



Classificador Haar cascade para detecção de leucócitos em imagens de exames de sangue: um estudo inicial

Haar cascade classifier for leukocyte detection in blood test images: An initial study.

José Otávio Botelho de Oliveira*, Vander Teixeira Prado[†].

RESUMO

A visão é uma das funções mais importantes do ser humano e é devido a esta que ele tem a capacidade de enxergar tudo a sua volta. O processo de passar essa função para as máquinas a partir da captação e processamento de imagens é denominado visão computacional. O foco deste projeto são os estudos de processamento de imagens, medicina e biomedicina, ao tentar coletar informações do sangue humano através de um esfregaço e analisa-los através de um classificador Haar cascade. Utilizou-se o software Atom para desenvolver, a partir de comando já existentes da linguagem python, um código capaz de contar e identificar essas células sanguíneas retiradas de hemogramas. Entretanto, ao aplicar o classificador nas amostras do banco de dados com imagens de esfregaços sanguíneos, nem todas as células foram detectadas, ocorrendo muitos falsos positivos e negativos, conduzindo à necessidade de novos testes e combinações de parâmetros de treinamento.

Palavras-chave: Visão computacional, processamento de imagem, medicina.

ABSTRACT

Vision is one of the most important functions of the human being and it is because of this that he has the ability to see everything around him. The process of passing this function on to machines from image capture and processing is called computer vision. The focus of this project is imaging, medicine and biomedical studies, by trying to collect information from human blood through a smear and analyze it through a Haar cascade classifier. The Atom software was used to develop, based on existing commands in the python language, a code capable of counting and identifying these blood cells taken from hemograms. However, when applying the classifier to the database samples with blood smear images, not all cells were detected, with many false positives and negatives occurring, leading to the need for new tests and combinations of training parameters.

Keywords: Computer vision, image processing, medicine.

1 INTRODUÇÃO

O processo de tentar repassar para máquinas a capacidade de visão é denominado como visão computacional. Esta área de estudo busca captar imagens, melhorá-las, separá-las em regiões e extrair informações. Segundo Backes (2016), assim como a visão humana, a visão computacional também pode ser dividida em algumas etapas: aquisição, processamento, segmentação, análise de imagens e reconhecimento de padrões. Os benefícios da visão computacional na medicina são diversos, como por exemplo redução de tempo e aumento na eficácia de exames e diagnósticos devido a falhas humanas, principalmente relacionadas à repetição de ações, como exames laboratoriais. De acordo com Ushizima (2004), usando um banco de dados com imagens de exames de sangue, por exemplo, pode ocorrer em alguns utilizar métodos da visão

* Engenharia eletrônica, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil; joseo@alunos.utfpr.edu.br

[†] Universidade Tecnológica Federal do Paraná, Campus Cornélio Procópio; vanderprado@utfpr.edu.br



computacional para extração de características nas células da amostra, e com isso realizar a contagem de leucócitos, identificação de leucemia linfóide, etc.

Segundo Bessis (1997) o sangue é constituído por três grupos de celulares: glóbulos vermelhos, eritrócitos ou hemácias; glóbulos brancos ou leucócitos; plaquetas ou trombócitos. Essas células podem ser identificadas pelo seu conteúdo interno: as hemácias não são células completas, pois não possuem núcleo definido e tem a função de transportar oxigênio pelo corpo. As plaquetas, assim como as hemácias, não são células completas, porém são utilizadas em processos bioquímicos que ocorrem no corpo; como a coagulação. O último, mas não menos importante, são os glóbulos brancos, que possuem núcleo bem definido e atuam no sistema de defesa do organismo. Os leucócitos têm núcleos diferentes, cada um deles possui uma propriedade e sua identificação é feita através deles.

Nos hemogramas, a contagem dessas células é feita automaticamente por máquinas específicas, como a Sismex CS 1000i, que já disponibiliza a quantidade de células da amostra. Porém existem alguns problemas como o alto do equipamento e manutenção, o que abre margem para pesquisas e desenvolvimento de equipamentos de menor custo para os mesmos fins objeto de estudo deste projeto.

A fim de identificar as células sanguíneas de uma amostra, embasou-se nas técnicas de detecções de objetos através da teoria sobre haar cascade.

2 MÉTODO

2.1 Software

O software utilizado para a realização foi o Atom, um editor de texto gratuito que suporta a linguagem de programação Python. Outros recursos como o prompt de comando foram necessários para o desenvolvimento do classificador.

2.2 Classificador

De acordo com Macedo (2020), o classificador haar cascade se trata de uma técnica para identificação. O funcionamento do classificador gerado por este método depende de três componentes, os quais são: características de haar, AdaBoost e classificador em cascata.

O primeiro componente é uma combinação de características haar: dois ou mais retângulos juntos de cores branca e preta, utilizadas como entrada para o classificador. Através de uma soma de pixels em áreas retangulares definidas obtêm-se características de linha, de borda ou de objetos centralizados. O AdaBoost é um algoritmo de aprendizagem supervisionado que permite além de treinar o classificador, selecionar as características que melhor separam os conjuntos de imagens positivas e negativas, onde, as imagens positivas são as que contêm o objeto de estudo e as imagens negativas não o possuem. No final do algoritmo é realizada a combinação de classificadores em cascata. De acordo com Macedo (2020), o procedimento em cascata permite obter altas taxas de detecção de acordo com o quanto o classificador foi treinado, onde o objeto somente será detectado com êxito, se a sub janela a qual pertence for aprovada pelos classificadores da cascata.

2.3 Parâmetros do classificador utilizando a biblioteca openCv

De acordo com Granatyr (2021), existem 4 parâmetros que determinam a eficácia do classificador e o poder de processamento necessário para que ocorra a detecção, os quais são: *Scale Factor*, *minNeighbors*, *minSize* e *maxSize*.



O *Scale Factor* (fator de escala) especifica a proporção em que se quer redimensionar o objeto, levando em consideração que objetos mais próximos a uma câmera quando uma foto é tirada, são maiores. Quanto menor o valor escolhido para o parâmetro, maior o valor de processamento computacional requerido.

O segundo parâmetro é o *minNeighbors*, o qual regula quantos vizinhos cada retângulo candidato deve ter, pois quando muitos objetos estão próximos é difícil reconhecer todos de forma eficaz. Quando definido com valores altos apresenta detecções mais exatas, porém ocorrem menos detecções.

Os outros dois parâmetros, *minSize* e *maxSize*, especificam o tamanho mínimo e máximo de um objeto a ser encontrado, respectivamente.

2.4 Etapas de criação do haar cascade

Como o objetivo da tarefa é a criação de um classificador forte, que possua uma grande quantidade de acertos nas detecções do objeto de estudo, o primeiro passo é selecionar um objeto que não possua grandes quantidades de variação de cor, formato e outros aspectos, portanto, foi selecionado como objeto de estudo a célula de um leucócito já que são parecidas no formato mesmo com conteúdo nuclear diferente.

A segunda etapa é selecionar imagens negativas, estas imagens são fotos de prováveis fundos do objeto a ser detectado e de imagens aleatórias as quais não possuem o leucócito. O terceiro passo é selecionar imagens positivas, isto é, imagens que possuem os objetos de estudo.

Selecionadas todas as imagens, gera-se então um vetor de estudo para se treinar o cascade e, portanto, criar o haar cascade que identifica a célula.

2.5 Criação do haar cascade especializado

Primeiramente é necessário instalar alguns recursos da biblioteca openCv, os quais são: *cratesamples* e *traincascade*, cuja função é criar imagens a partir de um único objeto e treinar o classificador, respectivamente. Em seguida, é preciso obter um grande número de imagens negativas (por volta de 300 vezes o número de imagens positivas usadas) de mesmo tamanho e em tons de cinza. O banco de imagens negativas utilizado foi obtido a partir do curso “Detecção de objetos com python e opencv” da plataforma Udemy.

Uma vez que o banco de imagens disponível contém poucas imagens para um bom treinamento, utilizando o comando da Figura 1, a partir de uma imagem negativa e uma imagem com o objeto de estudo, criou-se uma pasta de imagens positivas, variando o objeto em torno de seus eixos, resultando em 300 novas imagens. O comando da Figura 1 ainda limita o tamanho do objeto e torna todo o fundo branco em transparente. Este recurso foi utilizado 10 vezes para aumentar o número de imagens positivas e melhorar o classificador, sendo que as 10 imagens escolhidas têm formatos de células parecidas, porém não são idênticas, pois diferenciam-se entre si a partir do núcleo celular.

Figura 1 – Utilização do comando *createsamples* da biblioteca *opencv* para geração de novas imagens positivas.

```
C:\Users\José Otávio Botelho>opencv_createsamples -img celula.jpg -bg negativas/bg.txt -info positivas/positivas.lst -maxxangle 0.5 -maxyangle 0.5 -maxzangle 0.5 -w 48 -h 48 -num 300 -bgcolor 255 -bgthresh 8
```

Fonte: Autoria própria (2021).

Posteriormente, gerou-se um vetor de imagens positivas, o vetor possui em cada linha o caminho de cada imagem e vai ser utilizado no treinamento do próximo passo. O comando para gerar o arquivo está representado na Figura 2.



Figura 2 – Criação do vetor de imagem.

```
C:\Users\José Otávio Botelho>opencv_createsamples -info positivas/positivas.lst -num 300 -w 18 -h 18 -vec positivas.vec_
```

Fonte: Autoria própria (2021).

Este comando, assim como o anterior, gera um arquivo que mostra as posições exatas onde a célula está em cada imagem. Para treinar o classificador, criou-se uma nova pasta dentro da pasta de imagens negativas, a qual precisa conter os arquivos instalados do site da biblioteca openCv. O treinamento do classificador se dá a partir do comando da Figura 3;

Figura 3 – Treinamento do classificador.

```
C:\Users\José Otávio Botelho>opencv_traincascade -data classificador -vec positivas.vec -bg bg.txt -numPos 1600 -numNeg 800 -numStages 10 -w 18 -h 18 -precalcBufSize 1024 -precalcIdxBufSize 1024_
```

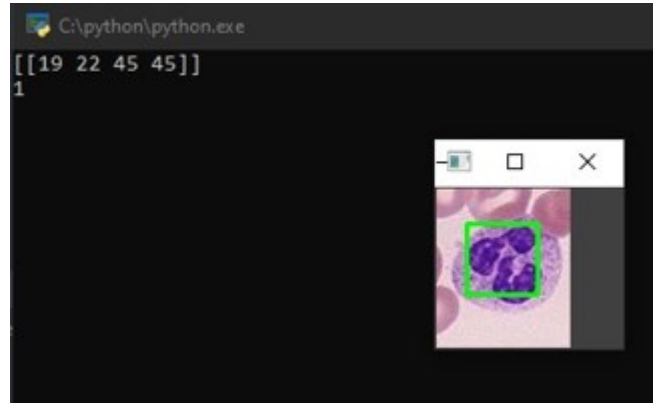
Fonte: Autoria própria (2021).

Fez-se então um treinamento de 10 estágios, como mostrado na figura 3. Dois parâmetros são gerados neste processo, estes são o *Hit Rate* e o *False Alarm*, os quais mostram a taxa de acerto da classificação de imagens positivas e a classificação de imagens negativas dadas como positivas, respectivamente. Assim, fica possível analisar a qualidade do classificador sem aplicá-lo em imagens.

3 RESULTADOS

Para realização do teste do classificador, criou-se um código no software Atom que: i) importa a biblioteca openCv; ii) abre a imagem a qual quer se realizar a detecção; iii) carrega o classificador a partir do arquivo xml gerado na etapa de aprendizagem; iv) converte a imagem para escala de cinza; e v) gera retângulo em volta do objeto detectado. Como saída, tem-se a quantidade de objetos detectados, a posição e o tamanho de cada objeto. Para um teste controlado, utilizou-se uma das imagens que foram utilizadas como positiva, o resultado é mostrado na Figura 4. O retângulo gerado em torno do objeto detectado se inicia no 19 ° pixel na horizontal e 22 ° pixel na vertical. Os lados do retângulo possuem 45 pixels na vertical e 45 na horizontal.

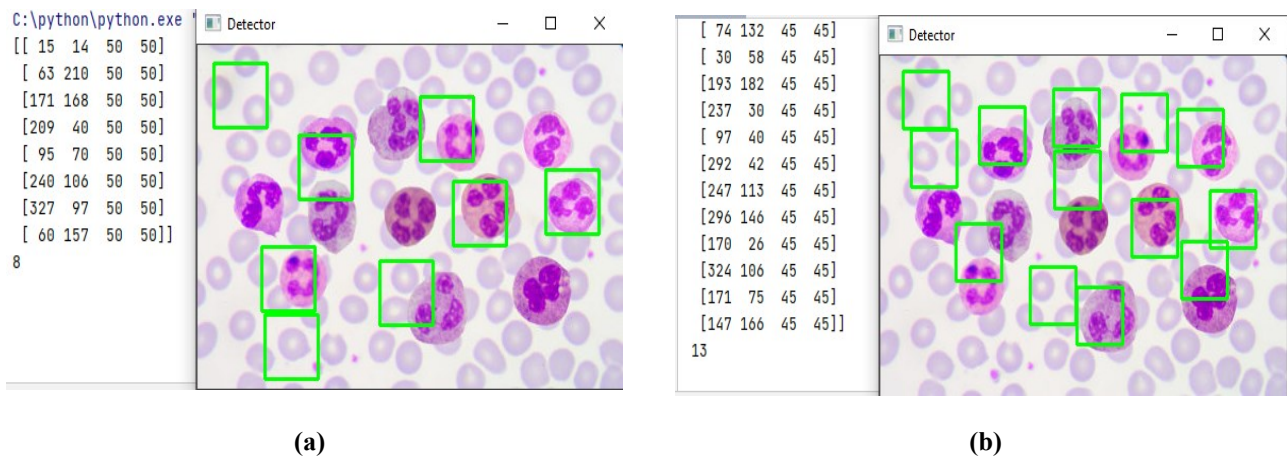
Figura 4 – Imagem utilizada para o teste controlado do classificador.



Fonte: Autoria própria (2021).

Como esperado, o classificador obteve êxito ao encontrar o objeto de estudo. Para os testes posteriores, utilizaram-se algumas imagens do banco de dados e imagens gratuitas retiradas da internet e o resultado de um caso está na figura 5(a). Dos 8 objetos identificados (quadrados verdes), 3 estão bem posicionados sobre os leucócitos, 3 estão entre o objeto a ser detectado e o fundo, dando margem a erros de interpretação e 2 são falsos-positivos, detectando leucócito onde não tem.

Figura 5 – Teste com imagens não utilizadas na fase de treinamento



Fonte: Autoria própria (2021).

Mudaram-se alguns parâmetros do classificador para execução do teste, como fatores de escala e outros. Um dos resultados está ilustrado na Figura 5(b). Neste caso, embora mais quadrados verdes estejam sobre os leucócitos, ainda se posicionam entre o objeto a ser identificado e o fundo, não centralizando e gerando margem de erros de interpretação, principalmente no caso de leucócitos próximos entre si. Apresentou-se novamente pelo menos 3 falsos-positivos.



4 CONSIDERAÇÕES

Observa-se que para as imagens utilizadas no treinamento, como esperado, o resultado adequado, tendo em vista que foi encontrado o objeto de estudo. Considerando-se outras imagens há casos com resultados coerentes, porém há também imagens com muitos falsos positivos e negativos, mostrando que há muito a ser melhorado no processo de reconhecimento de leucócitos em imagens de exame de sangue.

Por se tratar de um estudo inicial, não se testou um conjunto relativamente grande de imagens para se ter um percentual de acertos ou mesmo utilização de índices de acurácia e performance do algoritmo.

Há vários fatores a serem considerados e analisados na continuação do trabalho, desde a quantidade de imagens do banco, sem criação de imagens a partir das mesmas imagens iniciais, a diferentes e exaustivos testes com mais variações dos parâmetros do classificador e qualidade das imagens.

REFERÊNCIAS

- BACKES, A; JUNIOR, J. **Introdução a visão computacional usando MATLAB®**. Rio de Janeiro: ALTA BOOKS, 2016.
- BESSIS, M; BRECHER, G. **A second look at stress erythropoiesis**. Unanswered questions. Blood Cells. 1975; 1:409.
- BESSIS, M. **Blood smears reinterpreted**. Tradução G. Brecher. Ed. Springer-Verlag, Berlin, 1997, 1ª edição. 270.
- GRANATYR, J. **Deteção de objetos com python e openCv**. Disponível em: <<https://www.udemy.com/course/deteccao-de-objetos-com-python-e-opencv/>>. Acesso em: 17 de fev. 2021.
- MACEDO, A; GOMES, I. **Algoritmo Haar Cascade Aplicado na Deteção das Placas de Para Obrigatória e de Velocidade Máxima Permitida**. Universidade Federal do Pará, 2020.
- RIBEIRO, D; ÁVILA, T. **Desenvolvimento de um software para contagem específica de leucócitos de sangue periférico**. Alfenas: Curso de Computação/UNIFEMAS, 2012.
- SANTOS, Vanessa Sardinha dos. **"O que é leucócito?"**; *Brasil Escola*. Disponível em: <https://brasilecola.uol.com.br/o-que-e/biologia/o-que-e-leucocito.htm>. Acesso em: 03 de setembro de 2021.
- USHIZIMA, D. **Diagnóstico de leucemia linfóide auxiliado por computador**. Tese (Doutorado) Instituto de Física de São Carlos, 2004.