

Aplicação para demarcação de bordas em lesões dermatoscópicas

Pedro Vinícius Macêdo de Araújo

Curso de Ciência da Computação – Universidade Federal do Maranhão (UFMA) –
Campus do Bacanga
(98) 3272 8000 – São Luís – MA – Brasil

pedro.vma96@gmail.com

***Abstract.** Precise border detection is often the first step in dermoscopy lesion analysis, as various clinically interesting features like lesion symmetry and extraction of clinical features depends of a precise border detection. In this paper, the performance of various thresholding algorithms will be evaluated in a task of dermoscopy lesion segmentation.*

***Resumo.** A detecção precisa de bordas é o primeiro passo para a análise de lesões dermatoscópicas, já que várias características de interesse clínico como simetria da lesão e extração de atributos clínicos dependem criticamente de uma detecção precisa da borda. Neste trabalho será analisado o desempenho de vários algoritmos de limiarização na tarefa de segmentar lesões dermatoscópicas.*

1. Introdução

Dermatoscopia é uma técnica não invasiva de obtenção de imagens que utiliza ampliação óptica e imersão líquida fazendo estruturas abaixo da superfície da pele se tornarem mais facilmente visíveis quando comparadas com imagens clínicas tradicionais. A dermatoscopia permite a identificação de várias características morfológicas como redes de pigmentos, glóbulos e manchas. Isso reduz erros de triagem, porém foi demonstrado que a dermatoscopia pode diminuir a precisão do diagnóstico nas mãos de dermatologistas inexperientes. Assim, para diminuir esses erros, é necessário o desenvolvimento de técnicas de análise computadorizadas.

Detecção automática de bordas é geralmente o primeiro passo na análise automática ou semiautomática de imagens dermatoscópicas. Uma detecção precisa é crucial para um analista e fornece informações importantes. Características clínicas como redes de pigmentos, glóbulos e áreas de cores atípicas dependem de uma detecção precisa da borda. A detecção automática é uma tarefa que apresenta problemas devido a diversos fatores: baixo contraste entre a lesão e a pele em volta, bordas irregulares e embaçadas, artefatos na imagem como bordas pretas, cabelo sobre a lesão e vasos sanguíneos, coloração variada dentro da lesão entre outros fatores.

A limiarização de imagens, onde um objeto é extraído do plano de fundo de uma imagem de entrada é uma das aplicações mais comuns em análise de imagens. Entre os métodos de limiarização, a limiarização em dois níveis separa os pixels de uma imagem em duas regiões (o objeto e o fundo); uma região contém os pixels com valores de cinza menores que o limiar (*threshold*) e a outra contém pixels com valores de cinza maior que o *threshold*. Geralmente, o *threshold* é encontrado no vale mais profundo do

histograma, porém quando o vale não é evidente, é muito difícil encontrar um valor de *threshold* apropriado.

Neste trabalho será analisado o desempenho de vários algoritmos de limiarização, com enfoque especial em uma implementação do algoritmo *threshold fusion* de M. Emre Celebi *et al.*

2. Descrição dos algoritmos utilizados

2.1. Huang & Wang (*Image thresholding by minimizing the measures of fuzziness*)

Este algoritmo utiliza o conceito de conjunto *fuzzy* para obter um valor de *threshold*. Dado um certo valor de *threshold*, a função de pertinência de um pixel (função que determina se o pixel pertence ao objeto a ser destacado ou ao plano de fundo) é definida como a diferença absoluta entre o seu valor de cinza e o valor médios das áreas ao redor (isto é, o objeto ou o plano de fundo). Depois de calculadas as funções de pertinência, uma medida *fuzzy* pode ser minimizada para obter um valor de *threshold* ótimo.

2.2. Kittler & Illingworth (*Minimum error thresholding*)

Este algoritmo considera a limiarização como um problema de classificação. Ou seja, se a distribuição dos níveis de cinza do objeto e do plano de fundo são conhecidas, então o *threshold* de erro mínimo pode ser obtido utilizando métodos de decisão estatística. A ideia do algoritmo é otimizar a função de custo relacionada com a taxa média de erro de classificação do pixel.

2.3. Kapur *et al.* (*A new method for gray-level picture thresholding using the entropy of the histogram*)

O algoritmo opera calculando a entropia do histograma. Calculando a entropia, maximiza-se o valor e é obtido uma quantidade igual de pixels brancos e pretos. Com isso é possível obter uma função de avaliação, que pode ser maximizada para o obter o valor ótimo de *threshold*.

2.4. Otsu

O algoritmo assume que a imagem contém duas classes de pixels: pixels do plano de fundo e do primeiro plano. A partir do histograma, o algoritmo calcula o *threshold* separando as duas classes de pixels para que sua variância intra-classe seja mínima ou sua variância inter-classe seja máxima. O algoritmo envolve iterar sobre todos os valores de *threshold* possíveis e calcular a medida de variância intra-classe para os pixels de ambos os lados do *threshold* (pixels no plano de fundo ou no primeiro plano). O objetivo é encontrar um *threshold* que tenha uma soma de variância intra-classe mínima.

3. Threshold Fusion

3.1. Introdução

Na maioria das imagens dermatoscópicas, a lesão pode ser precariamente separada da pele em volta por meio de um método de limiarização aplicado no canal azul da imagem. No entanto, a eficácia de um determinado método depende de características particulares da imagem. Um possível método de superar esse problema é fundir os resultados de vários métodos de limiarização diferentes. Assim é possível aproveitar o melhor de cada método de limiarização independente das características da imagem e

obter decisões finais mais robustas que o possível com apenas um método de limiarização individual.

3.2. Definição

O algoritmo é definido da seguinte forma:

Seja $X = \{x_{mn} ; m = 0,1, \dots, M-1, n = 0,1, \dots, N-1\}$ a imagem original com dimensões M por N com L possíveis níveis de cinza e $Y = \{y_{mn} ; m = 0,1, \dots, M-1, n = 0,1, \dots, N-1\}$ o resultado binário da fusão de limiarização. Seja um conjunto de P métodos de limiarização. Seja T_i e A_i ($i = 1,2, \dots, P$) o valor do *threshold* e a imagem binária associada com o i -ésimo algoritmo de limiarização, respectivamente. Dentro de um quadro de Campo Aleatório de Markov, o problema de fusão pode ser definido como um problema de minimização de energia. Assim, a energia U_{mn} a ser minimizada para um pixel (m,n) é:

$$U_{mn} = \beta_{SP} \cdot U_{SP} [y_{mn}, Y^S(m, n)] + \sum_{i=1}^P \beta_i \cdot U_{II} [y_{mn}, A_i^S(m, n)]$$

Onde S é uma janela 3 por 3 associada com o pixel (m,n) . U_{SP} e U_{II} são as funções espacial e inter-imagem respectivamente. E B_{SP} e B_i ($i = 1,2,\dots, P$) representam os parâmetros espacial e inter-imagem respectivamente. A função de energia espacial pode ser expressa como:

$$U_{SP} [y_{mn}, Y^S(m, n)] = - \sum_{y_{pq} \in Y^S(m, n)} I(y_{mn}, y_{pq})$$

Onde I é a função indicadora (retorna 1 se os dois valores forem iguais ou 0 caso contrário). A função de energia inter-imagem é definida como:

$$U_{II} [y_{mn}, A_i^S(m, n)] = - \sum_{A_i(p, q) \in A_i^S(m, n)} \alpha^i(x_{pq}) \cdot I[y_{mn}, A_i(p, q)]$$

α^i é uma função-peso definida como:

$$\alpha^i(x_{mn}) = 1 - \exp(-\gamma |x_{mn} - T_i|)$$

3.3. Descrição da implementação

A estratégia definida pelo artigo é a que segue:

1. Aplicar cada algoritmo de limiarização para gerar o conjunto de imagens;
2. Inicializar Y minimizando para cada pixel a função de energia local U_{mn} , sem o termo de energia espacial ($B_{SP} = 0$);
3. Atualizar Y minimizando para cada pixel a função de energia local U_{mn} incluindo o termo de energia espacial ($B_{SP} \neq 0$);
4. Repetir o passo (3) k vezes ou até o número de formas diferentes em Y nas duas últimas iterações ser insignificante.

Como sugerido pelo artigo, o passo 3 apresenta baixa performance e não trás contribuições significativas para o desempenho do algoritmo. O parâmetro γ , como recomendado, foi definido como 0.1. Diferentemente do aconselhado pelo artigo, depois

de ter sido gerada uma imagem com os valores de cada pixel, o *threshold* foi definido como o valor médio dos valores de cinza da imagem.

4. Resultados

A precisão dos resultados dos algoritmos foi analisada utilizando o coeficiente de Dice. O coeficiente de Dice é uma estatística para comparar a similaridade de duas amostras, sendo ideal para comparar resultados de algoritmos com máscaras de referência em cenários de segmentação de imagens. As 375 imagens utilizadas para testar os algoritmos são provenientes da base de imagens ISIC 2016, composta de várias imagens dermatoscópicas e de suas respectivas máscaras de referência (que determinam a lesão precisamente destacada).

A Tabela 1 abaixo mostra o coeficiente de Dice médio de cada algoritmo depois de ser aplicado nas 375 imagens da base de dados.

Tabela 1. Resultados

Algoritmo	Média	Desvio padrão
Huang & Wang	0.624128	0.361586
Kittler & Illingworth	0.483177	0.419018
Kapur <i>et al.</i>	0.633653	0.340096
Otsu	0.668652	0.316972
Threshold Fusion	0.37632	0.27337

As tabelas 2 e 3 mostram a quantidade de imagens sobre e abaixo um determinado limiar do coeficiente de Dice

Tabela 2. Resultados com limiar 0.5

Algoritmo	≥ 0.5	< 0.5
Huang & Wang	275	100
Kittler & Illingworth	206	169
Kapur <i>et al.</i>	274	101
Otsu	297	78
Threshold Fusion	121	254

Tabela 3. Resultados com limiar 0.9

Algoritmo	≥ 0.9	< 0.9
Huang & Wang	88	287

Kittler & Illingworth	81	294
Kapur <i>et al.</i>	97	278
Otsu	89	286
Threshold Fusion	8	367

5. Análise dos resultados

Analisando os resultados, observa-se que o algoritmo de Otsu obteve a maior média geral de precisão em encontrar uma borda precisa, dentre todos os algoritmos estudados. Isso é confirmado pelo maior índice de resultados com coeficiente de Dice ≥ 0.5 . Porém, o algoritmo *Kapur et al.* obteve um desempenho melhor em encontrar bordas com precisão superior a 0.9, indicando que este algoritmo tem desempenho mais elevado em casos particulares. Devido à uma implementação diferente da aconselhada pelo artigo do *threshold fusion*, os resultados do algoritmo foram insatisfatórios e divergentes dos apresentados no artigo.

5.1. Huang & Wang

Exemplo de imagem com alto coeficiente de Dice:



Figura 1. Exemplo de lesão destacada pelo algoritmo Huang & Wang com coeficiente de Dice 0.98 (direita) comparada com a lesão fornecida pela base de dados (centro).

Exemplo de imagem com baixo coeficiente de Dice

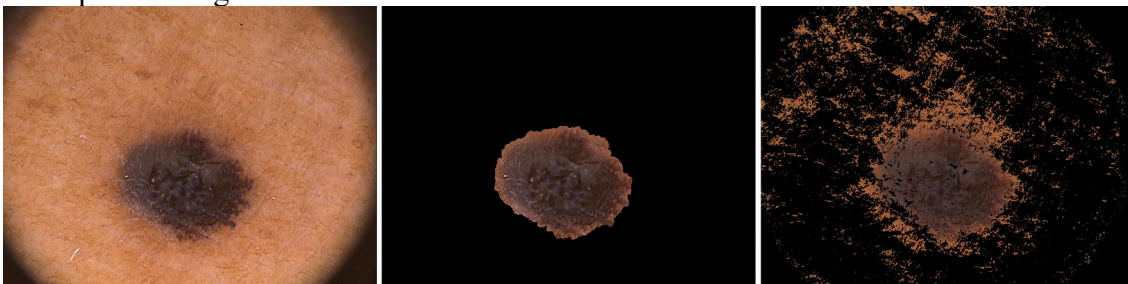


Figura 2. Exemplo de lesão destacada pelo algoritmo Huang & Wang com coeficiente de Dice 0.64 (direita) comparada com a lesão fornecida pela base de dados (centro).

O algoritmo *Huang & Wang* utiliza conceitos de lógica difusa com o objetivo de particionar as imagens em regiões de interesse minimizando as medidas de *fuzzy*. Na primeira imagem o alto contraste entre pele a lesão e uma borda da lesão bem definida garantiram uma demarcação precisa, enquanto no segundo caso, a presença de cantos escuros na imagem prejudicaram o resultado do algoritmo, apesar da lesão ser bem definida e ter bom contraste com a pele.

5.2. Kittler & Illingworth

Exemplo de imagem com alto coeficiente de dice

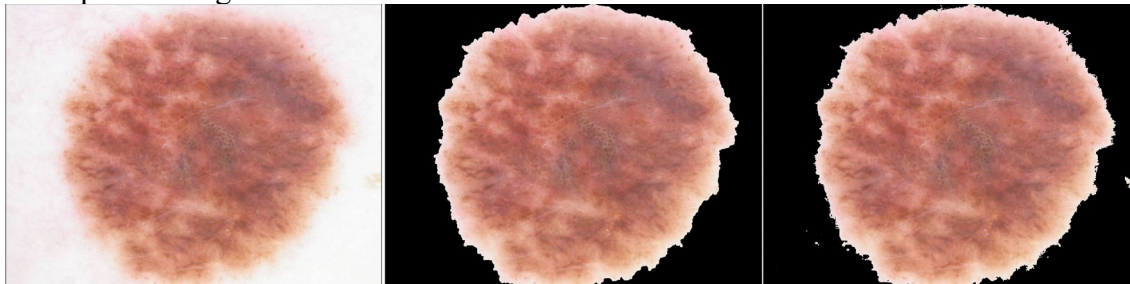


Figura 3. Exemplo de lesão destacada pelo algoritmo Kittler & Illingworth com coeficiente de Dice 0.98 (direita) comparada com a lesão fornecida pela base de dados (centro).

Exemplo de imagem com baixo coeficiente de dice



Figura 4. Exemplo de lesão destacada pelo algoritmo Kittler & Illingworth com coeficiente de Dice 0.39 (direita) comparada com a lesão fornecida pela base de dados (centro).

O algoritmo *Kittler & Illingworth* usa uma estimativa da distribuição dos níveis de cinza dos pixels do fundo e do primeiro plano da imagem para calcular um *threshold* de erro mínimo usando teoria da decisão estatística. Assim, no primeiro caso o alto contraste da lesão com a pele facilita uma detecção precisada borda e no segundo exemplo, a coloração similar entre a pele e a lesão acaba por destacar várias áreas na pele que possuem coloração intensa similar à lesão.

5.3. Kapur *et al.*

Exemplo de imagem com alto coeficiente de dice

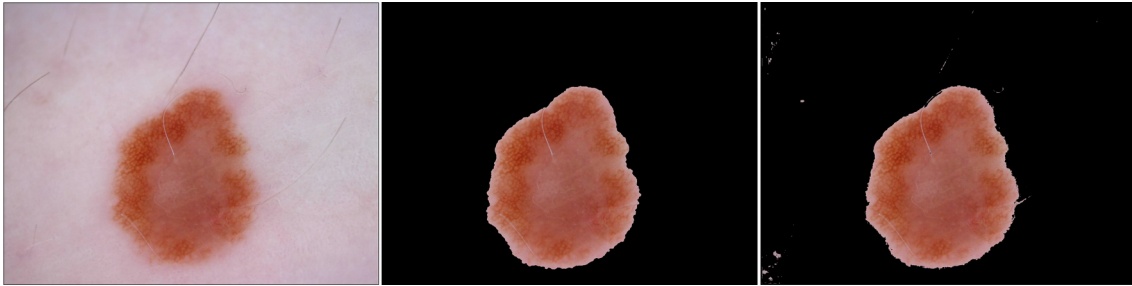


Figura 5. Exemplo de lesão destacada pelo algoritmo Kapur *et al.* com coeficiente de Dice 0.98 (abaixo) comparada com a lesão fornecida pela base de dados (acima).

Exemplo de imagem com baixo coeficiente de dice



Figura 6. Exemplo de lesão destacada pelo algoritmo Kapur *et al.* com coeficiente de Dice 0.39 (direita) comparada com a lesão fornecida pela base de dados (centro).

Assim como no algoritmo *Kittler & Illingworth*, o alto contraste e a borda bem definida garantem uma detecção precisa (apesar de alguns pontos na pele de coloração intensa serem perceptíveis). Já no segundo caso, a lesão contém bordas e texturas extremamente irregulares, além de cantos pretos, fazendo com que o algoritmo só destaque as partes de coloração muito intensa.

5.4. Otsu

Exemplo de imagem com alto coeficiente de dice



Figura 7. Exemplo de lesão destacada pelo algoritmo de Otsu. com coeficiente de Dice 0.98 (direita) comparada com a lesão fornecida pela base de dados (centro).

Exemplo de imagem com baixo coeficiente de dice

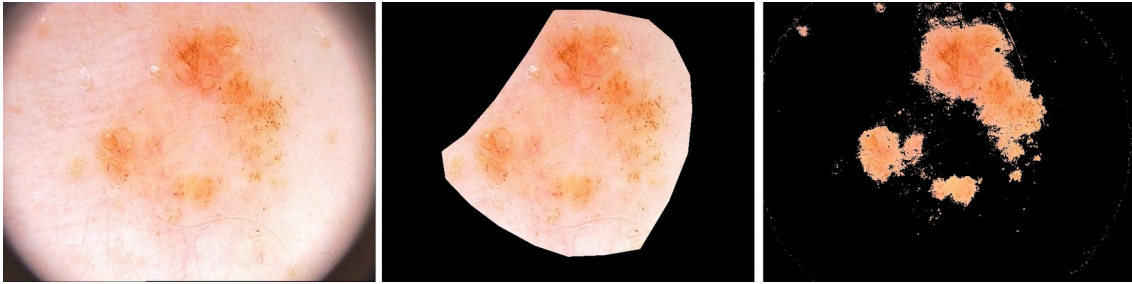


Figura 8. Exemplo de lesão destacada pelo algoritmo de Otsu. com coeficiente de Dice 0.39 (direita) comparada com a lesão fornecida pela base de dados (centro).

No caso do algoritmo de Otsu, este espera que a imagem contenha duas classes de pixels (uma para o fundo e outra para o primeiro plano da imagem) e busca um limiar ótimo para separar essas duas regiões, no caso, a pele e a lesão respectivamente. No primeiro caso observa-se que o algoritmo obteve sucesso em destacar a lesão da pele, devido ao alto contraste entre a pele e a lesão. Já no segundo caso, a lesão tem formato e coloração bastante irregular, como resultado, o algoritmo só destacou as áreas de tons mais escuros.

5.5. Threshold fusion

Exemplo de imagem com alto coeficiente de dice

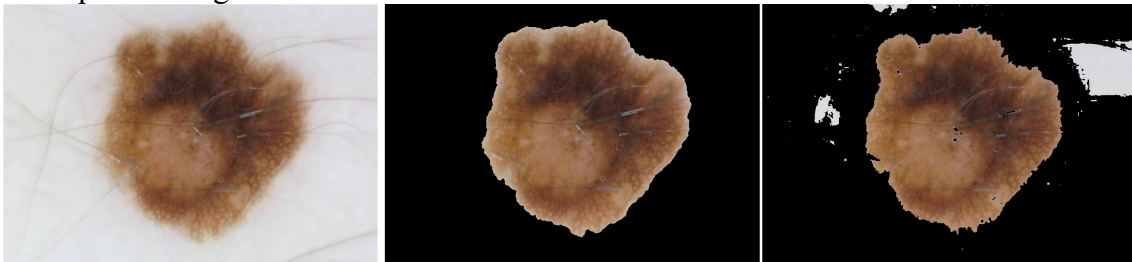


Figura 9. Exemplo de lesão destacada pelo algoritmo de Threshold Fusion. com coeficiente de Dice 0.83 (direita) comparada com a lesão fornecida pela base de dados (centro).

Exemplo de imagem com baixo coeficiente de dice



Figura 10. Exemplo de lesão destacada pelo algoritmo Threshold Fusion. com coeficiente de Dice 0.35 (direita) comparada com a lesão fornecida pela base de dados (centro).

O algoritmo de *threshold fusion* apresentou um desempenho abaixo do esperado e um resultado divergente dos apresentados pelo artigo, devido à uma implementação diferente da sugerida pelo artigo.

6. Conclusões

Neste artigo foi feita uma análise do desempenho de vários algoritmos de limiarização na tarefa de segmentar um conjunto de imagens, com um enfoque numa tentativa de implementação do algoritmo *threshold fusion*.

Referências

M. Emre Celebi; Quan Wen ; Sae Hwang ; Hitoshi Iyatomi ; Gerald Schaefer. Lesion Border Detection in Dermoscopy Images Using Ensembles of Thresholding Methods- December 31, 2013

M. Emre Celebi ; Hitoshi Iyatomi; Gerald Schaefer; William V. Stoecker. Lesion border detection in dermoscopy images- 13 November 2008

Liang-Kai Huang; Mao-Jiun J. Wang. Image Thresholding by Minimizing the Measures of Fuzziness – April 20, 1994

J.N. Kapur; P.K Sahoo; A.K.C Wong. A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram - October 16, 1984

J. Kittler; J. Illingworth; Minimum Error Thresholding – 8 July 1985

<http://computervisionwithvaibhav.blogspot.com.br/2015/10/otsu-thresholding.html>.
Acesso: março/2017

http://imagej.net/Auto_Threshold. Acesso: dezembro/2016