

Capítulo

4

Processamento Digital de Imagens Médicas com Python e OpenCV

Marcos Vinícius dos Santos Ferreira, Edson Damasceno Carvalho, Antonio Oseas de Carvalho Filho e Alcilene Dalília de Sousa

Abstract

In the medical field, many anomalies are diagnosed from digital images. Early detection of these abnormalities is fundamental for a more efficient treatment and with fewer risks to the patient, besides increasing the chances of a more favorable prognosis. To improve medical diagnosis through imaging, digital image processing studies requirements to extract technical information to enhance the parameters of a more accurate diagnosis. Decision support systems, called Computer Aided Detection / Diagnosis - CAD / CADx, are computational tools to aid medical diagnosis through imaging, giving the expert a second opinion for a more accurate diagnosis. Thus, it is proposed to present an introduction to Digital Image Processing applied to Medical Imaging with the OpenCV library using the Python language. It is intended to allow an introductory view of the PDI through the content provided, providing a greater understanding of the learning with practical applications, arousing greater interest and new ideas for applications in the most varied areas of knowledge.

Resumo

Na área médica, muitas anomalias são diagnosticados a partir de imagens digitais. A detecção precoce dessas anomalias é fundamental para um tratamento mais eficiente e com menos riscos ao paciente, além de aumentar as chances de prognóstico mais favorável. Com o objetivo de melhorar o diagnóstico médico através de imagens, o processamento digital de imagens estuda requisitos para extrair informações técnicas a fim de melhorar os parâmetros para um diagnóstico mais preciso. Sistemas de apoio a decisão, chamados de Computer Aided Detection/Diagnosis - CAD/CADx, são ferramentas computacionais para auxiliar o diagnóstico médico através de imagens, proporcionando ao especialista uma segunda opinião para um diagnóstico mais preciso. Assim, tem-se como proposta

apresentar uma introdução ao Processamento Digital de Imagens aplicado a Imagens médicas com a biblioteca OpenCV usando a linguagem Python. Deseja-se permitir, através do conteúdo ministrada, uma visão introdutória ao PDI, provendo um maior entendimento no aprendizado com aplicações práticas, despertando um maior interesse e novas idéias para aplicações nas mais variadas áreas do conhecimento.

4.1. Introdução

Com a popularização dos dispositivos eletrônicos para captura de imagens e vídeos, a análise automática desses dados através de programas computacionais tornou-se essencial nas mais variadas áreas de domínio da ciência. Para isso, algoritmos e representações são desenvolvidos para permitir que uma máquina reconheça objetos, pessoas, cenas e atividades.

O Processamento Digital de Imagens (PDI) estuda os aspectos visual de certas feições estruturais para o analista humano e fornecer outros subsídios para a sua interpretação, inclusive gerando produtos que possam ser posteriormente submetidos a outros processamentos [Câmara et al. 1996]. A principal função do processamento digital de imagens é fornecer ferramentas para facilitar a identificação e a extração de informações contidas nas imagens [Silva 2001].

O PDI não é uma tarefa simples, na realidade envolve um conjunto de tarefas interconectadas. Algumas etapas do processamento digital de imagens são:

1. **Aquisição:** É nessa etapa que as imagens são capturadas e representadas de forma computacional para serem interpretadas na etapa seguinte.
2. **Pré-Processamento:** Essa etapa prepara a imagem para as fases seguintes que visam à identificação de estruturas de interesse e o relacionamento da estrutura detectada com uma base de conhecimento.
3. **Segmentação:** Essa etapa consiste em técnicas que de alguma maneira consigam formar padrões de agrupamento, gerando sub-regiões que possuem entre si alguma similaridade.
4. **Extração de Características:** Esta etapa tem como objetivo representar, através de valores, uma imagem ou partes dela. Estes valores são medidas fundamentais que representam propriedades contidas nas imagens.
5. **Reconhecimento e Interpretação:** Nessa etapa, os valores obtidos na etapa de extração de características são os insumos para que uma técnica de aprendizado de máquina possa, então, discernir entre possíveis padrões contidos em um grupo de imagens.

Aplicações com finalidades diversas têm sido desenvolvidas por vários grupos de pesquisas, visando a auxiliar na composição de diagnósticos como uma forma de contribuir para a detecção precoce de doenças.

O objetivo deste capítulo é apresentar técnicas de processamento de imagens, estabelecendo como foco principal as aplicações sobre imagens médicas, destinadas a constituírem fontes de dados para sistemas de auxílio ao especialista na tomada de decisão. São apresentados conceitos sobre a aquisição de imagens médicas, técnicas de processamento de imagens e seus efeitos sobre imagens médicas, além de exemplos de aplicações reais de técnicas com o objetivo de auxiliar na detecção precoce de doenças.

As técnicas aqui apresentadas podem ser aplicadas a imagens provenientes de qualquer modalidade. Por questões de delimitação de objeto de estudo, maior ênfase será dada às imagens provenientes de Radiografia. Os exemplos apresentados considerarão apenas imagens em tons de cinza, visto que grande parte das modalidades médicas considera escalas de cinza para representar as diferentes estruturas. No entanto, os conceitos apresentados podem ser aplicados a imagens coloridas com algumas adaptações.

4.2. Informática Médica

A partir da problemática gerada por diversas doenças em nível mundial, uma gama de técnicas que auxiliam o trabalho dos profissionais de saúde tem sido estudadas e implementadas nas mais diversas áreas do conhecimento. Na computação, podemos pontuar os sistemas de auxílio ao especialista, que fornecem informações complementares de exames médicos e são capazes de contribuir significativamente às interpretações e tomadas de decisão de alto risco. Para construção destes sistemas são associadas técnicas relacionadas a Processamento digital de imagens, Inteligência artificial, Aprendizado de máquina, dentre outras áreas de pesquisa, apresentando respostas significativas às demandas apresentadas.

A Informática médica compreende um conjunto de soluções computacionais capazes de auxiliar no trabalho dos especialistas de saúde e na relação dos pacientes com diversas doenças. Esta área de conhecimento perpassa desde os sistemas de auxílio ao diagnóstico a aplicativos que realizam a prevenção ou acompanhamento de diversas doenças. Dessa maneira, tendo em vista a relevância da saúde na vida das pessoas e sociedade, todos os esforços neste sentido são significativos e satisfatórios.

Aplicações com finalidades diversas têm sido desenvolvidas por vários grupos de pesquisas, visando a auxiliar na composição de diagnósticos como uma forma de contribuir para a detecção precoce de doenças. O diagnóstico auxiliado por computador é aquele no qual o radiologista usa os resultados de uma análise computadorizada de imagens médicas como uma segunda opinião na detecção de lesões e na elaboração do diagnóstico.

Sistemas de detecção/diagnóstico auxiliado por computador, são sistemas, muitas vezes acoplados a equipamentos médicos, com a finalidade de auxiliar na tomada de decisão a respeito de um diagnóstico. A importância desses esquemas é enfatizada por [Chan et al. 1990], [Doi et al. 1991], [Ellis et al. 1993], [Giger e MacMahon 1996] e [Petrick et al. 1996], que apresentam taxas de diagnósticos errados em programas de rastreamento e mostram que o uso de sistemas CAD/CADx pode melhorar o desempenho de radiologistas no diagnóstico médico.

É extensa a lista de aplicações possíveis dos algoritmos de visão computacional, que são particularmente relevantes na realização de tarefas insalubres e repetitivas que

de outra forma seriam delegadas a seres humanos. Por ter base conceitual genérica e ampla, a ideia do CAD/CADx pode ser aplicada a todas as modalidades de obtenção de imagem, incluindo radiografia convencional, tomografia computadorizada, ressonância magnética, ultra-sonografia e medicina nuclear. Pode-se, também, desenvolver esquemas CAD/CADx para todos os tipos de exame de todas as partes do corpo, como crânio, tórax, abdômen, osso e sistema vascular, entre outros. Porém, os principais objetos de pesquisa para o desenvolvimento de sistemas CAD/CADx têm sido as áreas de mamografia, para a detecção precoce do câncer de mama, tórax, para a detecção de nódulos pulmonares e *Optical Coherence Tomography* (OCT) e *Heidelberg Retinal Tomography* (HRT), para diagnóstico precoce do glaucoma [de Azevedo-Marques 2001].

Através destes sistemas podemos observar uma contribuição significativa ao trabalho do especialista e, conseqüentemente, nos resultados para os pacientes. Um exemplo é a metodologia proposta por [Carvalho Filho 2016], para a classificação de nódulos pulmonares pertencentes à base de imagem LIDC-IDRI. Foram utilizados os índices de diversidade taxonômica e a distinção taxonômica da ecologia para descrever a textura de nódulos e não-nódulos e para classificação foi utilizado a Máquina de Vetor de Suporte - MVS. A metodologia foi empregada em 833 exames e obteve uma precisão média de 98,11% em sua classificação.

[Carvalho et al. 2017] propôs uma metodologia para o diagnóstico automático de tecidos da mama em maligno e benigno. A metodologia utilizada no estudo foi: aquisição de imagem oriundas da base DDSM, extração de características de textura baseado nos índices de diversidade filogenética, classificação utilizando múltiplos classificadores, e por fim, a validação da metodologia, utilizando de métricas estatísticas. A metodologia foi empregada em 1155 exames e obteve uma acurácia de 94,8% em sua classificação.

[Claro 2015] desenvolveu um método para detecção automática de Glaucoma, que é a segunda principal causa de cegueira no mundo e não possui cura. A metodologia utilizada no estudo foi: aquisição de imagem, pré-processamento nas imagens da retina, extração de características de cor e entropia na área alvo e logo após a seleção de atributos. Os melhores resultados produziram uma sensibilidade de 93,7%, especificidade de 93,6% e uma acurácia de 93,67%.

Os exemplos de aplicações da informática médica apresentadas utilizaram as técnicas de PDI para a manipulação e análise das imagens. A seguir são apresentados os passos desenvolvidos em um sistema completo de auxílio ao especialista, fornecendo informações para compreender as técnicas específicas de manipulação de imagens, ou construir novas abordagens e metodologias que contribuam para o contexto da informática médica.

4.3. Processamento Digital de Imagens

Uma imagem pode ser definida como uma função bidimensional, onde x e y são coordenadas espaciais e a amplitude de f no par de coordenadas $f(x, y)$ é denominado intensidade ou nível de cinza. Quando x e y e a intensidade dos valores de f são finitos e discretos, tem-se então uma imagem digital. Esses elementos possuem uma localização e um valor particular, chamado de pixel [Sousa et al. 2011].

Para aplicações práticas, a imagem é uma função contínua, representada por medidas obtidas em intervalos regularmente espaçados. Os valores assumidos em cada ponto medido são quantificados em um número pertencente a uma escala de diferentes cores. Em imagens médicas, geralmente essas cores são relacionadas a níveis de cinza, sendo atribuído o valor zero à cor mais escura (preto) e o valor máximo M à cor mais clara da escala (branco). Dessa forma, pode-se representar uma imagem como uma matriz onde cada ponto é um valor discreto, conforme mostra a Equação 1, onde n e m correspondem à quantidade de colunas e linhas, respectivamente.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,n-1) \\ f(1,0) & f(1,1) & \dots & f(1,n-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(m-1,0) & f(m-1,1) & \dots & f(m-1,n-1) \end{bmatrix} \quad (1)$$

Na Figura 4.1 ilustra a representação de uma imagem em tonalidade de cinza e a outra colorida. Imagens em tonalidade de cinza apresentam somente um valor em cada posição da matriz para representação dos pixels, formando a imagem, como ilustrado na Figura 4.1(a). Dada a junção das cores vermelho, verde e azul é possível formar combinações entre elas para representar as demais cores. Visto isso, imagens digitais coloridas apresentam 3 valores em cada posição da matriz, representando as cores vermelho, verde e azul, para compor a cor de um pixel na escala RGB, como ilustra a Figura 4.1(b).

97	97	96	98
97	97	97	98
97	97	98	99
96	97	98	98

(a)

234	234	234	234
192	192	192	191
118	118	118	118
233	233	234	234
189	189	189	190
116	116	117	117
232	232	233	233
185	185	186	187
113	113	114	115
230	230	231	232
181	181	182	183
110	110	111	112

(b)

Figura 4.1. (a) imagem em níveis de cinza, (b) imagem no esquema de cores Red, Green e Blue (RGB).

O objetivo de definir matematicamente a imagem é a possibilidade de manipular o seu conteúdo a fim de transformá-la ou retirar dela informações importantes. Ao vasto conjunto de operações que podemos aplicar em uma matriz que representa uma imagem denominamos processamento de imagem.

O processamento e a análise de imagens é uma ciência que permite modificar, analisar e manipular imagens digitais, originalmente contínuas, a partir de um computa-

dor. Os algoritmos de processamento de imagens são procedimentos passo-a-passo para otimizar as operações de tratamento de imagens.

A função primordial do PDI de sensoriamento remoto é a de fornecer ferramentas para facilitar a identificação e a extração da informação contidas nas imagens, para posterior interpretação. Nesse sentido, sistemas dedicados de computação são utilizados para atividades interativas de análise e manipulação das imagens brutas. O resultado desse processo é a produção de outras imagens, estas já contendo informações específicas, extraídas e realçadas a partir das imagens brutas [Silva 2001].

O sistema visual humano possui uma notável capacidade de reconhecer padrões. Contudo, ele dificilmente é capaz de processar o enorme volume de informação presente numa imagem. Vários tipos de degradações e distorções, inerentes aos processos de aquisição, transmissão e visualização de imagens, contribuem para limitar ainda mais essa capacidade do olho humano.

O processamento de imagem digital vem para remover essas barreiras, inerentes ao sistema visual humano, facilitando a extração de informações a partir de imagens. Nesse contexto, o processamento digital deve ser encarado como um estágio preparatório, embora quase sempre obrigatório, da atividade de interpretação das imagens de sensoriamento remoto.

Atualmente existem diversas bibliotecas para manipulação das operações de PDI, como *Insight Segmentation and Registration Toolkit* (ITK), *scikit-image*, *OpenCV*, entre outras. Para demonstração das operações básicas de PDI em imagens médicas utilizou-se a biblioteca *OpenCV*, visto que a mesma compreende um conjunto de técnicas e ferramentas de manipulação, análise de imagens e visão computacional, com código aberto e multiplataforma [Bradski e Kaehler 2008].

O OpenCV foi projetado para eficiência computacional e com foco forte em aplicativos em tempo real. O OpenCV está escrito em C otimizado e pode aproveitar os processadores multicore. Ele está disponível para desenvolvimento nas linguagens Python, Ruby, Matlab, e outras. Na metodologia faz-se uso da linguagem python para desenvolver soluções com a biblioteca por ser uma linguagem de alto nível simples de compreender.

4.3.1. Base de Imagem

Tudo se inicia com a captura de uma imagem, a qual, normalmente, corresponde à iluminação que é refletida na superfície dos objetos, realizada através de um sistema de aquisição. Após a captura por um processo de digitalização, uma imagem precisa ser representada de forma apropriada para tratamento computacional. Imagens podem ser representadas em duas ou mais dimensões, de acordo com o problema abordado.

Desde que Röntgen, em 1895, descobriu a existência de uma radiação até então desconhecida, capaz de atravessar materiais, ser parcial ou totalmente absorvida por eles e fixar-se em anteparos, a dinâmica dos exames médicos sofreu drásticas mudanças. Com a descoberta dos Raios-X, tornou-se possível a visualização de estruturas internas do corpo e diagnosticar anomalias.

A partir daí, o avanço na aquisição, processamento e armazenamento de imagens médicas vem permitindo o aperfeiçoamento de diagnósticos e tratamentos de doenças

de naturezas diversas. Cada modalidade de imagem médica atua de forma diferenciada. Sua aquisição visa a ressaltar as diferenças entre os diversos tipos de tecidos existentes na natureza e, especialmente, aqueles que compõem o corpo humano, fazendo com que estruturas de interesse se sobressaíam e, assim, possam ser identificadas como eventuais anomalias.

Algumas bases de imagens médicas são dispostas públicas com o objetivo de facilitar a pesquisa sólida no desenvolvimento de algoritmos computacionais para trazerem soluções auxílio do diagnóstico de patologias. Exemplos de bases são a RIM-ONE, DRIONS-DB, DRISTHI GS1 para o glaucoma, as bases DDSM, MIAS e IRMA para o câncer de mama e a base ISIC para o melanoma, que é o câncer de pele. Na Figura 4.2(a) tem-se um exemplo de uma imagem médica da base ISIC, na Figura 4.2(b) uma imagem da base DDSM, e na Figura 4.2(c) uma imagem da base DRISTHI.

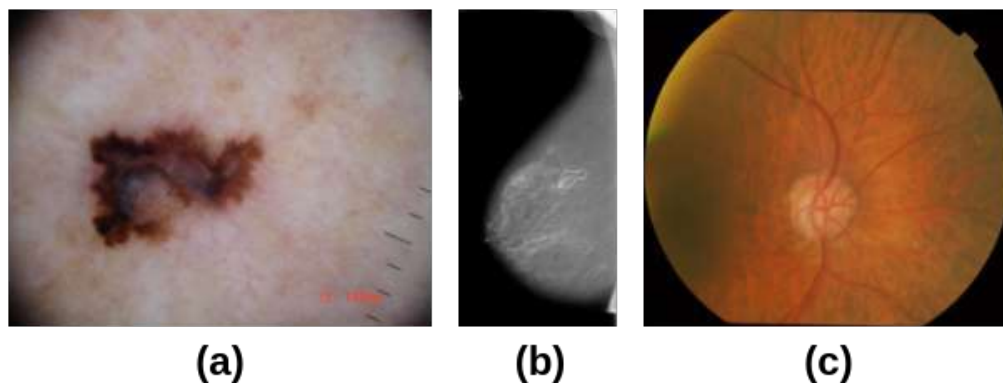


Figura 4.2. Exemplos de Imagens Médicas.

Para realização dos testes práticos, será utilizado imagens da base RIM-ONE, por a mesma possuir imagens que podem ser aplicadas todas as etapas de PDI para o desenvolvimento de um sistema CAD/CADx. Para a concretização deste trabalho, foram utilizadas 60 imagens, sendo 30 normal e 30 com presença de glaucoma. Todas as lesões no banco de dados são demarcadas por médicos especialistas.

4.3.2. Pré-Processamento

As técnicas de pré-processamento têm a função de melhorar a qualidade da imagem. Estas técnicas envolvem duas categorias principais: métodos que operam no domínio espacial e métodos que operam no domínio da frequência. Técnicas de processamento no domínio espacial baseiam-se em filtros que manipulam o plano da imagem, enquanto que as técnicas de processamento no domínio da frequência se baseiam em filtros que agem sobre o espectro da imagem. É comum para realçar determinadas características de uma imagem, combinar vários métodos que estejam baseados nestas duas categorias.

4.3.3. Segmentação

A segmentação de imagem é um processo importante para a maioria das tarefas de análise de imagens médicas. Ter uma segmentação boa irá beneficiar clínicos e pacientes, pois fornecem informações importantes para visualização, planejamento cirúrgico e detecção

precoce da doença [Ng et al. 2006].

Para o desenvolvimento de sistemas CAD/CADx é necessário que as imagens passem por uma etapa de segmentação. Segmentar, em processamento digital de imagem, significa subdividir uma imagem em suas partes ou objetos constituintes. A segmentação envolve a divisão de imagens em subseções que são de particular interesse, como a definição de áreas de uma imagem apropriada para subsequentemente serem analisadas ou a descoberta de círculos, linhas ou outras formas de interesse.

A segmentação pode parar quando esses objetos de interesse foram isolados. Os algoritmos de segmentação para imagens monocromáticas geralmente são baseados na descontinuidade das intensidades de imagem, como bordas em uma imagem ou em semelhanças julgadas por critérios predefinidos [Patton et al. 2006].

A segmentação automática é uma das tarefas mais difíceis em processamento de imagens. Esse passo determina o eventual sucesso ou fracasso na análise. De fato, a segmentação efetiva quase sempre garante sucesso no reconhecimento. Por essa razão, um cuidado considerável deve ser tomado para se melhorar as chances de uma segmentação robusta [Gonzalez e Woods 2000].

No *OpenCV* a função utilizada para segmentação é *cv2.threshold*. O primeiro argumento é a imagem de origem, que deve ser uma imagem em escala de cinza. O segundo argumento é o valor de limiar que é usado para classificar os valores de pixel. O terceiro argumento é o *maxVal* que representa o valor a ser dado se o valor do pixel for maior que (às vezes menor do que) o valor do limite.

Considerando uma imagem bimodal, em palavras simples, a imagem bimodal é uma imagem cujo histograma tem dois picos. Para essa imagem, podemos tirar aproximadamente um valor no meio desses picos como valor limiar. Isso é o que a binarização *Otsu* faz. Então, em palavras simples, ele calcula automaticamente um valor limiar do histograma de imagem para uma imagem bimodal. Para imagens que não são bimodais, a binarização não apresenta uma limiarização precisa.

O Código Fonte 4.1 mostra os passos para realizar a segmentação de imagens da retina baseado no limiar de Otsu. Para isso, a função *cv2.threshold()* é usada, mas passa um método extra, *cv2.THRESH_OTSU*. Para o valor limiar, simplesmente passa zero. Em seguida, o algoritmo encontra o valor de limiar ideal e retorna como a segunda saída, *retVal*.

```

1 import glob as g # pacote para obter path de arquivos em um diretorio
3 import cv2 # opencv
  from descritor_textura import extrair_caracteristicas # script de
    extraao de features
5 from classificador import classificar # script de classificacao

7 def aquisicao_imagens(path, extensao):
    """
9     :param path: caminho do diretorio que contem as imagens
    :param extensao: extensao das imagens. EX: .png
11    :return: lista com o caminho de todas as imagens do diretorio
    """

```



```

13     lista = g.glob(pathname=path+'*'+extensao)
14     return lista
15
16
17 def sobrepor(imagem, mascara):
18     """
19     :param imagem: imagem origem
20     :param mascara: imagem que contem a mascara apontando a regio de
21     interesse
22     :return: a imagem sobreposta
23     """
24     for i in range(mascara.shape[0]): # percorre as linhas
25         for j in range(mascara.shape[1]): # percorre as colunas
26             if mascara[i][j] == 0:
27                 imagem[i][j] = 0
28     return imagem
29
30
31 def segmentar_com_otsu(path_img):
32     """
33     :param path_img: caminho da imagem a ser segmentada
34     :return: retorna a imagem segmentada e a imagem origem nos 3 canais
35     RGB
36     """
37
38     # le a imagem em tons de cinza
39     img = cv2.imread(path_img, 0)
40     # le a imagem nos tres canais
41     img_channel = cv2.imread(path_img)
42     # segmenta a imagem atraves de um limiar obtido pelo algoritmo OTSU
43     ret, imagem_segmentada_por_limiar = cv2.threshold(img, 127, 255,
44     cv2.THRESH_OTSU)
45     # retorna a imagem original e a segmentada
46     return img_channel, imagem_segmentada_por_limiar
47
48
49 if __name__ == '__main__':
50     # diretorio onde contem as imagens
51     dir_in = '/home/pavic/PycharmProjects/PDI-Medical/data/imagens/
52     normal'
53     # diretorio onde as imagens segmentadas e sobrepostas serao salvas
54     dir_out = '/home/pavic/PycharmProjects/PDI-Medical/data/segmentadas
55     /normal'+str('/')
56
57     # variavel para encontrar o nome da imagem
58     separador = dir_in.split('/')
59     separador = separador[len(separador)-1]
60
61     # obtem uma lista com o caminho de todas as imagens
62     caminho_imagens = aquisicao_imagens(path=dir_in+'/', extensao='.
63     png')
64
65     # ciclo que realiza o processamento
66     for i in caminho_imagens:

```

```

63     # segmenta a e sobrepoe a imagem
    img_seg = segmentar_com_otsu(path_img=i)

65     # obtem o nome da imagem
    nome_img = i.split('/')+separador+'/'[1]

67     # salva a imagem
69     cv2.imwrite(filename=dir_out+nome_img, img=sobrepoe(img_seg[0],
    img_seg[1]))

71     # realizando a extracao de caracteristicas
    extrair_caracteristicas(path_normal='/home/pavic/PycharmProjects /
    PDI-Medical/data/segmentadas/normal/',
73                             path_doente='/home/pavic/PycharmProjects /
    PDI-Medical/data/segmentadas/doente/',
                                path_arquivo_descritor='/home/pavic /
    PycharmProjects/PDI-Medical/data/descritor/descriptores.libsvm',
75                             extensao='.png')

77     #classificar
    classificar(path_arquivo_descritor='/home/pavic/PycharmProjects/PDI
    -Medical/data/descritor/descriptores.libsvm')

```

Código Fonte 4.1. Segmentação de Imagens Médicas.

O glaucoma é uma anomalia que se dá pela lesão no nervo óptico, localizado no centro da retina na estrutura do Disco Óptico (DO). Assim para ter um diagnóstico mais promissor, é necessário isolar o DO do restante da imagem para ser analisada, pois ela é a região de interesse. Na Figura 4.3 ilustra essa primeira etapa para isolamento do DO. Na Figura 4.3(a) tem-se a imagem utilizada para teste. Na Figura 4.3(b) tem-se a imagem segmentada pelo algoritmo Otsu e na Figura 4.3(c) tem-se a sobreposição da máscara com a imagem original para isolamento da região de interesse, o DO.

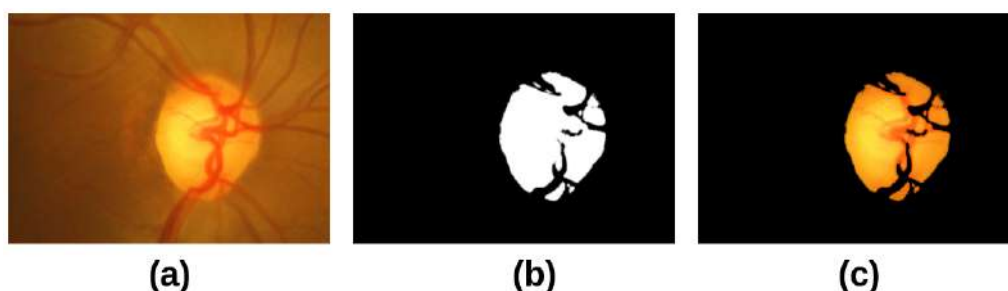


Figura 4.3. Passos para isolamento da região de interesse.

4.3.4. Extração de Características

Na análise de imagens, visão computacional e reconhecimento de padrões, os sistemas necessitam-se reconhecer elementos ou objetos de interesse nas imagens, para isso, são estabelecidos parâmetros quantificáveis, que dependem de cor, posição, orientação, dimensões, textura, para poder ser realizado computacionalmente [CONCI et al. 2008].

A extração de características é uma forma utilizada para extrair dados em uma

imagem que possa caracterizar a mesma, onde, a partir de um conjunto grande de características, seja extraída aquelas que são mais redundantes para o problema, formando um conjunto mais reduzido de características por imagem digital. A extração de características tem como objetivo mensurar e quantificar padrões perceptíveis ou não visualmente, com a finalidade de categorizar objetos nas imagens, de acordo com aspectos morfológicos de interesse.

Na maioria das metodologias CAD/CADx, a etapa de extração de informações é baseada em: 1) forma, que caracteriza a geometria dos objetos, como o tamanho, a curvatura e a suavidade dos contornos e 2) na textura, que é caracterizada por variações locais em valores de pixel que se repetem de maneira regular ou aleatória ao longo da imagem.

A extração de características foi realizado por meio da textura das imagens baseado nos índices de diversidade filogenética. A filogenia é um ramo da biologia responsável pelo estudo das relações evolutivas entre as espécies, pela verificação dos relacionamentos entre elas, a fim de determinar possíveis ancestrais comuns [Webb 2000].

Diversidade filogenética é uma medida de uma comunidade que incorpora as relações filogenéticas das espécies [Magurran 2004]. A forma mais simples da aplicação do índice de diversidade em imagens é quando a comunidade representa uma imagem ou região da mesma, as espécies sendo os níveis de cinza, os indivíduos sendo os pixels e as distâncias filogenética sendo os número de arestas entre duas espécies [Oliveira 2013]. A Tabela 4.1 mostra a correspondência entre a biologia e a metodologia.

Tabela 4.1. Correspondência entre a biologia e a metodologia.

Biologia	Metodologia
Comunidade	Região de interesse da imagem
Espécies	Níveis de cinza da imagem
Indivíduos	Pixels da imagem
Distância filogenética	Número de arestas entre duas espécies

Nesse trabalho são feitos cálculos dos seguintes atributos de textura: *Phylogenetic Diversity* - PD (Equação 2) e *Sum of Phylogenetic Distances* - SPD (Equação 3).

$$PD = B \times \frac{\sum_i^B L_i A_i}{\sum_i^B A_i} \quad (2)$$

$$SPD = \left(\frac{S(S-1)}{2} \right) \times \frac{\sum \sum m < n^2 m n^a m^a n}{\sum \sum m < n^a m^a n} \quad (3)$$

Foram utilizados os índices de diversidade filogenética para descrever a textura das regiões das imagens médicas. A análise de textura dessas regiões tem como intuito encontrar padrões que serão utilizados para categorizar as imagens médicas. O Código Fonte 4.2 mostra os passos para extração de características de textura, baseado nos índices de diversidade filogenética.

```

import glob as g
import cv2
from util.util import getMaximum, getMinimum, normalize

def geraSVMfile(rotulo, lista_feat, path_file, modo):
    """
    :param rotulo: classe que pertence as features
    :param lista_feat: lista de características
    :param path_file: diretorio do arquivo onde sera salvo as
    características
    :param modo: modo de leitura do arquivo
    :return: void
    """
    arquivo = open(path_file, modo)
    featureFile = str(rotulo) + " "
    arquivo.write(featureFile)
    for i in range(len(lista_feat)):
        linha = str(str(i + 1) + ":" + str(lista_feat[i]) + " ")
        arquivo.write(linha)
    arquivo.write('\n')
    arquivo.close()

def descrever_indices(classe, lista, path_file_descritor, modo):
    """
    :param classe: classe a qual pertence a image, 0 para doente, e 1
    para normal
    :param lista: lista com o caminho de todas as imagens a serem
    processadas
    :param path_file_descritor: caminho do arquivo descritor que
    armazena as features
    :param modo: modo de leitura do arquivo descritor
    :param size: tamanho das listas
    :param inc: incremento para calculo da porcentagem
    :return: void
    """
    prodind = 0.0
    dis = 0
    valorMPD = 0.0
    SPD = 0.0
    S1 = 0.0
    S2 = 0.0
    PD = 0.0
    somal = 0
    somatorio = 0
    MNND = 0.0
    q = 0.0

    # calcula os ndices para todas as imagens
    for i in range(len(lista)):
        img = cv2.imread(lista[i], 0)
        min = getMinimum(image=img)
        max = getMaximum(image=img)

        sizeHistograma = 256

```

```

hist = list(range(sizeHistograma))
distance = list(range(sizeHistograma))
background = min

for b in range(img.shape[0]):
    for c in range(img.shape[1]):
        value = img[b][c]
        if int(value) > 0:
            hist[value] += 1

especie = 0
for d in range(0, sizeHistograma):
    if int(hist[d]) > 0:
        especie += 1

diagonal = 0.0
foraDiagonal = 0.0

for e in range(img.shape[0]):
    for f in range(img.shape[1]):
        if int(img[e][f]) != background:
            if int(e) == int(f):
                _valor = img[e][f] + min
                diagonal += normalize(_valor)
            else:
                _valor = float(img[e][f] + min)
                foraDiagonal += normalize(_valor)

indices = []
indices.append(float(((especie * diagonal) - foraDiagonal) / (
especie * (especie - 1))))
indices.append(float(especie * indices[0]))

PSV = indices[0]
PSR = indices[1]

for x in range(0, sizeHistograma):
    for y in range(x + 1, sizeHistograma):
        if x == 0:
            distance[x] = (y - x + 1)
        else:
            distance[x] = (y - x + 2)

        prodind = (distance[x] * hist[x] * hist[y])
        somatorio += (hist[x] * hist[y])

        for q in range(x, y):
            MNND = MNND + (distance[x] * hist[q])

t = ((sizeHistograma - 1) / 2.0) #
total = sizeHistograma * t #

for x in range(0, sizeHistograma):
    for y in range(x + 1, sizeHistograma):
        if x == 0:
            dis = (y - x) + 1

```

```

106         else:
107             dis = (y - x) + 2
108
109             prodind += (dis * hist[x] * hist[y])
110             somatorio += (hist[x] * hist[y])
111
112             for indice in range(x, (y + 1)):
113                 somal += hist[indice]
114
115             Ai = somal / ((y - x) + 1)
116             S1 += (dis * Ai)
117             S2 += Ai
118
119             valorMPD += prodind
120
121         PD = S1 / S2
122         q = valorMPD / somatorio
123         SPD = total * q
124
125         feat = []
126         feat.append(PD)
127         feat.append(SPD)
128
129         geraSVMfile(rotulo=classe, lista_feat=feat, path_file=
130         path_file_descritor, modo=modo)
131
132 def extrair_caracteristicas(path_normal, path_doente,
133         path_arquivo_descritor, extensao):
134     """
135     :param path_normal: diretorio das imagens normais
136     :param path_doente: diretorio das imagens dientes
137     :param path_arquivo_descritor: diretorio do arquivo descritor
138     :param extensao: extensao das imagens, EX. .png
139     :return: void
140     """
141     lista_normal = g.glob(path_normal + '*' + extensao)
142     lista_doente = g.glob(path_doente + '*' + extensao)
143     TAM = len(lista_normal) + (len(lista_doente))
144     a = open(path_arquivo_descritor, 'w')
145     a.close()
146     descrever_indices(classe=0, lista=lista_normal, path_file_descritor
147     =path_arquivo_descritor, modo='a')
148     descrever_indices(classe=1, lista=lista_doente, path_file_descritor
149     =path_arquivo_descritor, modo='a')

```

Código Fonte 4.2. Extração de Características de textura das Imagens.

4.3.5. Classificação

Classificação consiste em reconhecer novos objetos, a partir de descritores gerados dos mesmos, tomando decisões a partir da extração de características do mundo real em imagens, isso é feito por meio de indagações simples a respeito dos parâmetros extraídos, ou por meio de algoritmos de inteligência artificial [CONCI et al. 2008].

No processamento de imagens digitais, classificar uma imagem consiste em pegar

um conjunto de características extraídas de um grupo de imagens do objeto de interesse e treinar um sistema para que possa reconhecer novos objetos a partir dessas características.

O reconhecimento de padrões ou características, usa alguns métodos ou técnicas, desde as mais simples, baseadas em vizinhos mais próximos, funções discriminantes, árvore de decisão, até as mais complexas, baseadas em redes bayesianas, lógica fuzzy e redes neurais, para a classificação de imagens de acordo com os descritores extraídos [CONCI et al. 2008].

Usualmente são utilizados algoritmos de aprendizado de máquina supervisionados, ou seja, é preciso apresentar dados pré-classificados e ensinar o algoritmo a identificar diferentes objetos. Nesse caso, é necessário uma base de dados que contenha ao menos parte dos dados previamente avaliados e rotulados por um especialista. Dessa forma o algoritmo poderá aprender os padrões dos objetos de cada classe. Os algoritmos de aprendizado de máquina supervisionados utilizado foi: *Support vector machine* (SVM) [Cortes e Vapnik 1995], *AdaBoost* [Freund et al. 1996] e *Random Forest* (RF) [Breiman 2001].

O SVM [Cortes e Vapnik 1995], faz um mapeamento do espaço de entrada para um espaço de dimensionalidade maior. Em seguida, é calculado um hiperplano de separação ótimo. O hiperplano é escolhido de modo a maximizar a distância de separação entre as classes [Haykin 1994]. Diz-se que duas classes são linearmente separáveis se existe um hiperplano divisório entre as amostras de classes diferentes.

O SVM é um algoritmo de aprendizado supervisionado, ou seja, possui as fases de treinamento e teste. Na fase de treinamento, os atributos e as classes dos objetos são as entradas do classificador e o objetivo é projetar os vetores de suporte. Os vetores de suporte são utilizados para se obter o hiperplano de separação ótimo. Na fase de teste, o hiperplano é utilizado para encontrar uma saída a partir de um vetor de entrada.

Para utilizar SVM para reconhecimento de padrões, o algoritmo requer a transformação de funções não-linearmente separáveis em funções linearmente separáveis. Para isso, é necessário que seja aumentada a dimensionalidade do problema [Cover 1965]. As funções que aumentam a dimensionalidade do espaço de entrada são chamadas funções de kernel [Haykin 1994].

O AdaBoost [Freund et al. 1996], é um meta-algoritmo de aprendizagem de máquinas. Ele pode ser usado em conjunto com muitos outros tipos de algoritmos de aprendizagem para melhorar seu desempenho. O AdaBoost é adaptável no sentido de que aprendentes fracos subsequentes são modificados em favor de instâncias mal classificadas por classificadores anteriores. O AdaBoost é sensível a dados ruidosos e outliers. Entretanto para alguns problemas é menos suscetível a perda da capacidade de generalização após o aprendizado de muitos padrões de treino (overfitting) do que a maioria dos algoritmos de aprendizado de máquina.

O AdaBoost chama um classificador fraco repetidamente em iterações $t = 1, \dots, T$. Para cada chamada a distribuição de pesos D_t é atualizada para indicar a importância do exemplo no conjunto de dados usado para classificação. A cada iteração os pesos de cada exemplo classificado incorretamente é aumentado, ou alternativamente, os pesos classificados corretamente são decrementados, para que então o novo classificador trabalhe em

mais exemplos.

O algoritmo Random Forest é uma combinação de predições de diversas árvores em que cada árvore depende dos valores de um vetor independente, amostrados aleatoriamente e com a mesma distribuição para todas as árvores da floresta. Aqui, floresta é a nomenclatura dada a uma coleção de árvores de decisão. Após a geração de um grande número de árvores, as classes com maior número de votos são eleitas [Breiman 2001].

O Random Forest divide cada nó usando o melhor dentre um subconjunto de indicadores escolhidos aleatoriamente naquele nó. Esta estratégia, apesar de um tanto contraditória, funciona adequadamente em comparação com muitos outros classificadores, além de ser robusto a superajuste nos parâmetros. Além disso, é de fácil utilização pois possui apenas dois parâmetros: o número de variáveis no subconjunto aleatório em cada nó e o número de árvores da floresta [Silva et al. 2017].

A partir de um vetor de atributos, são gerados outros vetores de atributos, que são embaralhados em relação ao vetor original. É gerado um vetor para cada árvore do *Random Forest*. Em seguida, os vetores de atributos são passados como parâmetro para as árvores de decisão. Cada árvore irá gerar um resultado para a classificação e, os resultados são combinados obtendo uma saída unificada [Silva et al. 2017].

A classificação de objetos em python pode ser realizada através da biblioteca scikit-learn. Esta biblioteca permite a utilização de algoritmos de aprendizagem de máquina, clusterização, seleção de atributos, dentre outros. As etapas básicas para classificação são: divisão do conjunto de dados em treino e teste; criação de uma instância do classificador utilizado; treino do classificador; predição utilizando os dados de teste; e cálculo da taxa de acerto seguindo alguma métrica de avaliação de desempenho. O Código Fonte 4.3 mostra as etapas de classificação de um conjunto de dados para os classificadores SVM, AdaBoost e Random Forest.

```

1 from sklearn.svm import SVC
2 from sklearn.ensemble import RandomForestClassifier
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import AdaBoostClassifier
5 from util.util import plot_confusion_matrix, confusion_matrix
6 import numpy as np
7 from matplotlib import pyplot as plt
8 from validacao import main_validacao
9
10 def get_data(path_data):
11     """
12     formata os dados do arquivo dos descritores
13     para converter em uma lista de FEATURES e LABELS
14     """
15     FEATURES = [] # lista de características
16     LABELS = [] # lista de rotulos
17     file_data = open(path_data, 'r')
18     for i in file_data:
19         lista = str(i).split(' ')
20         F = []
21         for j in range(len(lista)):
22             if j == 0:
23                 LABELS.append(lista[j])

```

```

        else:
            if lista[j] != '\n':
                value = lista[j].split(':')[1]
                F.append(value)
            FEATURES.append(F)
    return LABELS, FEATURES

def gerar_matrix_confusao(y_test, y_pred):
    # calcula a matrix de confusao
    cnf_matrix = confusion_matrix(y_test, y_pred)
    tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
    np.set_printoptions(precision=2)

    class_names = [] # rotulos para o grafico da matrix de confusao
    class_names.append('Normal')
    class_names.append('Doente')

    # grafico de matrix de confusao com dados nao normalizados
    plt.figure()
    plot_confusion_matrix(cnf_matrix, classes=class_names,
                          title='Confusion matrix, without
normalization')

    # grafico de matrix de confusao com dados normalizados
    plt.figure()
    plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=
True,
                          title='Normalized confusion matrix')
    main_validacao(tn, fp, fn, tp) # exibi os valores de acuracia,
sensibilidade e especificidade
    plt.show() # exibe o grafico

def classificar_com_SVM(X_train, X_test, y_train, y_test):
    c_svm = SVC() # cria uma instancia do modelo SVM
    c_svm.fit(X_train, y_train) # treina o modelo
    y_pred = c_svm.predict(X_test) # faz a predicao sobre os dados de
teste
    gerar_matrix_confusao(y_test, y_pred) #exibe o grafico com a matrix
de confusao

def classificar_com_RandomForest(X_train, X_test, y_train, y_test):
    c_rf = RandomForestClassifier()
    c_rf.fit(X_train, y_train)
    rf_pred = c_rf.predict(X_test)
    gerar_matrix_confusao(y_test, rf_pred)

def classificar_com_AdaBoostClassifier(X_train, X_test, y_train, y_test
):
    csf_ad = AdaBoostClassifier()
    csf_ad.fit(X_train, y_train)
    prediction = csf_ad.predict(X_test)
    gerar_matrix_confusao(y_test, prediction)

```

```

73
75
77 def model(LABELS, FEATURES):
79     TRAIN = 0.2
81     TEST = 1 - TRAIN
83
85     # formata os dados de treino e teste
86     X_train, X_test, y_train, y_test = train_test_split(FEATURES,
87     LABELS, test_size=TEST)
88
89     classificar_com_SVM(X_train, X_test, y_train, y_test) #
90     classificador SVM
91     classificar_com_RandomForest(X_train, X_test, y_train, y_test) #
92     classificador Random Forest
93     classificar_com_AdaBoostClassifier(X_train, X_test, y_train, y_test
94     ) # classificador AdaBoostClassifier
95
96 def classificar(path_arquivo_descritor):
97     # obtém a lista de labels e características
98     L, F = get_data(path_arquivo_descritor)
99     # cria o modelo, classifica e gera as matrizes de confusão
100     model(L, F)

```

Código Fonte 4.3. Classificação de imagens utilizando os classificadores Random Forest, AdaBoost e Máquina de Vetor de Suporte.

4.3.6. Validação

A fim de considerarmos a presença ou ausência de anomalias em imagens médicas, para validação dos resultados, utiliza-se de métricas de avaliação baseadas em estatísticas como, Sensibilidade [Martinez et al. 2003], Especificidade [Martinez et al. 2003] e Acurácia [Metz 1986].

A matriz de confusão oferece uma hipótese das medidas efetivas do modelo de classificação, mostrando o número de classificações corretas versus as classificações preditas para cada classe, sobre um determinado conjunto de exemplo, como mostra a Tabela 4.2.

Tabela 4.2. Matriz de Confusão

Resultado do Teste	Doença	
	Presente	Ausente
Positivo	Verdadeiro Positivo - VP	Falso Positivo - FP
Negativo	Falso Negativo - FN	Verdadeiro Negativo - VN

A sensibilidade (S), Equação 4, é a capacidade de um teste diagnóstico identificar os verdadeiros positivos nos indivíduos verdadeiramente doentes. Quando um teste é sensível, raramente deixa de encontrar pessoas com a doença.

$$S = \frac{VP}{VP + FN} \quad (4)$$

Especificidade (E), Equação 5, é a capacidade de um teste diagnóstico identificar os verdadeiros negativos nos indivíduos verdadeiramente sadios. Quando um teste é específico, raramente cometerá o erro de dizer que pessoas sadias são doentes.

$$E = \frac{VN}{VN + FP} \quad (5)$$

Acurácia (A), Equação 6, é a proporção de acertos, ou seja, o total de verdadeiramente positivos e verdadeiramente negativos, em relação a amostra estudada.

$$A = \frac{VP + VN}{VP + VN + FP + FN} \quad (6)$$

O Código Fonte 4.4 mostra as etapas de validação dos resultados para Acurácia, Sensibilidade e Especificidade, que são métricas de avaliação baseadas em estatísticas.

```

2 def acuracia(vn, fp, fn, vp):
    return (vp+vn) / (vp+vn+fp+fn)
4
6 def especificidade(vn, fp):
    return (vn) / (vn+fp)
8
10 def sensibilidade(fn, vp):
    return (vp) / (vp+fn)
12
14 def main_validacao(vn, fp, fn, vp):
    """
16     :param vn: verdadeiros positivos
18     :param fp: falsos positivos
20     :param fn: falsos negativos
22     :param vp: verdadeiros positivos
    :return:
    """
    print(' Acuracia_____ [ %.2f%s ]' %((acuracia(vn, fp, fn, vp)
    )*100), "%"))
    print(' Especificidade_____ [ %.2f%s ]' %((especificidade(vn, fp)
    )*100), "%"))
    print(' Sensibilidade_____ [ %.2f%s ]' %((sensibilidade(fn, vp)
    )*100), "%"))

```

Código Fonte 4.4. Validação da Classificação.

4.3.7. Resultados usando a base RIN-ONE

Nesta seção, são apresentados os resultados obtidos com a metodologia proposta. O sistema CAD/CADx ideal tem um bom equilíbrio entre as três métricas utilizadas para avaliação (acurácia, sensibilidade e especificidade), uma vez que uma boa metodologia deve ser capaz de classificar com sucesso os casos normal e doentes.

As imagens utilizada na pratica foram dividido em 70% treinamento, constituído por 41 imagens e 30% das imagens para teste, constituído por 19 imagens. A Tabela 4.3

mostra os resultados da classificação obtidos para Acurácia, Sensibilidade e Especificidade. O melhor caso foi obtido usando o *RandomForest* e o *Adaboost*, que obtiveram uma acurácia de 100%, já o pior caso foi obtido com o *SVM*, apresentando uma acurácia de 42,11% na classificação de imagens da retina em normal ou glaucomatosa. A Figura 4.4 mostra a matriz de confusão obtido para o *SVM*, a Figura 4.5 mostra a matriz de confusão para o *RandomForest* e a Figura 4.6 mostra a matriz de confusão para o *Adaboost* após a classificação.

Tabela 4.3. Resultados da Classificação.

Classificador	Acurácia (%)	Sensibilidade (%)	Especificidade (%)
SVM	42,11	0	100
RandomForest	100	100	100
Adaboost	100	100	100

De acordo com a Tabela 4.3, o SVM não apresentou uma taxa de acurácia muito boa, caracterizando todas as imagens como glaucomatosas. O *RandomForest* e o *Adaboost* conseguiram de forma eficiente, caracterizar as imagens em normal e glaucomatosas.

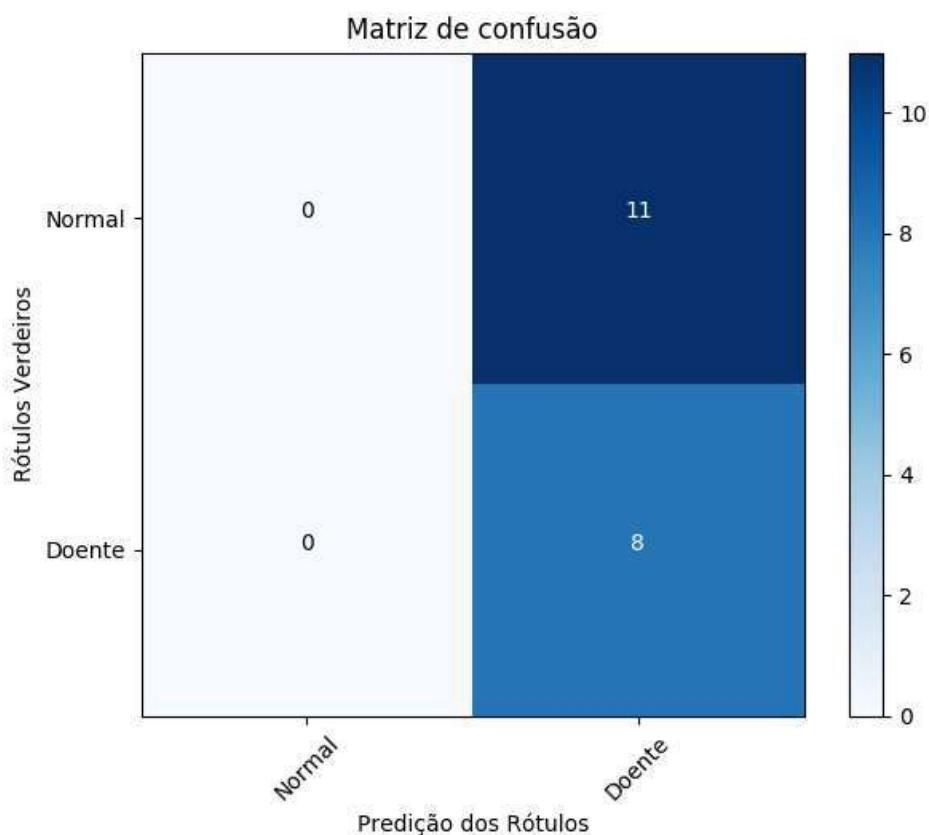


Figura 4.4. Matriz de Confusão da classificação do SVM.

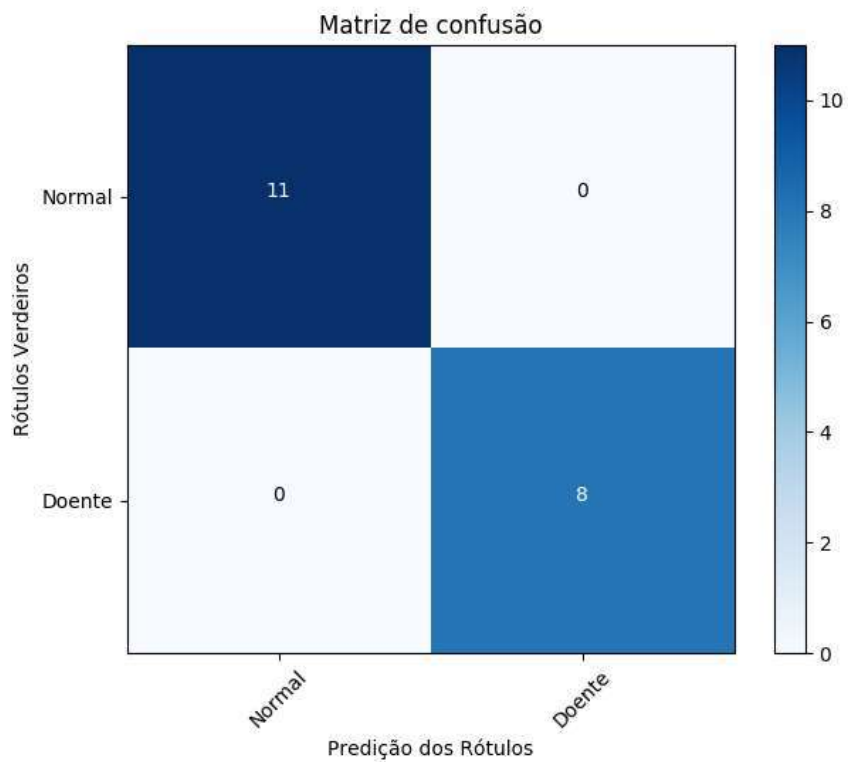


Figura 4.5. Matriz de Confusão da classificação do RandomForest.

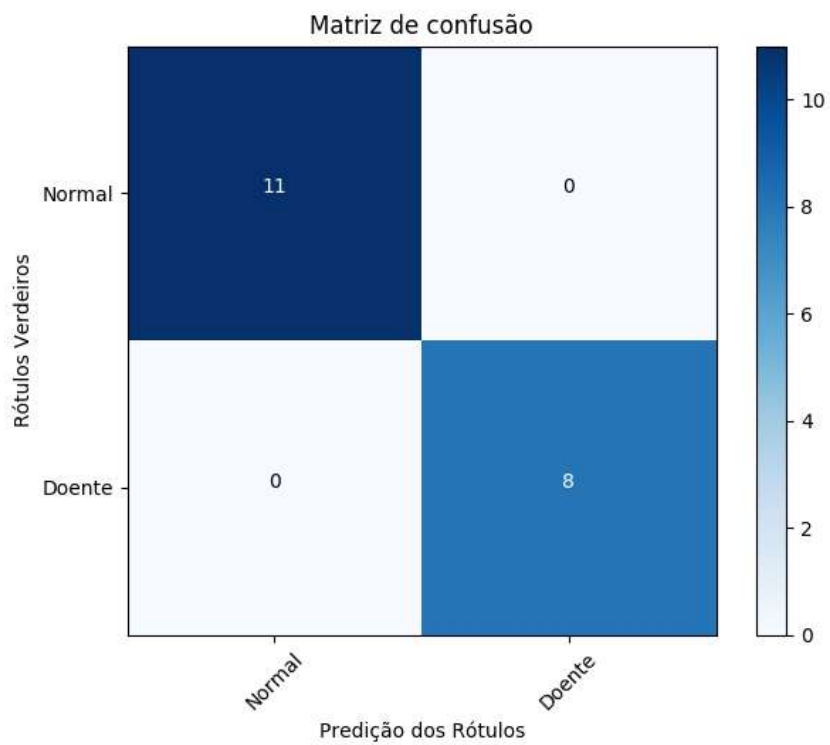


Figura 4.6. Matriz de Confusão da classificação do AdaBoost.

4.4. Conclusão

Neste capítulo foram mostrados conceitos de processamento de imagens e de visão computacional. Os exemplos descritos foram implementados utilizando a biblioteca OpenCV desenvolvida pela Intel. O OpenCV facilita a implementação de operadores simples até o desenvolvimento de sistemas mais complexos na área de processamento de imagens e visão computacional.

O primeiro desafio a ser considerado na construção de sistemas CAD/CADx é obter uma alta taxa de acerto de casos verdadeiros-positivos, isto é, indicar a presença de uma doença quando ela realmente existe, com uma baixa taxa de casos falsos-positivos, quando o sistema indica a existência de uma doença, sem esta de fato existir.

Para se obter um bom desempenho em relação aos acertos, é necessário conhecer profundamente o problema a ser detectado e a imagem médica que será utilizada para o diagnóstico, considerando seu processo de formação, que exerce influência nas suas características.

A utilização do OpenCV fornece enormes vantagens ao desenvolver sistemas que incluem segmentação, quantificação e visualização de imagens médicas. Destacadamente, a grande velocidade na implementação de esquemas de visualização complexos é um importante diferencial, pois agiliza processos que, se fossem desenvolvidos desde o começo, levariam muito tempo. Além disso, a considerável quantidade de algoritmos e técnicas que existe na biblioteca é de grande utilidade na resolução dos mais diversos problemas em processamento digital de imagens e/ou visualização científica.

Referências

- [Bradski e Kaehler 2008] Bradski, G. e Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. "O'Reilly Media, Inc."
- [Breiman 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- [Câmara et al. 1996] Câmara, G., Souza, R. C. M., Freitas, U. M., e Garrido, J. (1996). Spring: Integrating remote sensing and gis by object-oriented data modelling. *Computers & graphics*, 20(3):395–403.
- [Carvalho et al. 2017] Carvalho, E. D., Carvalho Filho, A. O., Sousa, A. D., Barros, P. V. S., e Drumond, P. M. L. L. (2017). Diferenciação de padrões de benignidade e malignidade em tecidos da mama baseado na diversidade taxonômica. *17 WIM - Workshop de Informática Médica / XXXVII Congresso da Sociedade Brasileira de Computação*, 37:1911–1920.
- [Carvalho Filho 2016] Carvalho Filho, A. O.; Silva, A. C. P. A. C. G. R. A. N. M. (2016). Lung-nodule classification based on computed tomography using taxonomic diversity indexes and an svm. *Journal of Signal Processing Systems for Signal, Image, and Video Technology*, 83.
- [Chan et al. 1990] Chan, H.-P., Doi, K., Vybrony, C. J., Schmidt, R. A., Metz, C. E., Lam, K. L., Ogura, T., Wu, Y., e MacMahon, H. (1990). Improvement in radiolo-

- gists' detection of clustered microcalcifications on mammograms: The potential of computer-aided diagnosis. *Investigative radiology*, 25(10):1102–1110.
- [Claro 2015] Claro, M. L.; Araújo, F. H. D. (2015). Uso de classificadores para a detecção automática do glaucoma.
- [CONCI et al. 2008] CONCI, A., AZEVEDO, E., e LETA, F. L. (2008). *Computação Gráfica: Teoria e Prática.*, volume 2. Elsevier.
- [Cortes e Vapnik 1995] Cortes, C. e Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- [Cover 1965] Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334.
- [de Azevedo-Marques 2001] de Azevedo-Marques, P. M. (2001). Diagnóstico auxiliado por computador na radiologia. *Radiologia Brasileira*, 34(5):285–293.
- [Doi et al. 1991] Doi, K., Giger, M., MacMahon, H., Hoffmann, K., Katsuragawa, S., Nishikawa, R., Yoshimura, Y., Sanada, S., Chen, X., Metz, C., et al. (1991). Computer-aided diagnosis: present and future. by H. Abe, K. A tsumi, T. linuma, M. Saito and M. Inoue, Elsevier Science Publishers BV, Amst erdam, pages 59–66.
- [Ellis et al. 1993] Ellis, I., Galea, M., Locker, A., Roebuck, E., Elston, C., Blamey, R., e Wilson, A. (1993). Early experience in breast cancer screening: emphasis on development of protocols for triple assessment. *The Breast*, 2(3):148–153.
- [Freund et al. 1996] Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156.
- [Giger e MacMahon 1996] Giger, M. e MacMahon, H. (1996). Image processing and computer-aided diagnosis. *Radiologic Clinics of North America*, 34(3):565–596.
- [Gonzalez e Woods 2000] Gonzalez, R. C. e Woods, R. E. (2000). *Processamento de imagens digitais*. Edgard Blucher.
- [Haykin 1994] Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.
- [Magurran 2004] Magurran, A. E. (2004). *Measuring biological diversity*. African Journal of Aquatic Science; v. 29, n. 2, p. 285-286.
- [Martinez et al. 2003] Martinez, E. Z., Louzada-Neto, F., e Pereira, B. d. B. (2003). A curva roc para testes diagnósticos. *Cad. saúde colet.*, (Rio J.), 11(1):7–31.
- [Metz 1986] Metz, C. E. (1986). Roc methodology in radiologic imaging. *Investigative radiology*, 21(9):720–733.

- [Ng et al. 2006] Ng, H., Ong, S., Foong, K., Goh, P., e Nowinski, W. (2006). Medical image segmentation using k-means clustering and improved watershed algorithm. In *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pages 61–65. IEEE.
- [Oliveira 2013] Oliveira, F. S. S. (2013). *Classificação de tecidos da mama em massa e não-massa usando Índice de diversidade taxonômico e máquina de vetores de suporte*. Dissertação de Mestrado. Curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão. São Luís - MA.
- [Patton et al. 2006] Patton, N., Aslam, T. M., MacGillivray, T., Deary, I. J., Dhillon, B., Eikelboom, R. H., Yogesani, K., e Constable, I. J. (2006). Retinal image analysis: concepts, applications and potential. *Progress in retinal and eye research*, 25(1):99–127.
- [Petrick et al. 1996] Petrick, N., Chan, H.-P., Sahiner, B., e Wei, D. (1996). An adaptive density-weighted contrast enhancement filter for mammographic breast mass detection. *IEEE Transactions on Medical Imaging*, 15(1):59–67.
- [Silva 2001] Silva, A. (2001). Curso processamento digital de imagens de satélite. XX CBC–68p.
- [Silva et al. 2017] Silva, R. R. V., Lopes, J. G. F. ., Araújo, F. H. D., Medeiros, F. N. S., e Ushizima, D. M. (2017). *Visão computacional em python utilizando as bibliotecas scikit-image e scikit-learn*, volume 1. III Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos, first edition.
- [Sousa et al. 2011] Sousa, U. S. et al. (2011). Classificação de massas na mama a partir de imagens mamográficas usando índice de diversidade de shannon-wiener.
- [Webb 2000] Webb, C. O. (2000). *Exploring the phylogenetic structure of ecological communities: an example for rain forest trees*. The American Naturalist 156.2, p. 145-155.