UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Avaliação entre a correlação de atributos das áreas alteradas em RMs de pulmão, bem como a concentração de marcadores inflamatórios no sangue de pacientes com COVID-19

Alisson Roberto Gomes

São Carlos – SP
2022

Alisson Roberto Gomes

Avaliação entre a correlação de atributos das áreas alteradas imagens de Ressonância Magnética de pulmão, bem como a concentração de marcadores inflamatórios no sangue de pacientes com COVID-19

Trabalho de conclusão de curso apresentado ao Programa de Graduação em Engenharia de Computação, do Departamento de Computação da Universidade Federal de São Carlos, como requisito parcial à obtenção do título de Bacharel em Engenharia de Computação.

Orientadora: Profª. Drª. Marcela Xavier Ribeiro

São Carlos - SP
2022

RESUMO

O SARS-CoV-2, também conhecido por COVID-19, foi responsável pela pandemia mundial iniciada em 2019 a qual repercute até os dias atuais.

Neste projeto objetivou-se encontrar uma correlação entre características extraídas das regiões inflamadas presentes em ressonâncias magnéticas pulmonares e indicadores inflamatórios de exames de sangue de um conjunto de pacientes diagnosticados com COVID-19, por meio de técnicas de processamento de imagem e mineração de dados.

Dentre os métodos utilizados para realização dessa análise estão presentes mineração por correlação, regras de associação e classificação por meio de árvore de decisão.

A partir dos resultados obtidos, foi possível encontrar uma correlação entre variáveis do estudo, onde uma alta taxa dos marcadores inflamatórios ureia e creatinina no sangue se associa com uma das características das ressonâncias magnéticas pulmonares, sendo esta o número de pixels presentes nas regiões inflamadas.

Entretanto, apenas essa correlação não torna conclusiva a possibilidade da utilização das ressonâncias magnéticas como método substituto aos utilizados atualmente para o diagnóstico de pacientes com COVID-19. Outras análises como, a classificação de pacientes, usando a correlação, para identificar quem necessita realizar o exame sangue, além disso é possível criar novas análises melhorando as técnicas utilizadas para extração de dados e até optando pela extração de outros atributos das imagens.

LISTA DE FIGURAS

SUMÁRIO

1 - Introdução

1.1 - Motivação

Vírus são organismos acelulares, que são constituídos essencialmente de proteínas e material genético, podendo ser de DNA ou RNA. Os coronavírus constituem um tipo de vírus de RNA causadores de infecções respiratórias que infectam animais como aves e mamíferos, de acordo com o artigo escrito por LANA (2020). Dos tipos de coronavírus existentes, LANA (2020) também indaga que existem sete tipos reconhecidos como patógenos em humanos.

O SARS-CoV-2 é uma nova variante do coronavírus, identificada como causadora de síndrome respiratória aguda severa (do inglês, SARS), foi responsável pela pandemia de COVID-19, o qual já infectou e causou a morte de milhares de pessoas em todo o mundo desde seu surgimento, em dezembro de 2019 (SINGH *et. al.*, 2021).

A SARS é uma pneumonia viral que evolui rapidamente para insuficiência respiratória. Normalmente, pesquisadores recomendam uma combinação de testes investigativos para encontrar sua causa, posição e a gravidade. Exames preliminares, como exame de sangue e avaliação do líquido pleural, são utilizados na detecção da gravidade de infecções (CHUNG M *et. al.*, 2019). Métodos de diagnóstico por imagem são usados frequentemente para retratar doenças no pulmão, os quais podem ser examinados por médicos especialistas ou arranjos computadorizados para descobrir a severidade da pneumonia. Comparado com Raio-X, imagens de tomografia computadorizadas de pulmões são mais frequentemente utilizadas pela possibilidade da visualização 3-D (SINGH *et. al.*, 2021).

Além das técnicas mencionadas acima, imagens de ressonância magnética (RM) constituem um método essencial e complementar de diagnóstico. Segundo Singh (2021), esta técnica é utilizada em uma grande variedade de diagnósticos clínicos guiados por imagem. Ressonância magnética (RM) de pulmão progrediu de forma exponencial nos últimos anos, devido às melhorias na qualidade das imagens e a rapidez com que essas imagens são reproduzidas. A principal vantagem da RM de pulmão é a sua combinação exclusiva de avaliação estrutural e funcional em uma

única seção de imagens (HOCHHERGGER *et. al.*, 2012). Entretanto, a avaliação de RMs isoladamente não se faz o suficiente para confirmação de diagnósticos.

Este trabalho objetiva encontrar padrões de correlação entre variáveis, aplicando técnicas de mineração de dados em conjunto de informações extraídas das RMs e da base de dados de resultados de exames laboratoriais fornecidos pelo professor Henrique Pott Júnior do departamento de Medicina da UFSCar.

É possível utilizar a técnica de mineração de dados em várias áreas, inclusive na medicina, a fim de caracterizar comportamentos de pacientes, identificar terapias de sucesso para diferentes doenças, busca por padrões de novas doenças e previsão de epidemias. Por meio desta técnica, pode-se usar grandes bases de dados para criação de modelos descritivos ou preditivos, assim como descobrir padrões inovadores (ZAKY *et. al.*, 2014; AVELAR *et. al.*, 2017).

1.2 - Objetivo Geral

Neste estudo foi obtida uma base de dados a partir a junção de dados extraídos de ressonâncias magnéticas (RM) pulmonares de pacientes com COVID-19; e, uma base de dados com valores associados a marcadores inflamatórios colhidos de exames de sangue desses pacientes e informações gerais do paciente como sexo, idade e dados extraídos.

Objetiva-se, a partir da base de dados criada, extrair padrões e encontrar uma correlação entre os atributos extraídos das RMs e a concentração de marcadores inflamatórios no sangue.

1.3 - Objetivo Específico

São objetivos específicos deste trabalho:

- Estudo e aplicação de métodos de processamento de imagens, para extrair informações referentes a área de inflamação e gerar uma base de dados usando essas informações;
- Integrar em uma única base de dados os atributos extraídos das imagens, dados colhidos de exames de sangue e informações gerais do paciente;

- Estudo de técnicas de mineração de dados envolvendo correlação, associação e classificação usando a árvore de decisão para identificar padrões na base de dados gerada.

2 - Fundamentação teórica

2.1 - SARS-CoV-2

Vários coronavírus descobertos inicialmente em aves domésticas causam doença respiratória, gastrointestinal, hepática e neurológica nos animais. Três desses coronavírus foram identificados como causadores de infeção respiratória grave em humanos que pode ser fatal: SARS-CoV (SARS), Mers-CoV (MERS) e o novo SARS-CoV-2 (COVID-19) (LAI *et. al.*, 2020).

O SARS foi identificado com uma letalidade de 7% em um província de Guangdong na China, mas segundo PEERI (2020), a variante foi contida em 2003. MERS teve seu primeiro caso em 2012, como causa da síndrome respiratória do Oriente Médio e uma letalidade de 35% (AL-OMARI A. *et. al.*, 2019).

O COVID-19 foi identificado na província de Wuhan na China em dezembro de 2019 (PEERI *et. al.*, 2020). A febre é o sintoma mais comum, seguido de tosse, a afetação bilateral dos pulmões é o resultado mais encontrado nas imagens de tomografia computadorizada nos pacientes infectados (LAI *et. al.*, 2020).

A análise evolucionária sugere que o SARS-CoV-2 é uma nova variante de coronavírus que foi introduzida aos humanos por meio de animais. Baseado nas descobertas da investigação genômica e a presença de morcegos e animais vivos no mercado de peixes em Wuhan, SARS-CoV-2 pode ter sido originada dos morcegos ou de seu excremento com material contaminado na região do mercado (LAI *et. al.*, 2020).

2.2 - Análise de imagem

O processamento de imagem para análise normalmente está atrelado ao objetivo de encontrar os elementos presentes em uma imagem (T.PAVLIDIS, 1988). Uma das tarefas mais importantes para a automatização na análise de imagens é a segmentação, que tem como principal função dividir uma imagem em regiões ou objetos, com o objetivo de extrair uma área de interesse para análise.

Algoritmos de segmentação de imagem podem ser avaliados de duas maneiras: o método analítico, que avalia a segmentação diretamente através da análise de seus princípios e propriedades, e o método empírico, que julga o algoritmo de forma indireta ao ser aplicado em testes de imagem para avaliação da qualidade da segmentação (ZHANG,1996).

2.2.1 - Técnicas de segmentação de imagens

As técnicas de segmentação podem ser divididas em quatro categorias (SILVA *et. al.*, 2011):

- *Thresholding*: que é muitas vezes baseado no histograma de cores da imagem original, baseia sua segmentação na busca de pixels que estejam dentro de um intervalo de valores definidos como limiares.
- Segmentação baseada em contornos: que apoia-se na definição geométrica dos elementos da imagem, ao analisar as descontinuidades nos níveis de cinza da imagem original.
- Segmentação baseada em regiões: é levado em consideração o conteúdo dos pixels da imagem, a fim de definir regiões por inclusão utilizando pixels que possuam características em comum. A classificação por pixels caracteriza uma região pela regularidade e repetição de características.

O método mais comum é a média de valores de *thresholds* devido à sua simplicidade de implementação e propriedades intuitivas (SILVA *et. al.*, 2011). Essa técnica é considerada simples por selecionar um valor limiar, de forma manual ou automática, para dividir uma imagem em grupo de pixels com valores iguais ou superiores ao limiar definido e um grupo de pixels com valores inferiores.

Também existem abordagens mais intuitivas como *thresholding* global, técnica mais adequada para imagens bimodais. Essa técnica consiste em selecionar apenas um limiar para toda imagem. Em contrapartida, caso o limiar dependa de propriedades locais de algumas regiões da imagem se faz o uso do *threshold* local.

2.3 - Mineração de dados

Mineração de dados é um campo multidisciplinar que envolve tecnologia de base de dados, estatística e reconhecimento de padrões (WU *et. al.*, 2021). De acordo com ZHANG (2016), este campo está em constante evolução na busca de maneiras de extração de dados para análise, permitindo métodos mais convenientes para integrar os dados de uma base.

O grande fluxo de dados coletados e armazenados por grandes organizações diariamente é um reflexo do avanço tecnológico para realização dessas coletas, assim como um aumento do volume e qualidade dos dados que podem ser obtidos. Entretanto, apenas a coleta destas informações não é o bastante: a quantidade de informações armazenadas ultrapassa a habilidade técnica e a capacidade humana na sua interpretação, conforme mencionado por AMORIM (2006).

A administração de dados pode ser o diferencial no sucesso de uma empresa. AMORIM (2006) cita que a gestão de dados consiste no desenvolvimento e execução de estratégias, assim como na criação de procedimentos para garantir o gerenciamento do ciclo de vida completo dos dados em uma empresa.

A mineração de dados também se faz presente no campo médico, que de acordo com ZHANG (2016), o grande volume de dados gerados pode ter diversas aplicações, como na avaliação dos riscos de uma doença, apoio nas decisões clínicas, predição no desenvolvimento de doenças, orientação prática nos usos de remédios, fortalecimento na gestão médica e beneficiando medicina baseada em evidência.

2.3.1 - Pré-processamento dos dados

Limpar, transformar, integrar e formatar dados faz parte da fase de preparação dos dados para análise, fase essa, que também pode ser chamada de pré-processamento dos dados, na qual dados ruídos e inconsistências da base são tratados. Esta fase abrange todas as atividades para construção do conjunto de dados que serão submetidos às ferramentas de mineração, a partir do conjunto de dados inicial (AMORIM, 2006).

2.3.2 - Métodos

2.3.2.1 - Associação

Método desenvolvido para gerenciar grandes quantidades de dados, descobrindo regras de associação e correlação entre itens da base. De acordo com SETHI (2012), este método é comumente utilizado em análises de mercado com o intuito de determinar uma combinação de itens que mais irão agradar os clientes baseado em registros anteriores.

Regras de associação identificam itens que ocorrem com frequência na base, e a partir desses itens, regras de associação são criadas (WU, 2021). Para que a listagem dos itens mais frequentes sejam criadas se faz necessário o uso de algoritmos como o Apriori, que se baseia no princípio de encontrar todos os itens de um banco de dados que atendam um conjunto mínimo de restrições (KOTSIANTIS *et. al.*, 2006)

Este método também pode ser usado para criação de regras de associação preditiva para a classificação de problemas. Segundo SETHI (2012), às informações coletadas ao utilizar a regra de associação podem ajudar na tomada de decisão de uma empresa, ou na criação de uma relatório para previsão de vendas futuras e até na determinação de fraudes.

Outra área que esse método é bem disseminado é o da medicina, na qual auxilia na descoberta da correlação entre os fatores de risco de uma doença por meio de associações. Um exemplo prático da utilização de regras de associação na medicina está presente no trabalho de LI (2017), que utilizou esse método e o algoritmo Apriori para descobrir as regras de associação entre os fatores de risco de acidente vascular cerebral (AVC).

2.3.2.2 - Correlação

O método padrão utilizado para descobrir o relacionamento entre as variáveis é o método de associação, previamente discutido, entretanto há outras abordagens que podem ser usadas para atingir esse resultado.

O método de correlação é uma variante da abordagem por associação. A associação é utilizada para dados categóricos ou discretos, enquanto a correlação é utilizada para valores contínuos. Segundo HANYUN (2015), o método de correlação

utiliza a relação de Pearson para fazer a medição entre conjuntos de dados para descobrir se eles estão relacionados.

O coeficiente de correlação de Pearson pode variar entre -1 e +1, onde o sinal indica a magnitude do relacionamento, e o valor indica a força da relação. de Acordo com PARANHOS (2014), o nível de associação das variáveis é interpretado pela proximidade que o coeficiente tem com os valores -1 e +1, sendo que mais perto de +1 mais diretamente relacionados esta, enquanto o qual mais próximo de -1 estiver mais inversamente relacionado esta. Já se estiver próximo de 0 indica-se uma ausência de correlação. Uma correlação de valor zero significa que as variáveis são ortogonais entre si (PARANHOS *et. al.*, 2014).

Entretanto, é incomum que valores extremos do coeficiente sejam atingidos. Por esse motivo, faixas de gradação foram estipuladas para interpretação da magnitude dos coeficientes. Para COHEN (1988), são considerados pequenos os valores entre 0,1 e 0,29; médios valores entre 0,3 e 0,49; e grandes qualquer valor igual ou superior a 0,5.

2.3.2.3 - Classificação usando árvores de decisão

O método de árvores de decisão representa um tipo de algoritmo de aprendizado de máquina que utiliza a abordagem dividir-para-conquistar (AMORIM, 2006). De acordo com HOUVAL (2014) uma árvore de decisão é um método de classificação e regressão que tem seu resultado similar a uma árvore com galhos e folhas. Onde os galhos representam as saídas de um atributo e cada nó folha representa uma classe, o nó folha também é conhecido como nó de decisão. No método de árvore de decisão, os atributos que estão presentes nos níveis mais baixos da árvore são os que têm maior correlação com o atributo de predição.

A medicina também se beneficia deste método (WU, 2021): estudos sobre prognóstico de pacientes com câncer de pulmão (SHOUVAL, 2014), diagnóstico para pedras no rim (MOMENYAN, 2018) e os de predição do risco de parada cardíaca súbita (TOPALOGLU, 2016), mostraram a utilidade da árvore de decisão em aplicações clínicas. Ou seja, ao construir uma predição clínica exploratória para os fatores de risco e prognósticos de pacientes, o método de árvore de decisão se mostrou promissor.

3 - Metodologia

Para analisar a possibilidade da existência de uma correlação entre os marcadores inflamatórios e áreas inflamadas presentes nas ressonâncias magnéticas (RMs) pulmonares dos pacientes infectados com o COVID-19, realizou-se a análise da base de dados dos marcadores inflamatórios e das imagens das RMs, bem como seu processamento pelas técnicas de mineração de dados previamente descritas.

3.1 - Descrição da base de dados de indicadores inflamatórios

A base de dados fornecida pelo professor Henrique Junior do Departamento de Medicina da Universidade Federal de São Carlos (DMed - UFSCar), conta com 114 entradas, onde cada uma apresenta os resultados dos exames de sangue para pacientes infectados pelo COVID-19. Cada linha é dividida em idade, sexo, cor e 41 indicadores inflamatórios como as taxas de creatinina, albumina, o número que plaquetas no sangue, entre outros.

Como a base não possui uma identificação para os pacientes, foi utilizado uma coluna numérica sequencial, “Id_anon”, para associar as imagens fornecidas com seus respectivos resultados dos exames.

3.2 - Acesso às imagens

Junto ao banco de dados também foram fornecidas imagens de ressonâncias magnéticas do pulmão de cada paciente, colhidas de um determinado corte do pulmão previamente determinado pelo professor Henrique por apresentar o maior potencial na realização da análise. Essas imagens vieram no formato dcm, comumente usado na medicina, o qual permite adicionar informações extras à imagem, como o nome do paciente, médico que realizou o procedimento, data, entre outras informações.

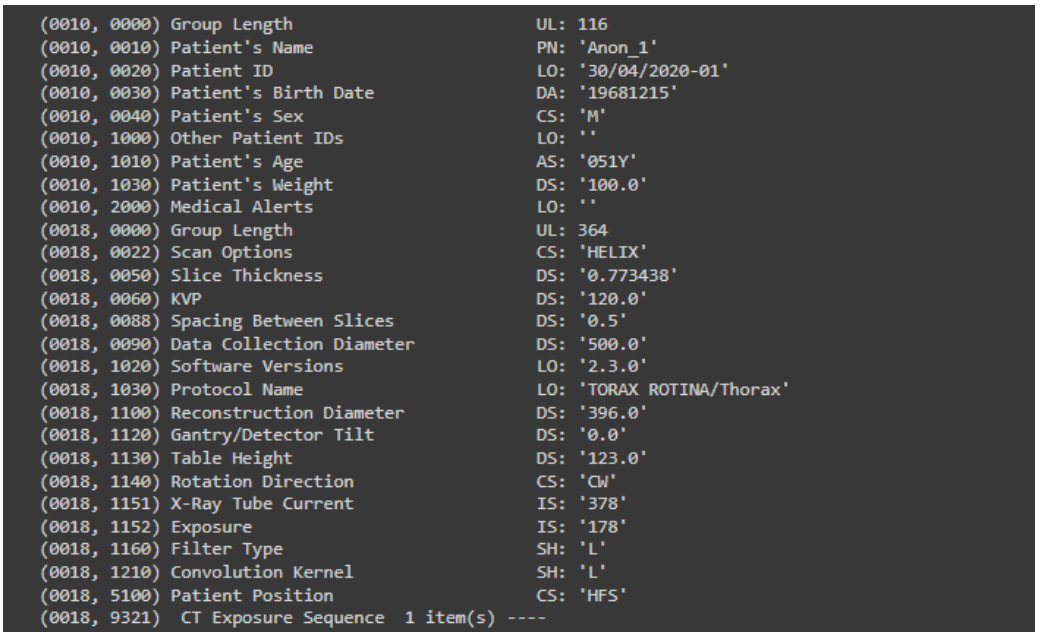
Para realizar a extração das características da imagem foi utilizado a plataforma Google Colaboratory, também conhecida como Google Colab. A linguagem de programação selecionada para a realização do projeto foi Python, considerando sua flexibilidade e o grande número de bibliotecas que facilitam o uso de métodos de processamento de imagens e mineração de dados.

Inicialmente todas as imagens fornecidas foram armazenadas em um repositório do Google Drive, pelo motivo de que a plataforma Google Colab permite o acesso dessas imagens sem a necessidade de *downloads* ou *uploads.* Esse procedimento foi realizado utilizando a biblioteca google.colab que permite que *drives* sejam acessados pela função drive.mount('/content/drive').

Com o *drive* montado, se faz possível acessar a imagens dos pacientes ao prover o caminho completo do conteúdo do *drive* até a imagem, por exemplo: ('/content/drive/MyDrive/base-Henrique e Trabalho Alisson Gomes/Anon_1/Torax - 36765/Anon_1_8260/IM-0001-0001.dcm').

Utilizando a biblioteca pydicom, a qual permite acessar figuras do tipo dcm, foi possível ler cada imagem presente e armazená-las em uma lista chamada de dicom_ds com o uso da função *dcmread()*. Onde cada membro dessa lista passou a possuir uma imagem. Por essas imagens estarem no formato dcm, é possível acessar e retornar os metadados das imagems como nome do paciente, data de nascimento, sexo do paciente utilizando uma função *printf()*, como pode ser visto na Figura 1.

Figura 1 - Metadados obtidos de uma imagem de ressonância magnética

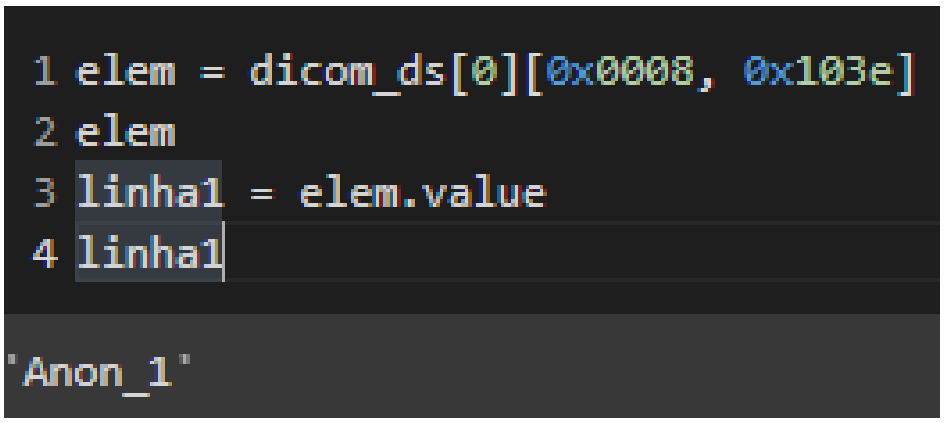


Fonte: elaboração própria

A biblioteca pydicom também permite o acesso individual das informações presentes no metadado. Para que essas informações sejam acessadas é necessário

saber a posição em que a informação se encontra e a utilização da extensão *value*, conforme demonstrado na Figura 2. A primeira linha atribui as informações na posição [0x0008, 0x103e], presente no metadado, a variável elem, enquanto as linhas três e quatro fazem com o que variável linha1 receba o elemento conteúdo presente na variável elem e o imprime respectivamente.

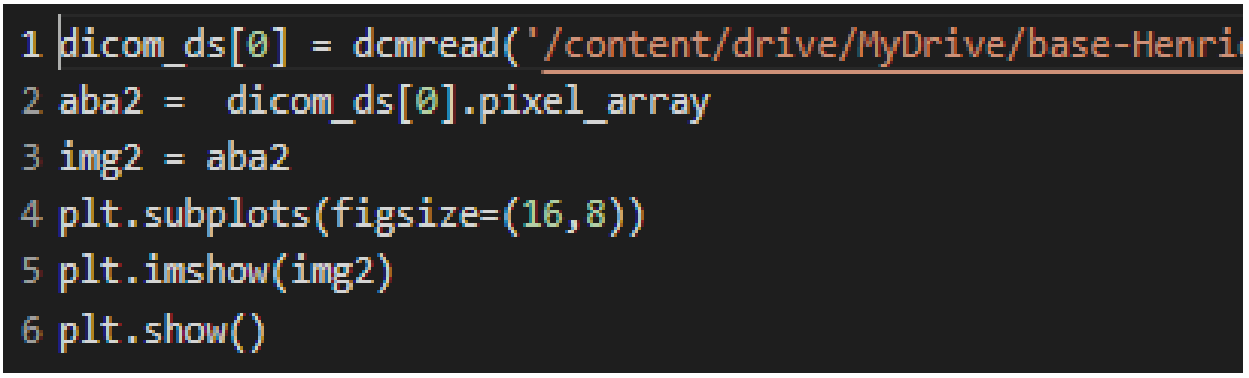
Figura 2 - Código de extração do nome do paciente



Fonte: elaboração própria

Já para retornar a imagem em si, é necessário utilizar a extensão .pixel_array junto da utilização da biblioteca matplotlib.pyplot para que imagem seja imprimida. O código presente na Figura 3 mostra o passos descritos para poder imprimir a imagem:

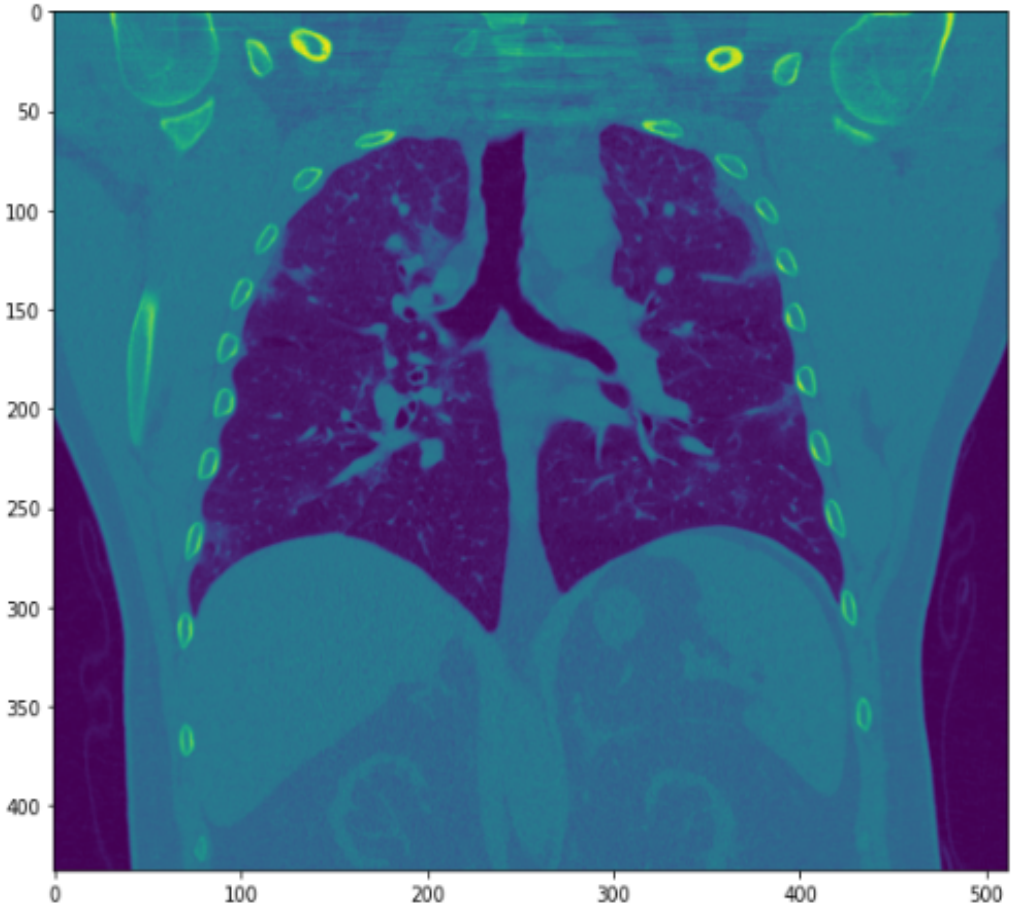
Figura 3 - Código para plotagem da imagem



Fonte: elaboração própria

Com o código finalizado é possível executá-lo para que seja visualizado na saída. Como o código mostrado refere-se a plotagem da ressonância magnética do pulmão de um paciente com COVID-19, ao executá-lo temos o retorno da Figura 4.

Figura 4 - Ressonância magnética de pulmão plotada



Fonte: elaboração própria

3.3 - Pré-processamento das imagens

Com acesso à matriz de pixels das imagens, se torna possível iniciar o tratamento das imagens para extração dos pontos de interesse. Esse pré-processamento foi realizado em duas etapas, sendo a primeira a binarização da imagem seguido da erosão da imagem.

Para a binarização da imagem foi necessário extrair tanto a altura quanto a largura de cada imagem para então criar um *loop* para verificar cada posição da matriz de pixels, ocultando tudo da imagem que não fosse de interesse. As áreas de interesse foram selecionadas utilizando o histograma equalizado da imagem.

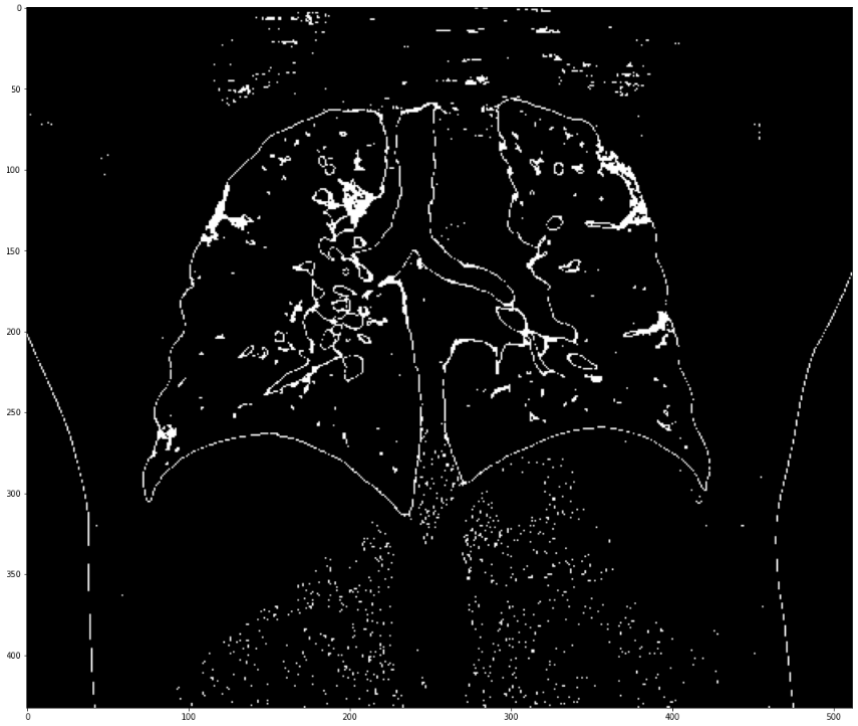
Figura 5 - Código para binarização da imagem

```
img3 =hist
altura, largura,= img3.shape
for y in range(0 ,int(altura)):
    for x in range(0 ,int(largura)):

      if  img3[y][x] <= 79  or  img3[y][x] >=90 :
        img3[y][x] =0
      else:
        img3[y][x] = 1

plt.subplots(figsize=(30,16))
plt.imshow(img3,cmap=plt.cm.gray)
plt.show()
```

Fonte: elaboração própria

O código da Figura 5 mostra um exemplo de como a binarização foi realizada, na imagem todos os pixels de valores inferiores ou iguais a 79 assim como os pixels maiores ou iguais a 90 se tornaram zero, fazendo com que tudo que recebeu esses valores fossem adicionado ao fundo preto da imagem, como é mostrado na Figura 6.

Figura 6 - Imagem de ressonância binarizada

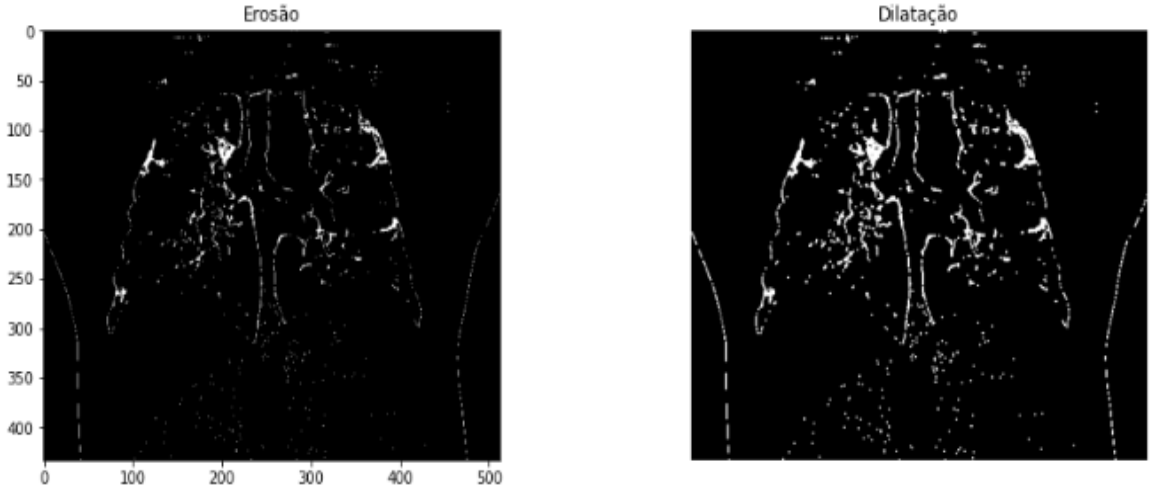


Fonte: elaboração própria

No entanto, somente com a binarização da imagem a mesma ainda apresenta ruídos. Para minimizar este problema foi utilizada a técnica de abertura de processamento de imagens que consiste na erosão da imagem: nela, uma matriz 2x2 (kernel) desliza por toda a imagem, comparando o valor de um pixel com os

valores dos pixels de sua vizinhança e atribuindo o valor 1 para pixel de saída caso todos os pixels também sejam igual a 1. Após isso, foi feita a dilatação da imagem que é o oposto da erosão, atribui-se o valor 1 ao pixel de saída caso qualquer pixels na vizinhança 2x2 seja 1. O processo de erosão e dilatação pode ser visto na Figura 7.

Figura 7 - Processo de erosão e dilatação de um imagem



Fonte: elaboração própria

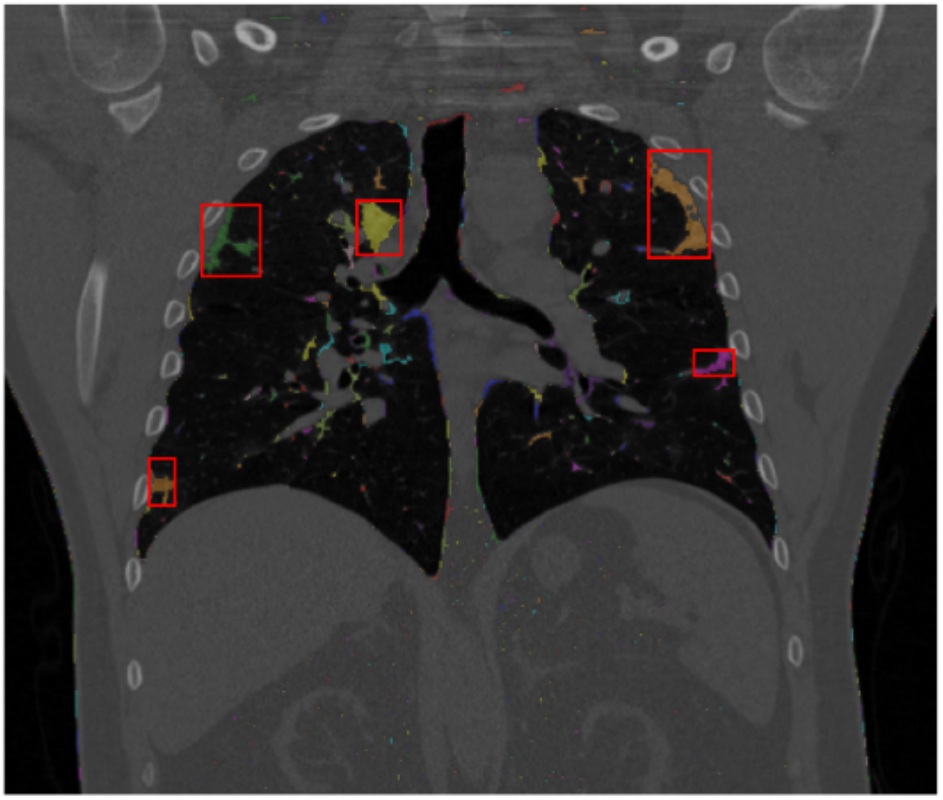
Por fim, para diferenciar as áreas de interesse da imagem, foi criada uma função para destacar essas áreas usando cor e contorno. A Figura 8 mostra essa função, que escaneia a imagem e ao encontrar uma região com área maior ou igual a indicada, nesse caso 100, cria um quadrado vermelho em volta dessa região e um preenchimento da mesma, conforme demonstrado na Figura 9.

Figura 8 - Código para diferenciar as áreas de interesse da imagem

```
for region in regionprops(label_image):
    # take regions with large enough areas
    if region.area >= 100:
        # draw rectangle around segmented coins
        minr, minc, maxr, maxc = region.bbox
        rect = mpatches.Rectangle((minc, minr), maxc - minc, maxr - minr,
                                  fill=False, edgecolor='red', linewidth=2)
```

Fonte: elaboração própria

Figura 9 - Imagem final após tratamento



Fonte: elaboração própria

3.4 - Extração das características

Para extração dos dados de cada imagem foram criadas listas vazias para armazenar os dados extraídos sempre que uma nova imagem fosse processada. Foram extraídos das regiões segmentadas: o número de regiões de interesse presentes; o somatório das áreas dessas regiões; e, o número de pixels presentes nessas regiões. As listas criadas para armazenar os valores desses atributos são apresentadas na Figura 10.

Figura 10 - Listas vazias criadas para extração de dados

```
[ ]   1 Nareas = []
      2 area = []
      3 number_of_white_pixel = []
```

Fonte: elaboração própria

A extração dos dados foi realizada junto à função criada para destacar as regiões de interesse, onde para cada região de interesse a variável *x* armazena a contagem de regiões, enquanto a variável *area1* recebe o somatório da área e a variável *number_of_white_pix1* recebe o número de pixels presentes na região. Assim que todas as regiões de interesse são avaliadas é utilizada a função *append*

para que os dados extraídos sejam integrados à lista. Esse processo é detalhado na Figura 11 onde a variável *x* representa o número de áreas de interesse encontradas na imagem; *area1*, o somatório dessas áreas; e, a variável *number_of_white_pixel1* recebe o número de pixels presentes nas regiões de interesse.

Figura 11 - Código final para destaque e extração da imagem

```
for region in regionprops(label_image):
    # take regions with large enough areas
    if region.area >= 100:
        # draw rectangle around segmented coins
        minr, minc, maxr, maxc = region.bbox
        rect = mpatches.Rectangle((minc, minr), maxc - minc, maxr - minr,
                                  fill=False, edgecolor='red', linewidth=2)
        ax.add_patch(rect)
        x = x + 1
        area1 = area1 + region.area
        number_of_white_pix1 = np.sum(bw == 1)

number_of_white_pixel.append(number_of_white_pix1)
Nareas.append(x)
area.append(area1)
```

Fonte: elaboração própria

3.5 - Base gerada

Com os dados das 114 imagens inseridas nas listas foi possível ordená-las em colunas para criação de uma base que consiste de um identificador numérico, que servirá para junção dessa base dados com a base de exames de sangue, o nome do paciente presente na imagem, o número de regiões de interesse presentes, o somatório das áreas dessas regiões e o número de pixels. A Figura 12 mostra a base de dados criada a partir dos dados extraídos.

Figura 12 - Base criada com os dados extraídos

	id_anon	Nomes	Num Area	Areas	number white pixel
0	1	Anon_1	5	1213	3905
1	2	Anon_2	6	2672	9405
2	3	Anon_3	10	2700	7894
3	4	Anon_4	9	2969	9700
4	5	Anon_5	6	1394	5134
...	...	...	...	...	...
109	110	Anon_110	5	1682	5090
110	111	Anon_111	4	987	5250
111	112	Anon_112	6	1835	11753
112	113	Anon_113	5	1214	5364
113	114	Anon_114	7	2134	6836

114 rows × 5 columns

Fonte: elaboração própria

A base de dados foi criada utilizando a função Dataframe() presente na biblioteca pandas, essa função permite com que listas de elementos que possuam o mesmo tamanho sejam reunidas em colunas, fazendo que cada elemento presente se torne uma linha na base.

3.6 - Junção das bases de dados para gerar a base integrada

Com a base de dados criada a partir do dados extraídos das imagens de ressonância magnética, o próximo passo feito foi juntar a tabela criada com a base de dados de indicadores inflamatórios.

Para a junção dessas bases foi necessário importar a tabela com as informações do exame de sangue para o Google Colab utilizando a combinação das bibliotecas google.colab, google.auth e gspread.

O código criado para importar a base de dados usa a função *open_by_key* seguido da extensão *Uniform Resource Locator* (URL) onde a planilha se encontra. A função *worksheet* serve para identificar em qual página a base se encontra. Por fim, a variável rows armazena o conteúdo da base pela função *get_all_values*, enquanto a função DataFrame.from_records recria a base dentro do Google Colab. O código descrito pode ser encontrado na Figura 13.

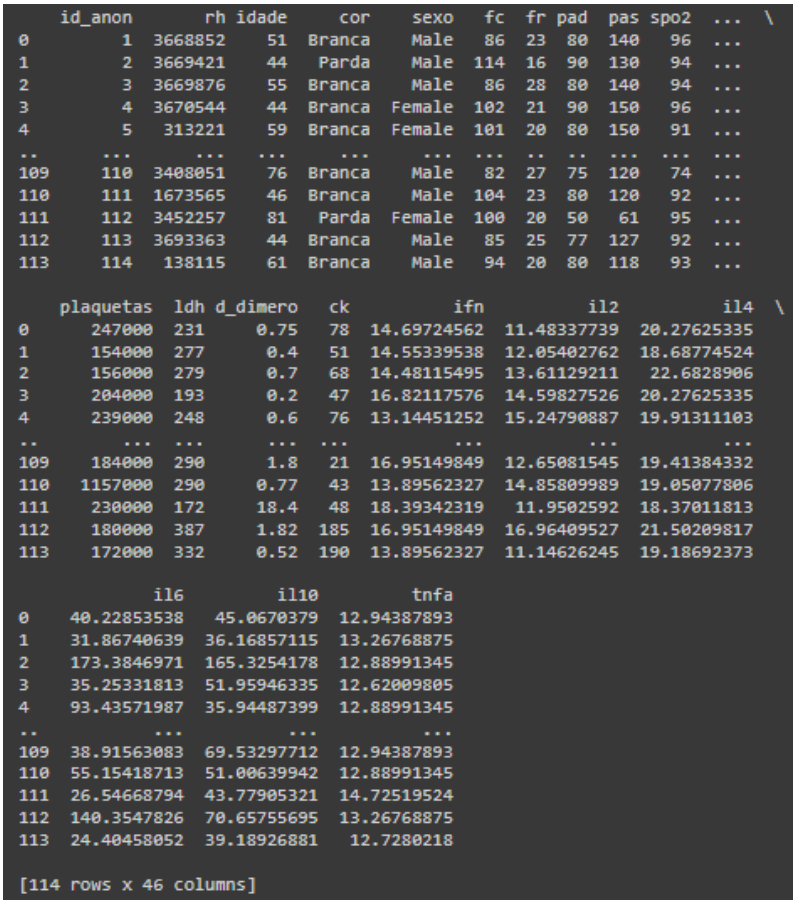
Figura 13 - Importação da base fornecida para o Google Colab

```
from google.colab import auth
auth.authenticate_user()

import gspread
from google.auth import default
creds, _ = default()

gc = gspread.authorize(creds)

wb = gc.open_by_key('1vH8XZvr6qtbxrFFOOMRjXUQaXcfF2ImlURRJRV0BHZM')
ws = wb.worksheet('Planilha1')


# get_all_values gives a list of rows.
rows = ws.get_all_values()

df = pd.DataFrame.from_records(rows[1:], columns=rows[0])
```

Fonte: elaboração própria

A tabela importada possui as informações fornecidas dos 114 pacientes, como pode ser visto na Figura 14 a seguir.

Figura 14 -Base de dados importada do Google Sheets



	id_anon	rh	idade	cor	sexo	fc	fr	pad	pas	spo2	...
0	1	3668852	51	Branca	Male	86	23	80	140	96	...
1	2	3669421	44	Parda	Male	114	16	90	130	94	...
2	3	3669876	55	Branca	Male	86	28	80	140	94	...
3	4	3670544	44	Branca	Female	102	21	90	150	96	...
4	5	313221	59	Branca	Female	101	20	80	150	91	...
..	...	...	...	...	...	...	..	..	...	...	...
109	110	3408051	76	Branca	Male	82	27	75	120	74	...
110	111	1673565	46	Branca	Male	104	23	80	120	92	...
111	112	3452257	81	Parda	Female	100	20	50	61	95	...
112	113	3693363	44	Branca	Male	85	25	77	127	92	...
113	114	138115	61	Branca	Male	94	20	80	118	93	...

	plaquetas	ldh	d_dimero	ck	ifn	il2	il4
0	247000	231	0.75	78	14.69724562	11.48337739	20.27625335
1	154000	277	0.4	51	14.55339538	12.05402762	18.68774524
2	156000	279	0.7	68	14.48115495	13.61129211	22.6828906
3	204000	193	0.2	47	16.82117576	14.59827526	20.27625335
4	239000	248	0.6	76	13.14451252	15.24790887	19.91311103
..	...	...	...	...	...	...	...
109	184000	290	1.8	21	16.95149849	12.65081545	19.41384332
110	1157000	290	0.77	43	13.89562327	14.85809989	19.05077806
111	230000	172	18.4	48	18.39342319	11.9502592	18.37011813
112	180000	387	1.82	185	16.95149849	16.96409527	21.50209817
113	172000	332	0.52	190	13.89562327	11.14626245	19.18692373

	il6	il10	tnfa
0	40.22853538	45.0670379	12.94387893
1	31.86740639	36.16857115	13.26768875
2	173.3846971	165.3254178	12.88991345
3	35.25331813	51.95946335	12.62009805
4	93.43571987	35.94487399	12.88991345
..	...	...	...
109	38.91563083	69.53297712	12.94387893
110	55.15418713	51.00639942	12.88991345
111	26.54668794	43.77905321	14.72519524
112	140.3547826	70.65755695	13.26768875
113	24.40458052	39.18926881	12.7280218

[114 rows x 46 columns]

Fonte: elaboração própria

Feito isso, foi possível utilizar a função *merge* da biblioteca pandas para realizar a junção das duas tabelas. Essa junção foi feita utilizando base fornecida como tabela principal e anexando as colunas com os dados extraídos das imagens por meio de uma coluna comum, Id_anon, entre as bases, como pode ser visto na Figura 15.

Figura 15 - Código da união das bases de dados

```
1 left_df = df
2 right_df = table
3 df3 = pd.merge(left_df, right_df, how='left', on='id_anon')
```

Fonte: elaboração própria

Com a união das duas tabelas, a planilha passa a ter 50 colunas e 114 linhas: o número de linhas permaneceu o mesmo, enquanto foram adicionadas as colunas Nomes, Num Area, Areas e *number white pixel* ao final das 46 colunas da base importada para o Google sheets. A planilha final pode ser vista na Figura 16.

Figura 16 - Resultado da união das bases

	id_anon	rh	idade	cor	sexo	fc	fr	pad	pas	spo2	...	ifn	il2	il4	il6	il10	tnfa	Nomes	Num Area	Areas	number white pixel
0	1	3668852	51	Branca	Male	86	23	80	140	96	...	14.69724562	11.48337739	20.27625335	40.22853538	45.0670379	12.94387893	Anon_1	5	1213	3905
1	2	3669421	44	Parda	Male	114	16	90	130	94	...	14.55339538	12.05402762	18.68774524	31.86740639	36.16857115	13.26768875	Anon_2	6	2672	9405
2	3	3669876	55	Branca	Male	86	28	80	140	94	...	14.48115495	13.61129211	22.6828906	173.3846971	165.3254178	12.88991345	Anon_3	10	2700	7894
3	4	3670544	44	Branca	Female	102	21	90	150	96	...	16.82117576	14.59827526	20.27625335	35.25331813	51.95946335	12.62009805	Anon_4	9	2969	9700
4	5	313221	59	Branca	Female	101	20	80	150	91	...	13.14451252	15.24790887	19.91311103	93.43571987	35.94487399	12.88991345	Anon_5	6	1394	5134
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
109	110	3408051	76	Branca	Male	82	27	75	120	74	...	16.95149849	12.65081545	19.41384332	38.91563083	69.53297712	12.94387893	Anon_110	5	1682	5090
110	111	1673565	46	Branca	Male	104	23	80	120	92	...	13.89562327	14.85809989	19.05077806	55.15418713	51.00639942	12.88991345	Anon_111	4	987	5250
111	112	3452257	81	Parda	Female	100	20	50	61	95	...	18.39342319	11.9502592	18.37011813	26.54668794	43.77905321	14.72519524	Anon_112	6	1835	11753
112	113	3693363	44	Branca	Male	85	25	77	127	92	...	16.95149849	16.96409527	21.50209817	140.3547826	70.65755695	13.26768875	Anon_113	5	1214	5364
113	114	138115	61	Branca	Male	94	20	80	118	93	...	13.89562327	11.14626245	19.18692373	24.40458052	39.18926881	12.7280218	Anon_114	7	2134	6836

114 rows × 50 columns

Fonte: elaboração própria

Por fim foi necessário alterar as variáveis do tipo *object* para *float* a fim de possibilitar com que a análises fossem feitas. Para realizar esse procedimento foi utilizado a função *astype,* presente na biblioteca pandas. A Figura 17 mostra o resultado dessa alteração onde 44 colunas passaram a possuir o Dtype *float128*.

Figura 17 - Mudança do tipo da variável para float

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 113 entries, 0 to 113
Data columns (total 44 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   fc                  113 non-null    float128
 1   fr                  113 non-null    float128
 2   pad                 113 non-null    float128
 3   pas                 113 non-null    float128
 4   spo2                113 non-null    float128
 5   tax                 113 non-null    float128
 6   news2               113 non-null    float128
 7   qsofa               113 non-null    float128
 8   sofa                113 non-null    float128
 9   creatinina          113 non-null    float128
 10  proteinas_totais    113 non-null    float128
 11  albumina            113 non-null    float128
 12  globulina           113 non-null    float128
 13  alb_glob            113 non-null    float128
 14  bil_total           113 non-null    float128
 15  bil_direta          113 non-null    float128
 16  tgo                 113 non-null    float128
 17  tgp                 113 non-null    float128
 18  fa                  113 non-null    float128
 19  ggt                 113 non-null    float128
 20  ureia               113 non-null    float128
 21  tp                  113 non-null    float128
 22  rni                 113 non-null    float128
 23  pcr                 113 non-null    float128
 24  hb                  113 non-null    float128
 25  ht                  113 non-null    float128
 26  leucocitos          113 non-null    float128
 27  bastonetes          113 non-null    float128
 28  segmentados         113 non-null    float128
 29  linfocitos          113 non-null    float128
 30  monocitos           113 non-null    float128
 31  plaquetas           113 non-null    float128
 32  ldh                 113 non-null    float128
 33  d_dimero            113 non-null    float128
 34  ck                  113 non-null    float128
 35  ifn                 113 non-null    float128
 36  il2                 113 non-null    float128
 37  il4                 113 non-null    float128
 38  il6                 113 non-null    float128
 39  il10                113 non-null    float128
 40  tnfa                113 non-null    float128
 41  Num Area            113 non-null    float128
 42  Areas               113 non-null    float128
 43  number white pixel  113 non-null    float128
dtypes: float128(44)
memory usage: 78.6 KB
```

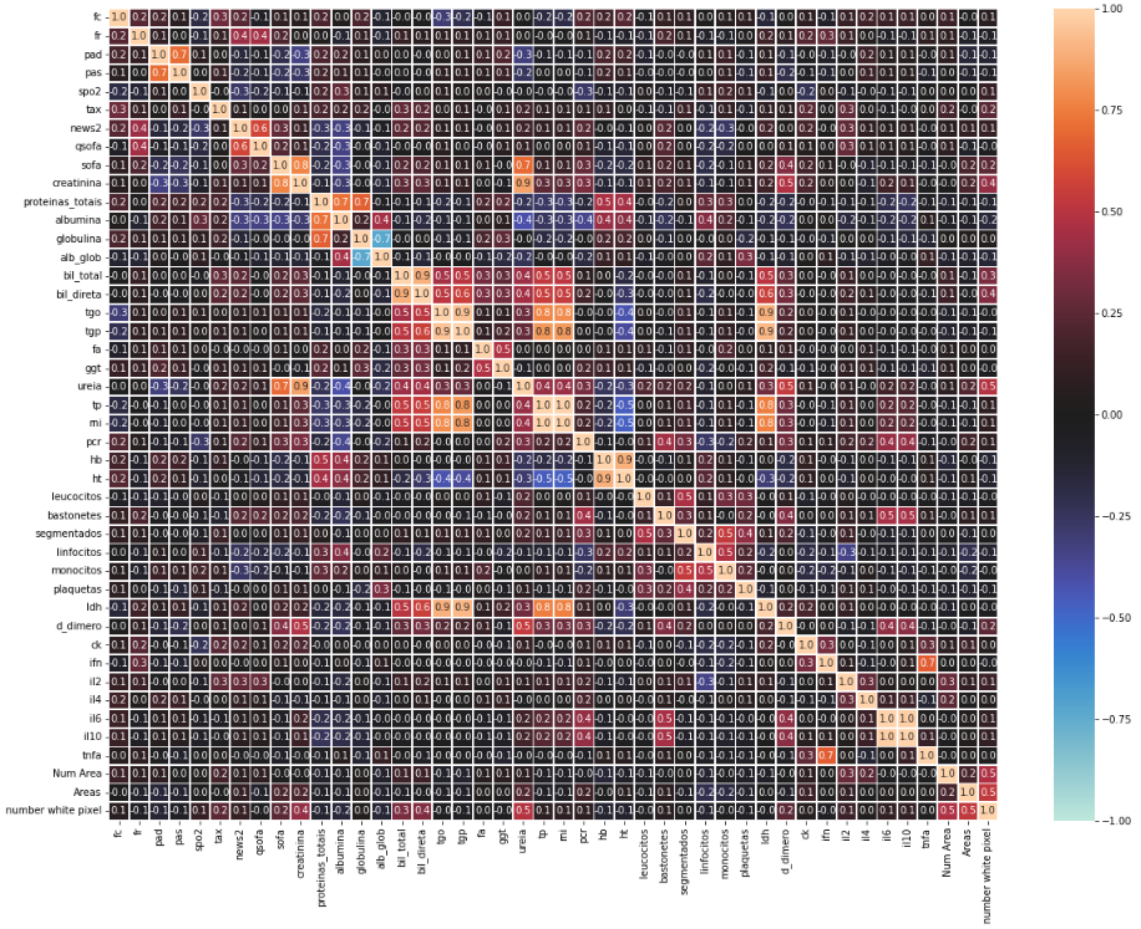
Fonte: elaboração própria

3.7 - Mineração usando correlação

Para análise de correlação foram utilizadas as bibliotecas pandas e seaborn a fim de criar uma matriz de correlação *heatmap* para visualização da força das relações entre as variáveis. A função *corr()* da biblioteca pandas foi chamada para determinar a correlação entre as variáveis, para então o gráfico ser plotado pelo método *heatmap()*.

A Figura 18 mostra o gráfico plotado com a barra lateral indicando a taxa de correlação entre variáveis por meio de sua cor, sendo que quanto mais amarelado o quadrado (+1), mais perto da correlação perfeita, enquanto preto indica uma não correlação, e o azul claro uma correlação inversa (-1), conforme estabelecido pelo coeficiente de Pearson.

Figura 18 - Gráfico de correlação entre variáveis da base unida



Fonte: elaboração própria

Utilizando a matriz de correlação da Figura 18, podemos inferir que as variáveis ureia, bil_total, bil_direta e creatinina, pertencentes à base fornecida,

possuem uma alta taxa de correlação com a variável number of white pixel extraída das imagens.

3.8 - Mineração de regras de associação

As variáveis ureia, creatinina, bil_total e bil_direta, foram utilizadas para mineração das regras de associação, usando a biblioteca mlxtend.frequent_patterns. O primeiro passo foi normalizar as variáveis utilizadas convertendo os dados para 1 ou 0, conforme taxas normais de uma pessoa adulta.

Os valores de referência para adultos do Laboratório Fleury (fleury.com.br) foram utilizados para criação das regras de normalização: considerou-se taxas normais de ureia no sangue entre 13 e 43 mg/dL, enquanto a taxa normal de creatinina no sangue fica entre 0,7 e 1,3 mg/dL. Em relação às taxas normais de bilirrubina total e direta, considera-se sendo até 0,3 mg/dL para direta e até 1,2 mg/dL para total. Sendo assim, a função transforma-se em 1 quando variável estivesse acima ou igual ao valor normal indicado e em 0 caso esse valor fosse menor. Já a normalização da variável number white pixel foi utilizado a média entre todos os valores para indicação do valor limite, como pode ser visto pelo código presente na Figura 19 .

Figura 19 - Código de padronização

```
somatoria = sum(val['number white pixel'])
num = len(val['number white pixel'])
media = somatoria/num

for x in range(len(val1)):
  #Ureia
  if val1['ureia'][x] >= 43:
    val1['ureia'][x] =  1
  else:
    val1['ureia'][x] =  0
  #Num white pixel
  if val1['number white pixel'][x] >= media:
    val1['number white pixel'][x] =  1
  else:
    val1['number white pixel'][x] =  0
  #creatinina
  if val1['creatinina'][x] >= 1.3 or val1['creatinina'][x] <= 0.7:
    val1['creatinina'][x] =  1
  else:
    val1['creatinina'][x] =  0
  #bil_total
  if val1['bil_total'][x] >= 1.2:
    val1['bil_total'][x] =  1
  else:
    val1['bil_total'][x] =  0
  #bil_direta
  if val1['bil_direta'][x] >= 0.3:
    val1['bil_direta'][x] =  1
  else:
    val1['bil_direta'][x] =  0
```

Fonte: elaboração própria

Com a discretização (binarização) pronta pode-se utilizar a função Apriori da biblioteca mlextend para descobrir a frequência de entradas no dataset. O resultado dessa função pode ser visto Figura 20 abaixo, onde *support* mostra a frequência que as variáveis aparecem no *dataset*.

Ao analisar as variáveis presentes, nota-se que os *itemsets* (number white pixel, ureia) com suporte de 0,122807, (number white pixel, creatinina) com suporte de 0,114035 e o suporte de 0,087719 para (number white pixel,ureia, creatinina), o que pode indicar um padrão onde a taxa elevada de ureia e creatinina leva a um alto número de pixels da região segmentada.

Figura 20 - Resultado da função Apriori

```
frequent_items = apriori(val1, min_support = 0.08, use_colnames=True)
frequent_items
```

	support	itemsets
0	0.412281	(ureia)
1	0.350877	(number white pixel)
2	0.377193	(creatinina)
3	0.184211	(bil_direta)
4	0.122807	(number white pixel, ureia)
5	0.245614	(creatinina, ureia)
6	0.114035	(number white pixel, creatinina)
7	0.087719	(number white pixel, ureia, creatinina)

Fonte: elaboração própria

Por fim, a função *association_rules* foi usada para mostrar os padrões entre as variáveis, como a confiança, número que expressa a possibilidade de um item ocorrer quando outro item correlato ocorre e *lift*, possibilidade de um item ocorrer em relação a outro item considerando seu suporte individual. Pela variável *lift* da Figura 21 é possível verificar um padrão entre as variáveis number white pixel, ureia, creatinina, onde o valores de *lift* maiores que 1 inferem que a presença das variáveis

na coluna de *Antecedents* leva a um aumento da chance das variáveis presentes na coluna *Consequents* ocorrerem.

Figura 21 - Regras de associação

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
6	(number white pixel, ureia)	(creatinina)	0.122807	0.377193	0.087719	0.714286	1.893688	0.041397	2.179825
11	(creatinina)	(number white pixel, ureia)	0.377193	0.122807	0.087719	0.232558	1.893688	0.041397	1.143009
7	(number white pixel, creatinina)	(ureia)	0.114035	0.412281	0.087719	0.769231	1.865794	0.040705	2.546784
10	(ureia)	(number white pixel, creatinina)	0.412281	0.114035	0.087719	0.212766	1.865794	0.040705	1.125415
2	(creatinina)	(ureia)	0.377193	0.412281	0.245614	0.651163	1.579416	0.090105	1.684795
3	(ureia)	(creatinina)	0.412281	0.377193	0.245614	0.595745	1.579416	0.090105	1.540628
8	(creatinina, ureia)	(number white pixel)	0.245614	0.350877	0.087719	0.357143	1.017857	0.001539	1.009747
9	(number white pixel)	(creatinina, ureia)	0.350877	0.245614	0.087719	0.250000	1.017857	0.001539	1.005848
4	(number white pixel)	(creatinina)	0.350877	0.377193	0.114035	0.325000	0.861628	-0.018313	0.922677
5	(creatinina)	(number white pixel)	0.377193	0.350877	0.114035	0.302326	0.861628	-0.018313	0.930409
1	(ureia)	(number white pixel)	0.412281	0.350877	0.122807	0.297872	0.848936	-0.021853	0.924508
0	(number white pixel)	(ureia)	0.350877	0.412281	0.122807	0.350000	0.848936	-0.021853	0.904184

Fonte: elaboração própria

A Figura 21, ainda ajuda na visualização das regras de associação encontradas como, a primeira linha a qual indica que a presença de valores acima da média no number white pixel e uma alta taxa de ureia levam a uma alta taxa de creatinina no sangue. Outras regras que valem ser comentadas são as presentes nas linhas três e sete, onde valores acima da média no number white pixel e uma alta taxa de creatina levam a uma alta taxa de ureia; e a alta taxa de ureia e creatinina no sangue leva à presença de uma alto valor no número de pixels presentes na segmentação.

3.9 - Classificação usando árvore de decisão

Classificação criada utilizando a biblioteca sklearn. O primeiro passo foi separar as variáveis nos tipos treino e teste, onde uma variável x recebeu os atributos das coluna ureia, creatinina, bil_direta e bil_total, e a variável y, atributo alvo, recebeu os valores da coluna number white pixel, por fim a junção *train_test_split* foi utilizada para criação das variáveis de teste e treino como pode ser visto a seguir na Figura 22.

Figura 22 - Código da criação das variáveis de teste e treino

```
X = val1[['ureia','creatinina','bil_direta','bil_total']] # Features
y = val1['number white pixel'] # Target variable

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

Fonte: elaboração própria

Após diferenciar as variáveis, a função *DecisionTreeClassifier* foi utilizada para criação do classificador da árvore de decisão e função *fit* foi utilizada para treinar esse classificador, conforme demonstrado na Figura 23.

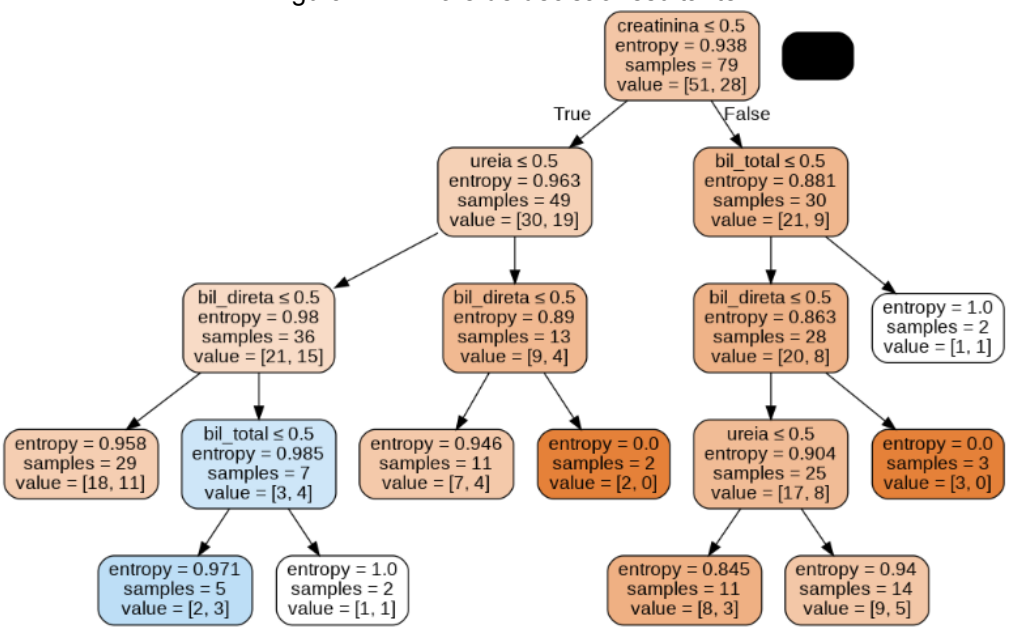
Figura 23 - Função DecisionTreeClassifier

```
# Criação do Objeto classificador de árvore de decisão e treinamento
clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)

#Predição da resposta para o conjunto de dados de teste
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Fonte: elaboração própria

Por fim a Figura 24, apresenta a árvore de decisão criada, a qual revela que os atributos creatinina e ureia são os mais influentes para determinação do rótulo, number white pixel.

Figura 24- Árvore de decisão resultante



Fonte: elaboração própria

A árvore de decisão presente na Figura 24 apresenta creatinina como raiz, indicando que das 114 entradas presentes no *dataset* 79 deles apresentam taxas de creatinina maiores que o indicado e valores de pixels presentes na segmentação

maiores que média, com uma entropia maior que 0,9 creatinina se mostra o maior influenciador para determinação do atributo *number white pixel.*

No mesmo caminho é possível constatar que a ureia também fica em segundo lugar na escala de influência, presente em 49 das 114 entradas e com uma entropia maior que 0,9 assim como a creatinina.

4 - Resultados

O *heatmap* criado pela classificação por correlação indicou uma correlação entre o número de pixels extraídos das imagens providas e as taxas de Ureia, Bilirrubina total, Bilirrubina direta e Creatinina encontradas no sangue, sendo que a taxa de Ureia tem a melhor probabilidade de apresentar um padrão.

Com as variáveis de maior correlação encontradas, foi possível realizar a discretização dessas variáveis utilizando os valores de referência do Laboratório Fleury, o que permitiu dividir esses atributos em entradas alteradas e não alteradas. Estes atributos foram utilizados em outros dois métodos de mineração de dados, com a intenção de encontrar algum padrão entre eles.

O método de regras de associação encontrou uma forte relação entre os atributos creatinina, ureia e número de pixels brancos extraídos. É possível notar na Figura 21, pela coluna *lift,* que os atributos ureia e creatinina sozinhos possuem valores menores que 1 para ocorrência do atributo number white pixel *e*xtraídos, ou seja, individualmente a taxa elevada ureia ou de creatinina não necessariamente indica uma média elevada nos número de pixels presentes nas áreas de interesse. Por outro lado, a Figura 21 mostrou que a ocorrência desses dois atributos juntos acabam elevando a possibilidade da ocorrência de um alto número de pixels da região segmentada.

Já o método de classificação por árvore de decisão mostrou que os atributos creatinina e ureia possuem uma grande influência para determinação do valor do atributo number white pixel, o que corrobora os resultados inferidos pelas regras de associação, os quais indicam que a alta taxa de ureia e creatinina no sangue estão diretamente relacionados ao elevado número de pixels presente na segmentação das imagens.

5 - Conclusões

Esse trabalho objetivou encontrar uma correlação entre os exames de sangue e as ressonâncias magnéticas de pulmão de pacientes com COVID-19. Essa correlação tem como intenção possibilitar a substituição dos exames de sangue pela análise das imagens do pulmão processadas para diagnosticar pacientes com COVID-19.

Com as técnicas de processamento de imagem e mineração de dados utilizadas, foi possível encontrar uma correlação entre as taxas de ureia e creatinina presentes no sangue, com o número de pixels presente nas áreas de interesse segmentadas das imagens de ressonância magnética, sendo a alta taxa desses atributos no sangue indicativa de que o número de pixels presentes nas regiões de interesse estará acima da média.

Entretanto, apenas essa correlação não torna conclusiva a possibilidade da utilização das ressonâncias magnéticas como método substituto aos utilizados atualmente para o diagnóstico de pacientes com COVID-19.

Outras análises podem ser conduzidas, como a utilização das ressonâncias magnéticas para classificar os pacientes e identificar quais realmente necessitam realizar o exame de sangue. Além disso, também é possível criar novas análises melhorando as técnicas utilizadas para extração de dados e até optando pela extração de outros atributos das imagens.

6 - Referências

AL-OMARI, Awad *et. al.* MERS coronavirus outbreak: Implications for emerging viral infections. **Diagnostic microbiology and infectious disease**, v. 93, n. 3, p. 265-285, 2019.

Amorim, Thiago. Conceitos, técnicas, ferramentas e aplicações de Mineração de Dados para gerar conhecimento a partir de bases de dados. **Pernambuco**. 2006

CHUNG, Michael *et. al.* CT imaging features of 2019 novel coronavirus (2019-nCoV). **Radiology**, 2020.

DE AVELAR, Cátia Fabíola Parreira; ROCHA, Thiago Augusto Hernandes; CRUZ, Flávia Juliesse Soares. Mineração de dados:: uma revisão da literatura em administração. **Revista Vianna Sapiens**, v. 8, n. 2, p. 25-25, 2017

DEY, Nilanjan *et. al.* Social group optimization–assisted Kapur's entropy and morphological segmentation for automated detection of COVID-19 infection from computed tomography images. **Cognitive Computation**, v. 12, n. 5, p. 1011-1023, 2020.

ESPOSITO, Susanna *et. al.* Levofloxacin for the treatment of Mycoplasma pneumoniae-associated meningoencephalitis in childhood. **International journal of antimicrobial agents**, v. 37, n. 5, p. 472-475, 2011.

FIGUEIREDO FILHO, Dalson Britto; SILVA JÚNIOR, José Alexandre. Desvendando os Mistérios do Coeficiente de Correlação de Pearson (r). **Revista Política Hoje**, v. 18, n. 1, p. 115-146, 2009.

HANYUN, Zhang; SHUNFANG, Hu. The Research of data mining analysis system based on Pearson relation. In: **2015 2nd International Conference on Information Science and Control Engineering**. IEEE, 2015. p. 508-511.

HOCHHEGGER, Bruno *et. al.* Magnetic resonance of the lung: a step forward in the study of lung disease. **Jornal Brasileiro de Pneumologia**, v. 38, p. 105-115, 2012

JIANG, Shibo; DU, Lanying; SHI, Zhengli. An emerging coronavirus causing pneumonia outbreak in Wuhan, China: calling for developing therapeutic and prophylactic strategies. **Emerging microbes & infections**, v. 9, n. 1, p. 275-277, 2020.1.

KOTSIANTIS, Sotiris; KANELLOPOULOS, Dimitris. Association rules mining: A recent overview. **GESTS International Transactions on Computer Science and Engineering**, v. 32, n. 1, p. 71-82, 2006.

LANA, Raquel Martins et. al. Emergência do novo coronavírus (SARS-CoV-2) e o papel de uma vigilância nacional em saúde oportuna e efetiva. **Cadernos de Saúde Pública**, v. 36, 2020.

MORAIS, N. W. S.; VIANA, N. F.; DE ABREU, H. F. G. Comparação entre as técnicas de segmentação de imagens, difração de raios xe ferritoscopia na quantificação da martensita induzida por deformação no aço AISI 301L. **Matéria (Rio de Janeiro)**, v. 16, p. 836-841, 2011.

PARANHOS, Ranulfo *et. al.* Desvendando os mistérios do coeficiente de correlação de Pearson: o retorno. **Leviathan (São Paulo)**, n. 8, p. 66-95, 2014.

PAVLIDIS, Theo. Image analysis. **Annual Review of Computer Science**, v. 3, n. 1, p. 121-146, 1988.

PEERI, Noah C. *et. al.* The SARS, MERS and novel coronavirus (COVID-19) epidemics, the newest and biggest global health threats: what lessons have we learned?. **International journal of epidemiology**, v. 49, n. 3, p. 717-726, 2020.

SETHI, Ashima; MAHAJAN, Prerna. Association rule mining: A review. **International Journal of Computer Science**, v. 1, n. 9, 2012.

SHOUVAL, R. *et. al.* Application of machine learning algorithms for clinical predictive modeling: a data-mining approach in SCT. **Bone marrow transplantation**, v. 49, n. 3, p. 332-337, 2014.

SILVA, Tatiana DCA; TAVARES, João Manuel RS. Algoritmos de segmentação de imagem e sua aplicação em imagens do sistema cardiovascular. In: **Actas do 10º Congresso Iberoamericano de Engenharia Mecânica (CIBEM 10)**. 2011.

SINGH, Asu Kumar *et. al.* COVID-19 infection detection from chest X-ray images using hybrid social group optimization and support vector classifier. **Cognitive Computation**, p. 1-13, 2021.

WU, Wen-Tao *et. al.* Data mining in clinical big data: the frequently used databases, steps, and methodological models. **Military Medical Research**, v. 8, n. 1, p. 1-12, 2021.

ZAKI, Mohammed J.; MEIRA JR, Wagner; MEIRA, Wagner. **Data mining and analysis: fundamental concepts and algorithms**. Cambridge University Press, 2014.

ZHANG, Yu Jin. A survey on evaluation methods for image segmentation. **Pattern recognition**, v. 29, n. 8, p. 1335-1346, 1996.

ZHANG, Yue *et. al.* Application and exploration of big data mining in clinical medicine. **Chinese Medical Journal**, v. 129, n. 06, p. 731-738, 2016.