

Nanodegree Engenheiro de Machine Learning

Proposta de projeto final

Vinicius dos Santos Cruz

17/12/2018

Proposta

O teste *Live/Dead* de viabilidade celular é baseado no princípio de permeabilidade a membrana. É muito usado para identificar a potencialidade de dano celular frente a diferentes formulações. Para quantificar as células vivas e mortas o devemos separar as células de outros “ruídos” na imagem. No caso em questão pode ser considerado ruído: sujeira do ambiente, material usado para armazenamento da célula entre outros. Depois é quantificar as células totais, depois as mortas, marcadas com um reagente para mudar sua cor na imagem, assim temos a quantidade de células vivas e mortas. O cálculo de viabilidade celular se dá, portanto, através do quociente de células vivas pelas células totais multiplicado por 100.

Existem softwares que fazem o trabalho de quantificar as células em amostras porém são inviáveis para o nosso caso, pois o trabalho para contagem de células é muito grande para uma única imagem e como a pesquisa científica é no sentido de desenvolver uma plataforma para que de uma única vez seja possível testar varias culturas de células, podemos ter dezenas e até centenas de culturas de células em um único experimento. O que torna o trabalho manual cansativo, estressante e suscetível a erros, já que é necessário fazer o *upload* de cada imagem de forma individualizada, identificar as áreas da imagem que não são de interesse, como impurezas, selecionar as áreas a serem analisadas e então realizar o comando de contagem.

Diante do exposto, Fui motivado a desenvolver esse projeto pois minha noiva é Doutoranda em Ciências da Saúde pela UFS/OSHU e no seu trabalho de pesquisa teve uma fase em que teve que justamente fazer a contagem das células, diversas vezes, para experimentos diferentes e repetições dos experimentos. Após ouvir suas queixas, conhecer como o software funcionava e descobrir o quão trabalhoso é fazer essa análise, resolvi desenvolver algo que pudesse ajuda-la e quem sabe futuramente virar um produto.

O problema é a contagem de células mortas e células vivas dentro do microambiente impresso numa impressora 3d. Uma foto do microscópio da amostra inteira tem vários microambientes e o que nos interessa é somente a contagem dentro de cada microambiente. A imagem [1] mostra uma imagem contendo todos os microambientes de uma amostra, cada quadrado azul é um microambiente onde deve ser cultivada as células.



1 – Foto Inteira tirada pelo microscópio.

Nosso conjunto de dados consiste em 189 imagens que são recortes, de alta definição, da imagem [1], tiradas pelo microscópio e serão usadas para o treinamento do nosso algoritmo. Temos 3 grupos de imagens para cada quadrante. Tomando como exemplo o quadrante Area_1_9 temos uma imagem Area_1_9.tiff que é a foto das células vivas e mortas com destaque do reagente que marca as células mortas, Area1_9_DAPI.tiff que é a foto das células vivas e mortas sem o destaque do reagente e a imagem Area1_9_GFP.tiff que é a foto somente das células que foram marcadas pelo reagente que identifica as células mortas.

O objetivo do projeto é usar visão computacional e algoritmos de classificação para delimitar os microambientes, classificar células e ruídos para gerar novas imagens que possam ser facilmente classificadas por um algoritmo que diferencie células mortas e vivas. A ideia é que isso possa ser feito, não somente para esse grupo de imagens mas para qualquer outro, desde que tenha o mesmo padrão, não importando as cores dos reagentes.

O modelo de referência a ser usado foi encontrado no artigo *Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm*

(https://www.researchgate.net/publication/283185016_Image_Segmentation_Using_K-means_Clustering_Algorithm_and_Subtractive_Clustering_Algorithm)

Ele propõe um algoritmo que consiste em fazer alguns tratamentos na imagem antes de aplicar o K-Means e depois compara o resultado do algoritmo com o resultado do K-Means. Usando como métricas equações de erro RMSE e PSNR.

RMSE (Root Mean Square Error) diz respeito a performance da imagem de saída, segmentada. Mede o quanto que a imagem de saída deriva da imagem de entrada.

$$RMSE = \sqrt{\frac{1}{n_x n_y} \frac{\sum_0^{n_x-1} \sum_0^{n_y-1} [r(x, y)]^2}{\sum_0^{n_x-1} \sum_0^{n_y-1} [r(x, y) - t(x, y)]^2}}$$

PSNR (Peak to Signal Noise Ratio) mede a qualidade da imagem de saída.

$$PSNR = 10 \cdot \log_{10} \left[\frac{\max(r(x, y))^2}{\sqrt{\frac{1}{n_x n_y} \frac{\sum_0^{n_x-1} \sum_0^{n_y-1} [r(x, y)]^2}{\sum_0^{n_x-1} \sum_0^{n_y-1} [r(x, y) - t(x, y)]^2}}} \right]$$

Valores pequenos para RMSE significa que a imagem tem boa qualidade e valores pequenos de PSNR significa que a imagem tem uma qualidade ruim.

Usarei como ponto de partida os passos demonstrados no artigo citado, com o complemento de mais algumas fontes sobre visão computacional e segmentação de imagens, como por exemplo esse artigo sobre remoção de ruído em imagens

(<https://ensinandomaquinasblog.wordpress.com/2016/01/13/remocao-de-ruído-de-imagens-utilizando-aprendizado-de-maquina/>) e esse trabalho sobre visão computacional para contar itens em uma imagem (<https://editora.unoesc.edu.br/index.php/acet/article/viewFile/6586/5077>)

O algoritmo proposto no artigo consiste em fazer um alongamento de contraste, tendo em vista que a maioria das imagens médicas tem um baixo contraste. Depois disso o é usado um algoritmo de “subtractive clustering” para encontrar os centros, baseado no valor potencial da imagem. O número de centros gerados dependem do número de cluster k. Esses centros são usados para inicializar os centros para o algoritmo K-means. Após a segmentação da imagem, a imagem ainda deverá conter ruídos ou regiões indesejadas. Para remover esses problemas usaremos um filtro médio, descrito no artigo.