Métodos de Análise de Imagem para Contagem Automática de Células

Duarte Rodrigues [1,2], Emanuel Ricardo Brioso [1,3], Mariana Xavier [1,4]

[1] Faculdade de Engenharia da Universidade do Porto
[2] up201708998@fe.up.pt
[3] up201705429@fe.up.pt
[4] up201705177@fe.up.pt

***Resumo* — Este projeto consiste no processamento de imagem em MATLAB com o objetivo segmentar uma *Region of Interest* (ROI) a partir de imagens microscópicas e de segmentar as células presentes, de forma a proceder à sua contagem. Em hemocitometria, a ROI situa-se na delimitação das linhas triplas, sendo obtida uma imagem binária que definia a branco esta região retangular. No que toca à segmentação celular individua, o aumento do contraste da imagem para a consequente deteção das bordas celulares arredondadas tornou possível a identificação da grande maioria das células.**

Palavras-chave — contagem celular, análise de imagem, região de interesse (ROI), segmentação, MATLAB, Índice Jaccard.

I. Introdução

O número de células numa cultura é uma medida de controlo altamente usada para obter informações acerca da densidade celular nas mesmas, usado como medida de avaliação em casos de proliferação, viabilidade ou toxicidade [1]. Geralmente este processo é feito de forma manual, o que, apesar de simples, implica subjetividade e concentração humana, que se vai refletir na fiabilidade dos resultados, e é um processo que demora bastante tempo [2]. Assim, de forma a melhorar a exatidão e a velocidade do processo, têm vindo a ser desenvolvidos métodos automáticos baseados em análise de imagem e visão computacional.

Este projeto tem como objetivo o desenvolvimento de métodos de análise de imagem que permitam fazer a contagem automática das células cancerígenas HL60, relacionadas com a leucemia, usando imagens obtidas por microscopia e por um hemacitómetro [3] (dispositivo cuja área de contagem é conhecida, delimitada por linhas de um determinado tamanho que ajudam na contagem, permitindo determinar a densidade celular em relação a um volume específico de solução [4]). Para tal, torna-se necessário em primeiro lugar delinear apenas a área correspondente à área de interesse e só posteriormente proceder à contagem celular no seu interior.

II. Métodos

A. Delineamento da Região de Interesse (ROI) – Task 1

Nesta primeira etapa começou-se por abrir o diretório das imagens provenientes da base de dados utilizada. Estas foram, então, lidas e processadas uma a uma. Olhando para as imagens, observa-se que estas contêm uma grelha de linhas individuais, sendo a região de interesse definida pelo quadrado entre quatro conjuntos de três linhas adjacentes. Assim, tentou-se isolar essas linhas e preencher todo o seu interior, obtendo-se a máscara binária desejada.

A extração da região de interesse foi feita através da função *segmentROI*, representando-a numa imagem binária como um retângulo branco em fundo preto. Posto isto, para avaliar a exatidão do processo, recorreu-se à função *evaluateROI*. Esta lê o *ground truth* e procede à comparação, determinando os três valores que vão permitir a sua avaliação: o índice de *Jaccard*, a média e a máxima das quatros distâncias Euclidianas entre os vértices das regiões (em pixéis). Juntamente com isto, as imagens obtidas foram gravadas numa nova pasta, juntamente com um relatório geral em ficheiro texto com o nome de todas as imagens e os respetivos índices obtidos.

1) Segmentação da ROI

Na função *segmentROI*, começou-se por fazer um pré-processamento à imagem. O primeiro passo foi a conversão para uma imagem em escala de cinza, em vez de RGB, de forma a poder efetuar todas as operações seguintes. De seguida, aplicou-se um filtro de mediana, capaz de remover ruído pontual da imagem, tanto claro como escuro. Tendo a imagem suavizada, esta foi binarizada, ou seja, imagem apenas em tons preto e branco, recorrendo ao método de *Otsu*, de forma a se obter apenas as linhas microscópicas e alguns contornos celulares mais evidentes.

Após o pré-processamento, recorreu-se à operação *close* com dois elementos estruturantes diferentes: um em forma de linha horizontal e outro de linha vertical. Esta operação é usada para fundir os conjuntos de três linhas adjacentes, preenchendo os espaços entre elas, obtendo-se uma imagem constituída por uma grelha de linhas individuais, em que quatro delas são bastante mais largas que as outras. Assim, passamos de ter de identificar quatro conjuntos de linhas para identificar apenas 4 linhas.

O objetivo seguinte consistiu na eliminação das linhas mais finas, que não delimitam a ROI. Para tal, recorreu-se a um elemento estruturante, na forma de disco, aplicado na função *open*, que elimina as linhas desejadas, sem erodir as linhas mais grossas.

Depois, recorreu-se à função *imfill* para preencher o espaço delimitado pelas 4 linhas, de forma a se obter a máscara branca. No entanto, agora tornou-se necessário cortar as linhas para lá dos seus cruzamentos, de forma a obter apenas o quadrado central desejado.

Para este corte, recorreu-se à função de MATLAB *bwboundaries* que retorna o conjunto das coordenadas dos pixéis que delimitam um objeto. Com dois simples ciclos conseguiu-se determinar, a partir das coordenadas obtidas, tanto as linhas superior e inferior como as linhas laterais, esquerda e direita, que delimitam a área de interesse, procedendo ao corte de tudo o que estava para lá delas, ou seja, definindo-lhes o valor de intensidade igual a zero.

2) Avaliação da Segmentação Obtida

Esta função abre, em primeiro lugar, o diretório contendo as máscaras segmentadas manualmente, lendo aquela que corresponde ao índice que lhe é passado como parâmetro. Esta funciona como *ground truth*.

Seguidamente é feito o cálculo da similaridade, dada pelo valor de *Jaccard*, aplicando diretamente a função de MATLAB *jaccard* com as duas imagens binárias como parâmetros.

A obtenção das distâncias Euclidianas necessitou de mais processamento. Recorreu-se novamente à função *bwboundaries* para determinar os pontos na fronteira da máscara de forma a achar os quatro cantos que a caracterizam, sendo estes incluídos num vetor de vértices. Isto foi feito tanto para a máscara obtida como para o *ground truth*. Por fim, com um simples ciclo calculou-se a distância Euclidiana para cada um dos quatro vértices, determinando tanto a sua média como o seu valor máximo.

Na figura abaixo encontra-se um resumo do processamento relativo à primeira tarefa.

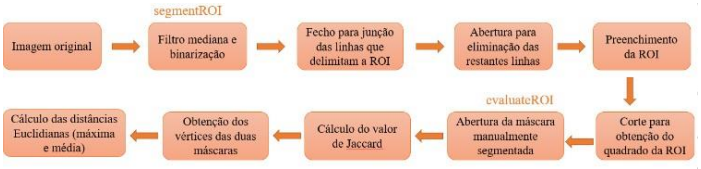

Fig.1. Esquema do processo utilizado para o delineamento da ROI.

B. Segmentação e Contagem Celular – Task 2

Nesta segunda etapa começou-se por abrir de novo o diretório das imagens provenientes da base de dados utilizada. Estas foram corridas e processadas uma a uma de forma a se segmentar as células nelas presentes.

Após a leitura de uma imagem, esta é convertida para a escala de cinza. A primeira função na etapa de processamento é a *getROI* que tem o objetivo de obter apenas a parte da imagem correspondente à ROI. Esta etapa é necessária visto que se pretende proceder à contagem apenas dentro desta. De seguida, recorreu-se à função *segmentCells* que faz o processamento necessário para a identificação das células, sendo retornadas sob a forma de retângulos que envolvem cada uma das células identificadas. Dentro desta função é ainda chamada a *excludeBorders* que permite garantir a condição: células que ultrapassem os limites inferior ou direito da ROI não entram na contagem. De seguida, é necessário obter também os retângulos correspondentes ao *ground truth* para efeitos de comparação. Isto é feito através da função *getGroundTruth*. Posto isto, para avaliar a exatidão do processo, recorreu-se à função *evaluateSegmentation* que permite determinar os seguintes valores: o número de células obtidas na segmentação automática, o número de verdadeiros positivos, falsos positivos e falsos negativos e, com isto, os valores de *recall*, *precision* e *F-measure* com β igual a 1. Juntamente com isto, a informação relativa às posições dos retângulos que envolvem as células é guardada num ficheiro .mat, numa nova pasta que contém os resultados desta etapa, juntamente com um *report* geral em ficheiro texto com o nome de todas as imagens e os respetivos índices obtidos.

1) Obtenção da ROI

A função *getROI* recebe como *input* a imagem a processar e o seu índice no diretório. Com isto, abre o diretório das máscaras correspondentes ao *ground truth*, obtém a máscara correspondente, procedendo à multiplicação entre esta e a imagem em causa.

Desta forma, obtém-se uma moldura preta, com um quadrado correspondente à ROI da imagem de *input* no seu interior, em escala de cinza.

2) Segmentação Celular

A função *segmentCells* é chamada aquando da segmentação celular.

Começa por aplicar um filtro de mediana, de forma a diminuir ruído na imagem, e a fazer uma equalização do seu histograma, que recorre ao algoritmo CLAHE de forma a aumentar o contraste da imagem.

Posto isto, obtém o gradiente da imagem, sendo este binarizado, evidenciando as linhas pertencentes à grelha e os contornos celulares, juntamente com algum ruído no seu interior.

Os contornos celulares são determinados pela função de MATLAB *imfindcircles*, tanto para objetos claros como para objetos escuros, sendo estes concatenados de forma a obter a totalidade das células.

De seguida, chama a função *excludeBorders*, que se baseia em parte do processamento realizado na primeira parte de forma a juntar as linhas adjacentes que formam os limites da ROI e eliminação das restantes linhas da grelha. Posto isto, recorre a *multithresholding* para binarizar apenas as linhas adjacentes, ou seja, que formam a moldura da ROI, deixando tudo o resto a zeros. Seguidamente é calculado o gradiente desta, obtendo as linhas que formam a moldura, tanto exteriores como interiores. Estas linhas são, de seguida, identificadas através da Transformada de Hough. Esta usa ângulos de 0 para identificação de linhas verticais e -90/89 para identificação de linhas horizontais. Correndo a matriz de linhas obtidas, identificou-se, em primeiro lugar, as linhas correspondentes aos limites exteriores direito e inferior e, tendo isto, à identificação dos limites interiores direito e inferior, que são retornados pela função.

Voltando à *segmentCells*, é aplicado o resultado da função anterior, excluindo as células que se encontrem para lá das linhas detetadas. O passo seguinte é fazer a passagem dos centros e raios obtidos até agora para coordenadas de um retângulo, sendo as coordenadas do ponto superior esquerdo e os comprimentos da altura e da largura guardados num *array*, que é retornado por esta função.

O último passo baseia-se no facto de, por vezes, uma célula ser encontrada tanto pelo modo *bright* como pelo modo *dark*. Assim, torna-se necessário eliminar as células que se encontrem repetidas. Células que foram encontradas múltiplas vezes apresentam *bounding boxes* em localizações sobrepostas logo, ao calcular o *overlap* entre elas, recorrendo a função de MATLAB *bboxOverlapRatio*, as que tiveram uma elevada área de sobreposição são testadas, eliminando a *bounding box* mais pequena, resultante da segmentação.

3) Avaliação da Segmentação

Esta recebe como parâmetros a informação que permite recriar os retângulos envolventes das células, tanto da segmentação manual como da automática.

Em primeiro lugar, obtém o número de células em cada uma delas, através do tamanho das matrizes.

Para cada célula são criadas máscaras que contêm um retângulo que indica a localização da célula. A partir dessas máscaras são calculados os índices de Jaccard de todas as células segmentadas manual e automaticamente, pela função de Matlab *bbOverlapRatio*.

As células TP (*True Positive*) correspondem a células cujo índice de Jaccard é igual ou superior a 0,5. Os FP (*False Positive)* correspondem a células segmentadas manualmente que não foram correspondidas a células no *ground truth*, ou seja, índice de Jaccard igual a 0. Neste caso, como havia interseção entre cantos de retângulos correspondentes a duas células diferentes, foi definido um *threshold* máximo de 0,01. Por fim, os casos de FN (*False Negative)* são todas as células que têm um índice de *Jaccard* entre 0 e 0,5 ou as células na segmentação automática que não foram correspondidas a células na segmentação manual, ou seja, índice de *Jaccard* igual a 0. Mais uma vez, o 0 foi substituído por um *threshold* correspondente a 0,01.

Seguidamente calcula os valores de *recall* (R), *precision* (P) e *F-measure* com β igual a 1 (F1) através da aplicação das fórmulas diretas:

$$R = \frac{TP}{TP+FN} \quad (1)$$

$$P = \frac{TP}{TP+FP} \quad (2)$$

$$F_\beta = \frac{(\beta^2+1)PR}{(\beta^2 P)+R} \quad (3)$$

Na figura 2 encontra-se um resumo do processamento relativo à segunda tarefa.

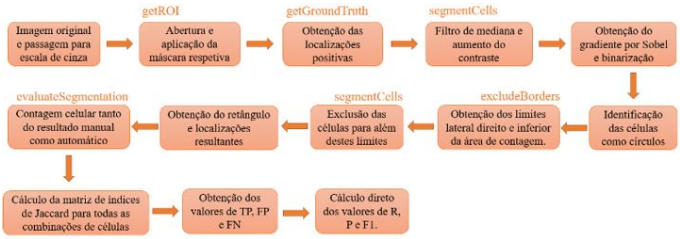

Fig.2. Esquema do processo utilizado para o segmentação e contagem celular.

III. Resultados e Discussão

A. *Delineamento da Região de Interesse (ROI)*

Quanto às 50 imagens de treino, estas mostraram resultados de segmentação como mostra a figura 3.

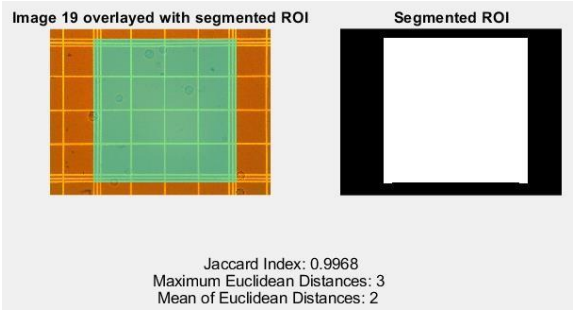

Fig.3. Resultado relativo à imagem 19 da base de dados.

Como se pode observar, a ROI segmentada é bastante semelhante à ROI fornecida. Observando de perto o bordo inferior verifica-se que este não é uma linha perfeitamente reta. Isto é algo que acontece em algumas das imagens devido a uma ligeira erosão dos bordos da ROI e ligeira inclinação de algumas imagens.

Quanto aos valores para avaliação, estes encontram-se resumidos para toda a base de dados de treino na seguinte tabela:

Tabela I. Resultados obtidos da tarefa 1 na fase de treino.

	Média	Desvio Padrão
Jaccard	0,9910	0,0023
Máximo Distâncias	6,2173	2,0312
Média Distâncias	4,5887	1,4158

Quanto à similaridade, os valores de Jaccard são bastante elevados e muito constantes entre si (baixo desvio padrão), o que mostra que as ROI obtidas são muito significativas da ROI verdadeira. Quanto às distâncias, estes valores estão representados em pixéis. Tendo em conta que as imagens originais têm dimensão 1200x1600 e as ROI's cerca de 1000x1000, então os valores obtidos são também bastante bons, estando os vértices em posições muito próximas entre si.

Procedendo da mesma maneira para a fase de teste, as segmentações foram também bastante significativas das ROI's manualmente segmentadas, estando os resultados da avaliação presentes na tabela abaixo.

Tabela II. Resultados obtidos da tarefa 1 na fase de teste.

	Média	Desvio Padrão
Jaccard	0,9909	0,0003
Máximo Distâncias	6,2862	2,2987
Média Distâncias	4,5912	1,4376

Por observação de ambas as tabelas, verifica-se que os valores são semelhantes em ambos os conjuntos de imagem, pelo que a abordagem escolhida teve sucesso também nos novos dados, para os quais não foi adaptada.

O tempo total para leitura, processamento, avaliação e armazenamento dos resultados de aproximadamente 30 segundos em ambos os conjuntos de imagens.

B. *Segmentação e Contagem Celular*

No que toca à segmentação celular, os resultados da nossa implementação mostraram-se satisfatórios conseguindo segmentar, identificar e contar a maioria das células presentes nas imagens de treino e teste. Em termos temporais a segmentação celular, quer na fase de teste e treino, demorou cerca de 8 minutos.

Contudo, existem alguns contratempos na nossa implementação no que toca à identificação da totalidade das células indicadas pelas posições *ground truth.*

O primeiro aspeto em que a nossa abordagem atravessa algumas dificuldades é na segmentação de células quando estas estão sob linhas. Isto verifica-se quer na região central quer nas bordas, em que certas vezes as células são corretamente encontradas e outros casos em que não.

Outro aspeto importante mencionar é a diferença de tamanho entre o *ground truth* e a segmentação automática. Muitas vezes, ao longo das imagens, quer em treino quer em teste, o tamanho da segmentação manual é muito maior do que o da célula em questão. Isto revelou-se um problema visto que a segmentação alcançada pelo nosso algoritmo cria *bounding boxe*s justapostas à célula.

O último problema a salientar reside no facto de nem todas as células serem perfeitamente redondas. Desta forma, como o algoritmo está programado para identificação de círculos, algumas mais deformadas acabam por não ser detetadas. A figura 4 é uma figura representativa dos nossos resultados, visto que apresenta alguns falsos negativos derivados de todos estes problemas.

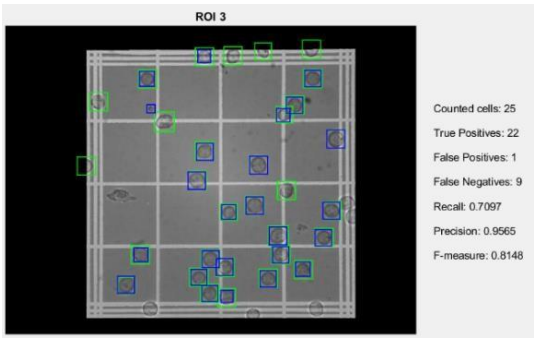

Fig.4. Resultado relativo à imagem 3 da base de dados.

Os resultados relativos às fases de treino e teste encontram-se nas tabelas III e IV. Em termos de comparação, os resultados da avaliação geral diminuíram ligeiramente, como seria de esperar na fase de teste, visto que os *thresholds* definidos ao longo do código não estavam adaptados a este novo *set* de resultados.

Tabela III. Resultados obtidos da tarefa 2 na fase de treino.

	Média	Desvio Padrão
Recall	0,7612	0,1152
Precision	0,8783	0,1449
F-measure	0,8058	0,0963

Tabela IV. Resultados obtidos da tarefa 2 na fase de teste.

	Média	Desvio Padrão
Recall	0,7497	0,1670
Precision	0,9151	0,1198
F-measure	0,8148	0,1340

Quanto aos significados dos valores propriamente ditos, os valores inferiores de *recall* devem-se à quantidade de falsos negativos encontrados, que podem ser justificados pelas razões previamente mencionadas. Nos valores de *precision* verifica-se que estes são bastante mais elevados, o que mostra que o nosso algoritmo não identifica células onde elas não existem, gerando assim poucos falsos positivos. Combinando estes dois valores na *F-measure*, sendo a ambos atribuída a mesma importância, obtêm-se valores finais bastante satisfatórios.

IV. Conclusão

Concluímos assim que o algoritmo implementado para a obtenção da região de interesse é bastante robusto, conseguindo resultados bastante precisos, alcançando uma máscara bem definida. Quanto à contagem celular, os nossos resultados foram satisfatórios, havendo certas células em localizações específicas e mais desafiantes que se tornaram mais difíceis de contabilizar. Apesar disso, as avaliações finais, através dos valores evidenciados nas tabelas, revelam que esta é uma abordagem assertiva.

V. Trabalhos Futuros

A segmentação das células nos limites da ROI é mais desafiante que a segmentação das restantes células, visto que estas são atravessadas por linhas e por vezes encontram-se parcialmente fora da ROI, sendo difícil identificar usando a abordagem adotada neste projeto. Portanto, com objetivo de conseguir segmentar corretamente mais células, o passo seguinte seria melhorar a segmentação nas delimitações, recorrendo a de medidas mais complexas como modelos de *machine learning*, que poderiam melhorar a precisão.

References

[1] "Cell Counting". BioTek.

[2] "Manual and Automated Cell Counting". Isogen.

[3] Akin Ozkan. "Computer Vision based automated cell counting pipeline: a case study for HL60 cancer cell on hematocytometer". Biomedical Research. 2018.

[4] Yevgeniy Grigoryev. "Cell Counting with a Hematocytometer: Easy as 1, 2, 3". BiteSize Bio. Dezembro 2014.