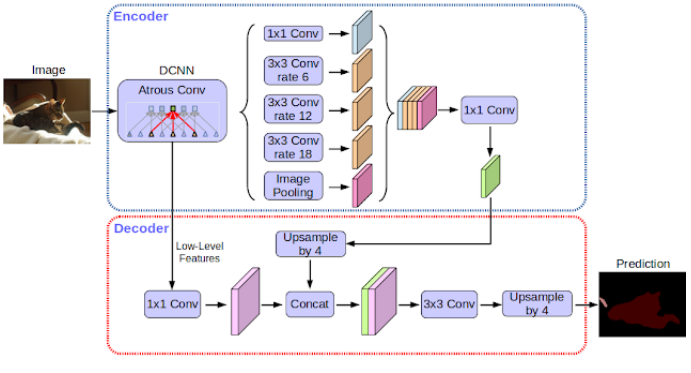

Figure 2: Arquitetura do DeepLabV3+

Como pode ser visto na figura 2, a arquitetura desse modelo tem uma parte de **codificação (encoder) que utiliza para conseguir distinguir as classes nas imagens, e após isso tem a parte de decodificação (decoder) que serve para marcar as regiões das imagens que se encontram as classes identificadas pela fase inicial da rede (encoder)**.
Algoritmos de aprendizado de máquina atualmente estão sendo amplamente utilizadas nos testes de bancos de dados, apresentando resultados muito superiores do que as segmentações implementadas à mão, já que elas aprendem não somente as texturas de cada região separadamente, mas faz uma análise profunda de cada píxel e o contexto onde está inserido

dentro da imagem, tendo uma capacidade maior de resolver problemas não-lineares dessa natureza, estabelecendo uma ligação mais forte entre os componentes presentes na imagem. Neste documento, deseja-se mostrar melhor o comparativo entre estas duas implementações.

2 Metodologia

Para as duas implementações realizadas, foi realizado uma estratégia de extração dos labels de cada imagem. Do banco de dados do Pascal VOC 2007 utiliza-se no código as pasta *Annotations*, contendo as informações em arquivo xml de cada imagem contida no banco; *ImageSets*, com os arquivos texto contendo o nome das imagens de treinamento e de teste,*JPEGImages*, com as imagens do banco de dados no formato jpg, e *SegmentationClass*, com o ground truth de segmentação das imagens, sendo cada cor referente a uma determinada classe de objetos.

2.1 Segmentação por Análise de Textura

Tendo as imagens de treinamento, realiza-se a captura das variáveis de textura para cada imagem, de cada uma das categorias descritas. Para isso, primeiramente realiza-se sobre a imagem inteira uma filtragem, que realiza uma segmentação na imagem por thresholding para se obter as regiões de maior interesse da imagem. Em seguida, desta imagem processada retira-se a GLCM e dela retira-se os descritores de textura[3], mostradas na seção 1. Armazena-se assim os descritores juntamente com a label que está sendo mostrada na figura. Feito isto com todas as imagens, realiza-se o teste, aplicando-se uma segmentação do tipo SLIC em cada imagem, para obter-se superpíxels e, destes superpíxels, calcula-se os descritores de textura e classifica-se com dois tipos de classificadores do tipo svm, o SVC e o Random Forest.

2.2 DeepLab

Foi utilizado o Framework Tensorflow e baseou-se na implementação [6] feita pela equipe da equipe do framework.
Para o melhor desempenho do código, a base de dados para o conjunto de treinamento foi transformada para o formato .tfrecords, os dados nesse formato são armazenados como strings sequenciais binarias. O modelo utilizado por eles e o repositório, é feito para alguns bancos de dados como coco, cityscapes e VOC2012. Para poder utilizar os modelos propostos pela equipe no banco de dados VOC2007, foram feitas algumas alterações, nos scripts, diretórios de busca dos dados e os tamanhos de entrada da rede neural, além de adaptar o código para retornar as saídas como imagen de segmentação onde cada pixel possui um valor entre 0 e 20 que representa qual classe que é, Isto foi feito para que seja possível a utilização do Script de avaliação proposto pelo professor e monitor da disciplina.
Todos os treinamentos, validações e visualizações foram realizados usando uma GPU Nvidia 1070.
Foram feitos alguns experimentos diferentes que servirão para comparação:

- Usando pesos pre-trainados na base de dados Pascal VOC 2012, conjunto trainval

[3] Com exceção da entropia. Ao invés dela, utiliza-se o segundo momento angular.

- Usando pesos pre-trainados na base de dados Pascal VOC 2012, conjunto val
- Treinando os pesos a partir de valores aleatórios por 3000 iterações diretamente na base de dados Pascal VOC 2007 conjunto trainval
- Treinando os pesos a partir de valores aleatórios por 3000 iterações diretamente na base de dados Pascal VOC 2007 conjunto val

A acurácia é calculada utilizado foi usando a nova maneira proposta pelos organizadores do desafio, **IOU**- (*Intersection Over Union*), que pode ser visualizada na figura 3:

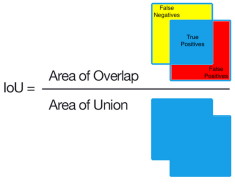

Figure 3: Intersecção sobre união

3 Resultados

3.1 Segmentação por Análise de Textura

Os resultados coletados foram feitos por meio de dois testes, sendo um utilizando o método explicitado na seção 2, e o outro aplicando-se a matriz GLCM sobre o objeto contido na bounding box dada nas anotações do banco de dados. Cada resultado foi analisado com os dois classificadores: o C-Support Vector Classification e o classificador Random Forest. Os resultados utilizando-se o classificador Random Forest pode ser visualizado na tabela 1.

(a) (b) (c)

Figure 4: Segmentação por comparação de textura: Imagem Original (a), Segmentação no primeiro teste (b) e segmentação obtida no segundo teste (c)

Utilizando-se o classificador SVC obteve-se para os dois casos a acurácia média de **4.1106%** e **4.7619%**. Entretanto, no primeiro caso, a maioria dos píxels foi rotulado como pessoa (acurácia de 82.5681%) e background (acurácia de 3.7560%), e no segundo caso, todos os píxels foram rotulados como pessoa, apresentando uma acurácia de 100% para classe pessoa e 0% para todas as outras. No segundo teste, rotular todos os píxels como pessoa acertou 6.0862%, enquanto que o método mostrado na 1 acertou 4.8920%. No primeiro teste, utilizando-se o método de Random Forest acertou 9.7406%. Um dos motivos para ser tão baixo esse índice deve-se ao fato de que não se houve treinamento no teste 2 para classes

Class	Accuracy
background:	9.44000%
aeroplane:	0.0%
bicycle:	0.0%
bird:	5.95795%
boat:	41.9524%
bottle:	0.0%
bus:	0.0%
car:	0.0%
cat:	0.0%
chair:	0.0%
cow:	17.0101%
diningtable:	0.0%
dog:	1.64191%
horse:	0.0%
motorbike:	0.0%
person:	38.9015%
pottedplant:	0.0%
sheep:	0.0%
sofa:	0.0%
train:	0.0%
tvmonitor:	0.0%
Avg:	**5.4716%**

Class	Accuracy
background:	0.0%
aeroplane:	9.96205%
bicycle:	4.49395%
bird:	0.0%
boat:	0.0%
bottle:	0.0%
bus:	11.8210%
car:	2.16393%
cat:	11.9646%
chair:	15.4233%
cow:	0.33692%
diningtable:	10.4373%
dog:	2.34252%
horse:	1.00269%
motorbike:	1.54168%
person:	60.0690%
pottedplant:	0.05848%
sheep:	11.1332%
sofa:	0.0%
train:	0.33686%
tvmonitor:	8.99616%
Avg:	**7.2421%**

Table 1: Segmentação por análise de textura utilizando o Random Forest Classifier - utilizando o threshold OTSU (esquerda) e utilizando as bounding boxes (direita)

.

background, pois esperava-se que seria segmentado todos os píxels como plano de fundo. Na classificação, para gerar os superpíxels, foi-se testado utilizando o método SLIC para gerar os superpíxels de avaliação, com 100 segmentos e sigma (comprimento do kernel de suavização Gaussiana) igual a 5, entretanto vale-se a tentativa de testar os outros métodos, principalmente o Felzenszwalb, que atualmente vem sendo bastante implementado, sendo um melhor generalizador.

3.2 DeepLab

Foram coletados os resultados para todos os tipos experimentos mencionados em cima:

Figure 5: Segmentação de imagem com a rede pretreinada no VOC2012

Figure 6: Segmentação de imagem com a rede treinada completamente no VOC2007

Figure 7: Imagem segmentada no formato que permite a avaliação pelo código do monitor

Class	Accuracy
background:	90.6344%
aeroplane:	88.2322%
bicycle:	70.6523%
bird:	73.3234%
boat:	73.4655%
bottle:	81.0744%
bus:	91.5807%
car:	81.3056%
cat:	89.6335%
chair:	65.8032%
cow:	79.3407%
diningtable:	85.0731%
dog:	74.2352%
horse:	64.0865%
motorbike:	82.3067%
person:	89.3504%
pottedplant:	47.8648%
sheep:	86.7603%
sofa:	66.6477%
train:	96.4708%
tvmonitor:	70.3976%
Avg:	**77.9490%**

Class	Accuracy
background:	90.5481%
aeroplane:	84.1272%
bicycle:	66.0642%
bird:	69.3998%
boat:	69.4431%
bottle:	77.0499%
bus:	96.5273%
car:	77.3462%
cat:	85.1277%
chair:	61.3272%
cow:	75.5208%
diningtable:	81.4731%
dog:	70.0618%
horse:	60.2963%
motorbike:	78.3909%
person:	84.6066%
pottedplant:	43.3831%
sheep:	82.0489%
sofa:	62.0179%
train:	92.3684%
tvmonitor:	66.8014%
Avg:	**74.1126%**

Table 2: Rede pre-treinada no banco VOC2012 - conjunto trainval (esquerda) e conjunto val (direita)

4 Conclusão

Os resultados mostrados apresentados ratificam o que foi apresentado na 1, onde os resultados obtidos com machine learning foram muito melhores do que o obtido com a implemen-

Classe	Acurácia
background:	85.0812%
aeroplane:	60.1414%
bicycle:	51.9711%
bird:	43.7158%
boat:	52.7624%
bottle:	50.9888%
bus:	70.3693%
car:	49.9140%
cat:	63.5079%
chair:	38.7517%
cow:	56.8232%
diningtable:	59.6392%
dog:	44.0131%
horse:	37.9425%
motorbike:	60.3082%
person:	50.0359%
pottedplant:	21.8332%
sheep:	61.3359%
sofa	13.1801%
train:	66.8393%
tvmonitor:	51.9289%
Avg:	**51.9563%**

Classe	Acurácia
background:	84.5481%
aeroplane:	59.1270%
bicycle:	41.0642%
bird:	44.3998%
boat:	44.4431%
bottle:	52.0499%
bus:	71.5273%
car:	52.3462%
cat:	60.1277%
chair:	36.3272%
cow:	50.5208%
diningtable:	56.4731%
dog:	45.0618%
horse:	35.2963%
motorbike:	53.3909%
person:	49.6066%
pottedplant:	18.3831%
sheep:	57.0489%
sofa:	37.0179%
train:	67.3684%
tvmonitor:	41.8014%
Avg:	**50.3776%**

Table 3: Rede treinada completamente no banco VOC2007 - conjunto trainval (esquerda) e conjunto val (direita)

tação sem aprendizagem. Entretanto, alguns dos pontos podem ser ressaltados na implementação 1:

- Para um bom resultado, é fundamental a utilização de um método capaz de analisar o contexto onde a textura foi inserida e não apenas o superpixel isolado. Um dos métodos que podem ser utilizados é o CRF (*Condition Random Fiels*)[1], que pode considerar este tipo de relação antes de ser feita a predição.
  Visualizou-se que o método de classificação por superpíxel foi comprometido para o classificador C-Support Vector Classifier, não conseguindo ser um bom segmentador. O Random Forest atendeu dentro do esperado, apesar do baixo desempenho, entretanto reforçando somente a discrepância obtida entre o segmentador utilizando *machine learning* e o método clássico.

- Concluiu-se que o uso de uma rede neural com os pesos pre-treinados num banco de dados mais sofisticado resulta em uma capacidade maior de generalização e assim uma acurácia maior. Isso foi notado tendo em vista que o resultado do mesmo modelo que foi treinado no banco de dados VOC2012 (maior e mais complexo) mesmo sendo aplicado no banco de dados VOC2007 resultou em uma acurácia consideravelmente mais alta do que a o mesmo modelo treinado completamente no mesmo banco de dados VOC2007

References

[1] Anurag Arnab, Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Måns Larsson, Alexander Kirillov, Bogdan Savchynskyy, Carsten Rother, Fredrik Kahl, and Philip Hilaire Sean Torr. Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 35:37–52, 2018.

[2] S. Beucher. The watershed transformation, may 2010. http://www.cmm.mines-paristech.fr/~beucher/wtshed.html.

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. URL http://arxiv.org/abs/1706.05587.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018. URL http://arxiv.org/abs/1802.02611.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. https://github.com/tensorflow/models/tree/master/research/deeplab.

[6] Scikit-Image development team. Comparison of segmentation and superpixel algorithm, 2019. https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_segmentations.html#sphx-glr-auto-examples-segmentation-plot-segmentations-py.

[7] P. Felzenszwalb e D. Huttenlocher. Efficient graph-based image segmentation. *Int'l J. Computer Vision*, 59(2), sep 2004.

[8] Mryka Hall-Beyer. Glcm texture: a tutorial. 06 2019.

[9] Jay Rambhia. Slic based superpixel segmentation, aug 2013. https://jayrambhia.com/blog/superpixels-slic.

[10] Linda Shapiro and George Stockman. *Computer Vision*. 2000. Ch. 10: Image Segmentation, http://nana.lecturer.pens.ac.id/index_files/referensi/computer_vision/Computer%20Vision.pdf.