

---

**Arquivos DICOM e Redes Neurais Convolucionais: Classificando a Vesícula Biliar**

Luiz Fernando Pereira Castiglioni, Rodrigo Zanin

## **Arquivos DICOM e Redes Neurais Convolucionais: Classificando a Vesícula Biliar**

### **Resumo**

Manipulação de arquivos DICOM de Tomografia e o treinamento de um modelo de Rede Neural Convolucional para detectar a presença ou a ausência da vesícula biliar.

**Palavras-chave:** CNN, Detecção de Objetos, Visão Computacional, Medicina Diagnóstica, Tomografia, DICOM, YOLO, YOLOv7, R.

### **1. Introdução**

Cada vez mais, o auxílio das máquinas nos ajuda a tomar decisões baseadas em dados. Na medicina diagnóstica um grande avanço ocasionado pelo uso dos dados é a constante criação de novas ferramentas que ajudam os médicos na qualidade e assertividade de seus diagnósticos. O uso de algoritmos de detecção de objetos em imagens de tomografia traz resultados interessantes que geram novas reflexões sobre o futuro da medicina.

Os exames de tomografia têm uma alta demanda em todo o nosso país, sendo hoje uma das principais ferramentas diagnósticas em todas as áreas da medicina. Esses exames aliados às técnicas de Aprendizado de Máquina e Inteligência Artificial, tem se tornado a solução preferíveis em diferentes campos médicos. Tudo isso aliado a um cenário tecnológico de aumento da capacidade computacional de processamento de grandes volumes de dados. O resultado são novas métricas, novos métodos, novas visões sobre o futuro e melhores decisões.

A tomografia é um exame de imagem que apresenta o corpo humano em fatias, através do raio-X das partes internas do corpo. São imagens fatiadas do corpo com, em média, 1 milímetro de espessura entre as fatias. O resultado do exame são fotos em escalas de cinza que contém os órgãos, tecidos e ossos, que facilitam a identificação visual de cada elemento que compõe a anatomia humana. A tomografia pode ser feita como um recorte horizontal e transversal do corpo. Pode produzir imagens em 2D e 3D. O exame é realizado em uma máquina chamada tomógrafo onde o paciente se deita em uma maca e o aparelho captura uma série de imagens em sequência. Como uma máquina fotográfica em Raio X. Uma das grandes vantagens é que o exame de tomografia pode armazenar várias imagens no mesmo exame em formato de séries, exibindo as imagens lado a lado ou permitindo que cada imagem seja analisada individualmente.

São frequentes os erros de diagnósticos relacionados a vesícula biliar, apesar da evolução da tomografia para diagnóstico de lesões e doenças nesse órgão. As falhas de diagnóstico da vesícula podem estar relacionadas as suas particularidades anatômicas. Erros também podem ocorrer na aquisição das imagens no aparelho de tomografia (qualidade do aparelho utilizado, protocolo de aquisição utilizado) e no processamento das imagens (protocolos de reconstrução). A "expertise" do médico que lauda o exame e a complexidade da identificação desse órgão também são fatores determinantes.

A vesícula biliar é um órgão em forma de saco, parecida com uma pera, localizada abaixo do lobo direito do fígado na região abdominal. Sua função é armazenar a bile, líquido produzido pelo fígado que atua na digestão de gorduras no intestino. A vesícula biliar tem de 7 a 10 cm de comprimento, e está conectada ao fígado e ao duodeno através do trato biliar. É um "saco biliar" que se enche e se esvazia de bile, dependendo da etapa do processo

digestivo que o corpo está realizando. Quando os alimentos (especialmente os alimentos gordurosos) estão sendo digeridos, a vesícula biliar se contrai e secreta a bile através de um pequeno tubo denominado ducto cístico, que se junta ao ducto hepático comum para formar o ducto biliar comum. O ducto biliar comum une-se ao ducto principal do pâncreas (ducto pancreático) para desembocar na primeira parte do intestino delgado (duodeno) na ampola de Vater. A vesícula biliar ajuda na digestão, mas não é indispensável para a vida.

Para detectar a presença da vesícula em uma tomografia, é necessário realizar o exame em jejum, pois assim a vesícula biliar estará cheia de bile com um tamanho próximo a 10 centímetros sendo possível identificá-la com maior facilidade. Porém, em casos de emergência, quando o paciente sofre um trauma e chega na UTI com necessidade de realizar os exames às pressas, muitas vezes esse paciente não está em jejum. Nesse caso, a vesícula estará vazia, sem líquido biliar, e seu tamanho pode chegar a menos de 3 centímetros. Isso torna a sua identificação muito mais difícil. Consequentemente ao contestar a presença da vesícula, o laudo da tomografia apontará um resultado duvidoso e facilmente contestável. Pelo fato de anatomicamente ser um órgão que varia de dimensões, a vesícula está associada a muitos erros de interpretação e dificuldades de detecção nas etapas do processo de exame tomográfico.

Quando o paciente não possui vesícula, é comum que outros órgãos se alojem no espaço deixado por ela, muitas vezes alças do intestino acabam ocupando o lugar da vesícula o que também pode ocasionar em um erro de interpretação, pois esses dois órgãos podem ser facilmente confundidos. Também é comum observar a presença de um clipe cirúrgico no local onde a vesícula foi retirada. Esse clipe é deixado durante a cirurgia de retirada da vesícula. É um clipe permanente, de fácil identificação e que ajudará nas inferências do modelo de classificação desse projeto. Pois onde há a presença do clipe não há vesícula, *ceteris paribus*.

As doenças da vesícula são um dos principais problemas de saúde pública, acometendo principalmente pessoas obesas. O diagnóstico correto do órgão é muito importante no contexto do tratamento do paciente e na definição dos prognósticos.

## **2. Material e Métodos**

O Banco de Dados utilizado será uma coleta de 200 exames de imagens de tomografia no formato DICOM. Estes exames contêm séries de imagens da região do abdômen de pacientes que podem ou não ter a vesícula biliar. O projeto utilizará a linguagem R de programação em ambiente Linux, também serão executados scripts em Python. O software utilizado neste projeto será o RStudio. Os pacotes R utilizados serão "oro.dicom", "dcmtk", "divest", "tidyverse", "dplyr", "readxl". Alguns comandos serão inseridos no Terminal e serão usadas as bibliotecas "dcmtk" e "labellmg". Será necessário clonar o repositório github da biblioteca "yolov7" desenvolvido em Python com o pacote "pytorch".

## 2.1. Análise Inicial dos arquivos DICOM



Figura 1. Tomografia DICOM da região do abdômen em 2D

O arquivo DICOM (Digital Imaging and Communication in Medicine) foi criado em 1993 e representa um marco na padronização do formato eletrônico utilizado no armazenamento e na comunicação das imagens médicas. Este formato de extensão “.dcm” foi fundamental para garantir a segurança e melhoria nos processos de análise e diagnóstico de exames.

Cada arquivo DICOM é lido pelo R como uma lista ‘list()’ contendo dois objetos: uma imagem de exame tomográfico em preto e branco e uma tabela que armazena toda a metadata e informações sensíveis aquela imagem.

A imagem (\$img) do arquivo é armazenada como uma matriz de tamanho 512 por 512. A tabela (\$hdr) contém todas as informações técnicas e apontamentos do exame colhido, como nome do paciente, idade, data, aparelho tomográfico, espessura das “fatias” do exame, número da imagem e diversos outros apontamentos. Durante o desenvolvimento desse projeto, cada um desses dois objetos recebe um tratamento de dados específicos.

Todos os arquivos DICOM e dados dessa pesquisa foram obtidos em parceria com a Escola Paulista de Medicina do Estado de São Paulo (Unifesp). Um arquivo comprimido ‘.rar’ contendo o Banco de Imagens DICOM foi obtido via download, e estava armazenado

em um drive virtual. Este arquivo '.rar' possui 38.4GB, aproximadamente 165mil arquivos DICOM com extensão ".dcm" divididos em 200 pastas, onde cada pasta representa as informações armazenadas de um paciente. Esses 200 pacientes foram selecionados a partir de exames tomográficos coletados entre 2014 e 2017 na rede pública hospitalar do estado de São Paulo e representam uma amostra com grande diversidade. Os pacientes possuem diferenças de idade, de gênero, de diagnósticos de doenças e de biotipos. E será necessário anonimizar esses exames conforme os protocolos científicos e éticos de estudos médicos. A quantidade de imagens por paciente também varia devido as diferenças entre os equipamentos tomográficos de onde os dados foram coletados. Também foi observada a diferença entre a quantidade de séries de imagens por paciente.

## 2.2. Análise da Tabela Anotada

Os 200 exames de imagens foram coletados pelo grupo de médicos e residentes do departamento de Inteligência Artificial da Escola Paulista de Medicina e em seguida foi anotada a detecção da presença ou da ausência da vesícula biliar na tabela *annot\_clean*.

```
R> read_excel("~/Desktop/USP/TCC/projeto_TCC/TABLEs/annot_clean.xlsx") %>%  
as_tibble() -> annot_clean  
R> gsub(pattern = "Não se aplica", replacement = "NA", annot_clean$`Densidade não  
habitual (sim=1, não=0)`) → annot_clean$`Densidade não habitual (sim=1, não=0)`  
R> gsub(pattern = "Não se aplica", replacement = "NA", annot_clean$`Corte INICIAL  
vesícula`) -> annot_clean$`Corte INICIAL vesícula`  
R> gsub(pattern = "Não se aplica", replacement = "NA", annot_clean$`Corte FINAL  
vesícula`) -> annot_clean$`Corte FINAL vesícula`  
R> annot_clean$`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)` <-  
as.factor(annot_clean$`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`)  
R> annot_clean$`Corte FINAL vesícula` <- as.numeric(annot_clean$`Corte FINAL vesícula`)  
R> annot_clean$`Corte INICIAL vesícula` <- as.numeric(annot_clean$`Corte INICIAL  
vesícula`)  
R> annot_clean$`Densidade não habitual (sim=1, não=0)` <-  
as.numeric(annot_clean$`Densidade não habitual (sim=1, não=0)`)
```

```
> annot_clean
# A tibble: 200 x 10
  `Acession Number` `Patient ID` Corte INICIAL vesíc...1 Corte...2 Hipod...3 Densi...4 Sem c...5 Arter...6 Portal Equil...7
      <dbl>          <dbl>      <dbl>      <dbl> <fct>      <dbl>      <dbl>      <dbl> <dbl>
1      5499993      3861927      NA      NA S          0          1          1          1
2      5493468      823083      NA      NA C          0          1          1          1
3      5480343      3586593      NA      NA S          0          0          1          0
4      5481425      2863787      NA      NA C          0          0          0          0
5      5480051      3880103      NA      NA C          0          0          0          0
6      5459110      3825619      NA      NA S          0          0          0          0
7      5465667      3132902      NA      NA C          0          1          1          1
8      5466142      3872936      NA      NA C          0          1          0          0
9      5435830      3227017      NA      NA C          0          1          1          1
10     5433748      870140      NA      NA S          0          0          0          0
# ... with 190 more rows, and abbreviated variable names 1`Corte INICIAL vesícula`, 2`Corte FINAL vesícula`,
# 3`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`, 4`Densidade não habitual (sim=1, não=0)`,
# 5`Sem contraste (sim = 1; não = 0)`, 6Arterial, 7Equilíbrio
# Use `print(n = ...)` to see more rows
```

Figura 2. Output Terminal RStudio objeto tibble annot\_clean

Fonte: 5QUINTO\_SCRIPT.R

A tabela contém 200 observações correspondente ao número de pacientes estudados e 10 variáveis. São elas:

“Acession Number” – esta variável quantitativa discreta refere-se ao número de acesso gerado pela requisição desse arquivo no servidor de onde esses exames estavam localizados.

“Patient ID” – esta variável quantitativa discreta refere-se a identidade única de cada paciente criada pelo padrão DICOM.

“Corte INICIAL vesícula” – esta variável quantitativa discreta refere-se ao número (InstanceNumber) da primeira imagem dentro de uma série onde foi identificada a presença vesícula.

“Corte FINAL vesícula” – esta variável quantitativa discreta refere-se ao número (InstanceNumber) da última imagem dentro de uma série onde foi identificada a vesícula.

“Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)” – esta variável categórica nominal refere-se a 4 categorias nas quais os pacientes foram separados a partir do resultado do laudo realizado pelos médicos:

V – Vesícula normal detectada

H – Vesícula Hipodistendida detectada

C – Clipe Cirúrgico detectado

S – Não foi detectada a presença da vesícula e nem do clipe cirúrgico.

“Densidade não habitual (sim=1, não=0)” – esta variável categórica nominal refere-se apenas aos exames onde o laudo médico detectou a presença da vesícula.

1 – Vesícula com alteração em seu tamanho e volume

0 – Vesícula com densidade normal

“Sem contraste (sim = 1; não = 0)” – esta variável categórica binária refere-se ao uso da substância de contraste no sangue do paciente. Procedimento médico que melhorar a qualidade da leitura do exame e gera 3 fases de leitura das imagens. Cada fase possui um nível de “contraste” diferente para laudar o exame. Consequentemente facilita a interpretação do exame pelo médico pois apresenta a mesma imagem com 3 distintas tonalidades de contraste.

“Arterial” – Esta variável categórica binária indica verdadeiro ou falso se o arquivo se refere a um exame coletado durante a fase arterial do contraste no sangue do paciente. Apenas para pacientes em que a substância contraste foi aplicada.

“Portal” – Esta variável categórica binária indica verdadeiro ou falso se o arquivo se refere a um exame coletado durante a fase portal do contraste no sangue do paciente. Apenas para pacientes em que a substância contraste foi aplicada

“Equilíbrio” – Esta variável categórica binária indica verdadeiro ou falso se o arquivo se refere a um exame coletado durante a fase equilíbrio do contraste no sangue do paciente. Apenas para pacientes em que a substância contraste foi aplicada.

### **2.3. Pré-processamento de Imagem: Achando soluções para os Impasses Computacionais gerados pelo processamento de grandes volumes de arquivos.**

O padrão DICOM de armazenamento de imagem provê interoperabilidade entre diferentes hardwares, mas não foi desenvolvido para facilitar uma manipulação de dados eficiente e processamentos de imagem. Como consequência, formatos adicionais foram desenvolvidos ao longo dos anos para acomodar melhor as análises e processamentos. (Whitcher et al., 2016).

Existem limitações consideráveis ao trabalharmos com os arquivos DICOM. A primeira consideração importante é o tamanho do arquivo que inicialmente tem aproximadamente 215kB cada, em um cenário de 165 mil arquivos (38.4GB do arquivo ‘.rar’ obtido). Mesmo quando falamos de um computador com grande capacidade de processamento é computacionalmente demandante processar grandes quantidades de imagens. O formato DICOM além de armazenar a imagem coletada no exame também armazena uma metadata única, complexa e detalhada. A metadata DICOM segue com rigidez todas as normas do padrão DICOM desenvolvidas em 1993.

A segunda limitação importante a ser discutida envolve a leitura do arquivo DICOM no ambiente R. O R ainda não é capaz de interpretar o arquivo DICOM em sua forma “RAW”, no formato nativo do arquivo. Portanto é necessário converter a imagem de dentro do arquivo DICOM para o formato JPEG. Esse processo duplica o tamanho do arquivo

DICOM consequentemente duplicando o tamanho do Banco de Imagens. Porém apenas com esse pré-processamento foi possível modelar os dados contidos no arquivo DICOM no RStudio.

A terceira limitação de grande importância que foi considerada nesse projeto é que o interpretador R lê, armazenada e processa o arquivo DICOM na memória RAM do computador inviabilizando qualquer análise de grandes volumes de arquivos. No caso, um computador de 16Gb de RAM só seria capaz de processar aproximadamente 12Gb de arquivos DICOM por vez, ainda sim este procedimento estaria comprometendo o funcionamento da máquina, e não há nenhuma garantia de funcionamento. Qualquer análise de grande volume de arquivos DICOM torna-se impossível dessa forma em um computador de pesquisa com configurações básicas. Quando pensamos em processar esses arquivos em um ambiente de nuvem, essas análises requerem uma configuração de máquina “pesada” em memória RAM que muitas vezes foge do orçamento de um projeto de pesquisa acadêmica. Resolver esse impasse de volume de processamento computacional talvez tenha sido o grande desafio desse projeto. Como demonstra o contados de tempo aplicado a uma amostra aleatória de arquivos DICOM:

```
R> bench_time(readDICOM("~/Documents/1.2.840.113704.1.111.3180.1469042540.1/",  
verbose = TRUE))
```

```
630 files to be processed by readDICOM()
```

```
| ===== | 100%  
process    real  
   41s   41.5s
```

```
R> paste(round(c(41*165794/630)), "segundos") # tempo total estimado (segundos) de  
processamento da base de dados DICOM com 165794 arquivos
```

```
[1] "10790 segundos"
```

```
R> paste(round(c(41*165794/630)/3600), "horas") # tempo total estimado (horas) de  
processamento da base de dados DICOM com 165794 arquivos
```

```
[1] "3 horas"
```



Para resolução desses problemas iniciais, foi estabelecida uma etapa de Pré-processamento dos arquivos. O Pré-processamento foi dividido em 3 subprocessos que conforme apresentados no Fluxograma são: Converter, Renomear e Anonimizar. Esta etapa foi desenvolvida para tornar possível a modelagem dos dados desse projeto.

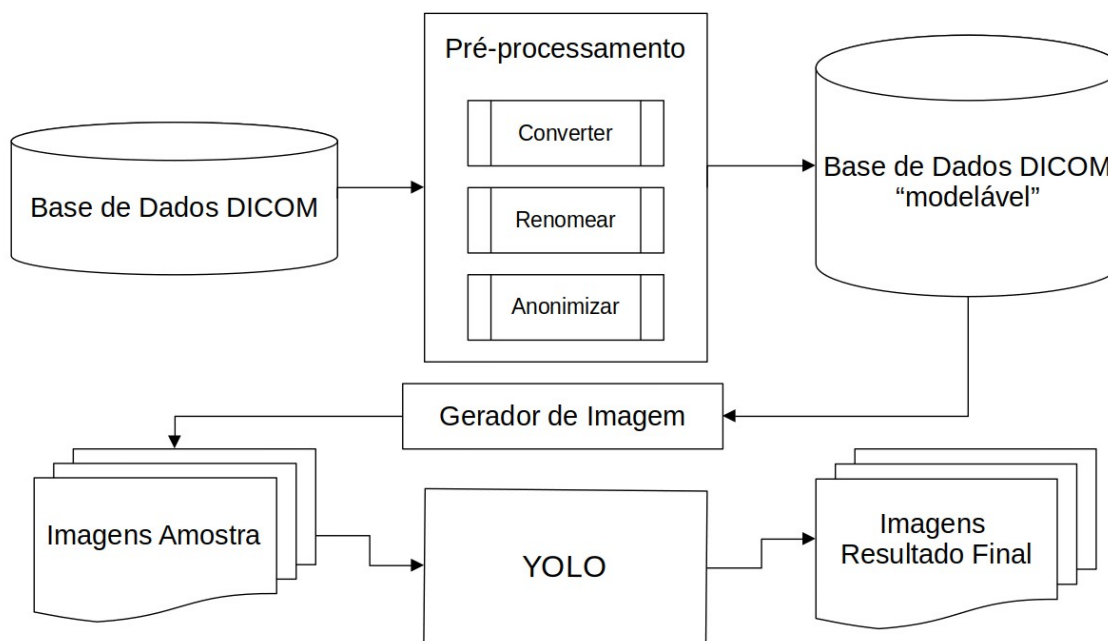


Figura 4. Fluxograma do tratamento dos dados

### 2.3.1. Converter os arquivos de imagem para JPEG.

Um arquivo DICOM contendo uma imagem JPEG torna possível a interpretação, visualização e a modelagem dos dados contidos nesse arquivo pelo RStudio. Sem essa etapa, mensagens de erro e incompatibilidade serão geradas pelo terminal do R. É necessário instalar a biblioteca dcmtool pelo Terminal, substituindo "path" pela localização dos arquivos DICOM no sistema:

```
$ sudo apt install dcmtool
```

```
$ find ~/path/* -name '*.dcm' -exec dcmdjpeg {} {} \;
```

### **2.3.2. Renomear os arquivos pelos valores armazenados em suas variáveis de interesse.**

Consiste em extrair da tabela do arquivo os valores de interesse que serão usados para substituir o nome original do arquivo. Essa etapa será realizada pelo pacote `divest::sortDICOM()`. As variáveis que darão o nome do arquivo são: "Patient ID", "SerieNumber" e "InstanceNumber". Esta etapa facilita a manipulação e catalogação do arquivo DICOM.

```
R> library(divest)
```

```
R> sortDicom(path = path, labelFormat = "%i_%s_%r", forceStack = FALSE)
```

### **2.3.3. Anonimizar**

Todas as imagens devem ser anonimizadas. Os arquivos devem ter todas as informações pessoais dos pacientes retiradas conforme protocolo de pesquisa médica. Nesse processo foi usado o pacote `dcmTk::dcm_anon()`:

```
R> library(dcmTk)
```

```
R> list.files(path) %>% as_tibble() -> anon_table
```

```
R> for (obs in anon_table) {paste(file.path(path),obs,sep="")} %>% as_tibble()-> anon_table
```

```
R> for (value in anon_table$value) {dcm_anon(value)}
```

## **2.4. Join das Tabelas e Análise Exploratória dos Dados**

A tabela `base_tratada`, contém as variáveis de interesse de todos os arquivos do banco de imagem. Essa tabela representa o join da tabela `annot_clean` com a tabela `index_dicom` gerada pelas variáveis contidas no nome dos arquivos DICOM após o pré-processamento.

```
R> list.files("path") %>% as_tibble() -> array_arquivos_dicom_tratados
```

```
R> separate(array_arquivos_dicom_tratados, col = value, sep = "_", into = c("PatientID",  
"SerieNumber", "InstanceNumber")) -> index_dicom
```

```
R> index_dicom$InstanceNumber %>% str_sub(end=-5) -> index_dicom$InstanceNumber
```

```
R> bind_cols(array_arquivos_dicom_tratados,index_dicom) -> index_dicom
```

```
R> names(index_dicom) <- c("FileName","PatientID","SerieNumber","InstanceNumber")
```

```
R> index_dicom$PatientID = as.numeric(index_dicom$PatientID)
R> index_dicom$SerieNumber = as.numeric(index_dicom$SerieNumber)
R> index_dicom$InstanceNumber = as.numeric(index_dicom$InstanceNumber)
R> index_dicom %>% group_by(PatientID) %>% left_join(annot_clean, by = c("PatientID" =
"Patient ID")) %>% ungroup() -> index_dicom
R> index_dicom$FilePath <- paste("path", index_dicom$FileName, sep="")
R> index_dicom %>% dplyr::select(1:9) -> base_tratada
R> base_tratada %>% dplyr::select(-5) -> base_tratada
```

```
> base_tratada
# A tibble: 165,794 x 9
  FileName PatientID SerieNumber InstanceNumber Corte INICIAL v...1 Corte...2 Hipod...3 Densi...4 FileP...5
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 0000121029_2_1.dcm 121029 2 1 NA NA S 0 /home/...
2 0000121029_2_10.dcm 121029 2 10 NA NA S 0 /home/...
3 0000121029_2_100.dcm 121029 2 100 NA NA S 0 /home/...
4 0000121029_2_101.dcm 121029 2 101 NA NA S 0 /home/...
5 0000121029_2_102.dcm 121029 2 102 NA NA S 0 /home/...
6 0000121029_2_103.dcm 121029 2 103 NA NA S 0 /home/...
7 0000121029_2_104.dcm 121029 2 104 NA NA S 0 /home/...
8 0000121029_2_105.dcm 121029 2 105 NA NA S 0 /home/...
9 0000121029_2_106.dcm 121029 2 106 NA NA S 0 /home/...
10 0000121029_2_107.dcm 121029 2 107 NA NA S 0 /home/...
# ... with 165,784 more rows, and abbreviated variable names 1'Corte INICIAL vesícula',
# 2'Corte FINAL vesícula', 3'Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)',
# 4'Densidade não habitual (sim=1, não=0)', 5FilePath
# Use 'print(n = ...)' to see more rows
```

Figura 5. Output Terminal RStudio objeto tibble base\_tratada

Fonte: 5QUINTO\_SCRIPT.R

Do total de 165.794 observações presentes na tabela, observa-se:

```
R> table(base_tratada$`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`)
```

```

C    H    S    V
55309 8353 27421 71244
```

```
R> base_tratada %>% ggplot(aes(x = `Hipodistendida/Vesícula normal/Clipe/sem clipe
(HVSC)`) + geom_bar(aes(fill = `Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`)
+ ylim(0, 80000) + geom_text(stat = 'count', aes(label = ..count..), vjust=-0.5) +
ggtitle("base_tratada") + ylab("contagem") + theme(legend.position = "none")
```

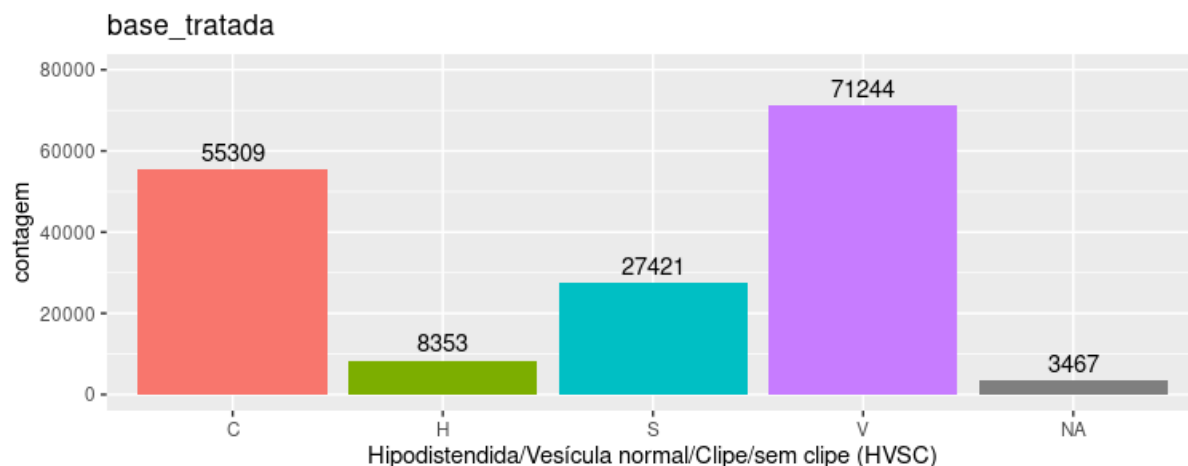


Figura 6. Tabela de frequência das categorias da tibble base\_tratada

Fonte: 5QUINTO\_SCRIPT.R

Comparativo com as frequências da tabela annot\_clean:

```
R> table(annot_clean$`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`)
```

```
C H S V
73 10 27 90
```

```
R> annot_clean %>% ggplot(aes(x = `Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`)) +
  geom_bar(aes(fill = `Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`)) +
  ylim(0, 105) + geom_text(stat = 'count', aes(label = ..count..), vjust=-0.5) +
  ggtitle("annot_clean") + ylab("contagem") + theme(legend.position = "none")
```

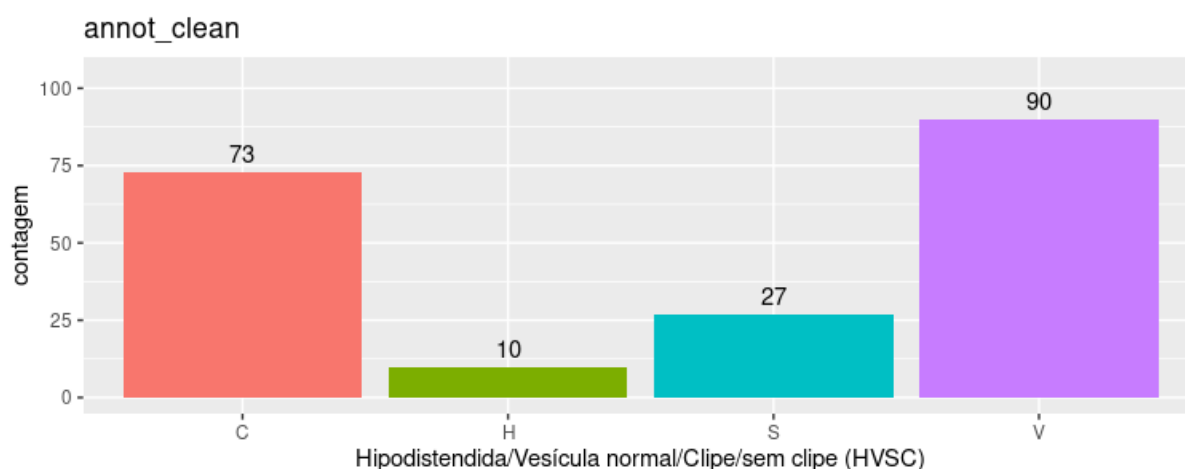


Figura 7. Tabela de frequência das categorias da tibble annot\_clean

Fonte: 5QUINTO\_SCRIPT.R

Para localizar as imagens que contém a vesícula é necessário transformar as duas variáveis `base_tratada$`Corte INICIAL vesícula`` e `base_tratada$`Corte FINAL vesícula`` em uma nova variável categórica binária (`posicao_interesse`) com valores de 1 para a presença da vesícula naquela observação ou 0 para a não visualização da vesícula naquela observação. Para isso é necessário filtrar uma tabela que contenha apenas as observações que possuem a anotação vesícula normal (V).

```
R> base_tratada %>% filter(`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)` == "V") -> base_vesicula_normal
R> base_vesicula_normal %>% mutate(posicao_interesse = ifelse(c(InstanceNumber >= `Corte INICIAL vesícula` & InstanceNumber <= `Corte FINAL vesícula`), 1, 0)) -> base_frames_vesicula_normal
```

```
> base_frames_vesicula_normal
# A tibble: 4,110 × 10
  FileName PatientID SerieNumber InstanceNumber Corte IN...1 Corte...2 Hipod...3 Densi...4 FileP...5 posic...6
  <chr>      <dbl>      <dbl>      <dbl>      <dbl> <dbl> <dbl> <fct>      <dbl> <dbl>
1 0000483240_4_40.dcm 483240      4          40          40 56 V 1 /home/... 1
2 0000483240_4_41.dcm 483240      4          41          40 56 V 1 /home/... 1
3 0000483240_4_42.dcm 483240      4          42          40 56 V 1 /home/... 1
4 0000483240_4_43.dcm 483240      4          43          40 56 V 1 /home/... 1
5 0000483240_4_44.dcm 483240      4          44          40 56 V 1 /home/... 1
6 0000483240_4_45.dcm 483240      4          45          40 56 V 1 /home/... 1
7 0000483240_4_46.dcm 483240      4          46          40 56 V 1 /home/... 1
8 0000483240_4_47.dcm 483240      4          47          40 56 V 1 /home/... 1
9 0000483240_4_48.dcm 483240      4          48          40 56 V 1 /home/... 1
10 0000483240_4_49.dcm 483240      4          49          40 56 V 1 /home/... 1
# ... with 4,100 more rows, and abbreviated variable names 1`Corte INICIAL vesícula`, 2`Corte FINAL vesícula`,
# 3`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`, 4`Densidade não habitual (sim=1, não=0)`,
# 5FilePath, 6posicao_interesse
# i Use `print(n = ...)` to see more rows
```

Figura 8. Output Terminal RStudio objeto tibble `base_tvesicula_normal`

Fonte: 5QUINTO\_SCRIPT.R

Em 4.110 observações foram detectada a presença da vesícula biliar. Foram separadas dessa amostra as imagens que localizam o meio da vesícula, para que a identificação do órgão fique o mais visível possível.

```
R> base_frames_vesicula_normal %>% unite("ID_SERIE", PatientID:SerieNumber) -> ID_SERIE
R> base_frames_vesicula_normal[!duplicated(ID_SERIE$ID_SERIE),] -> ID_UNICO
R> ID_UNICO$MEDIA <- c(ID_UNICO$`Corte INICIAL vesícula` + ID_UNICO$`Corte FINAL vesícula`) / 2
R> ID_UNICO$MEDIA = as.integer(ID_UNICO$MEDIA)
```

```
R> ID_UNICO %>% unite("PatientID","PatientID","SerieNumber","MEDIA", sep = "_") %>%
select(PatientID) -> gerador
R> ID_UNICO %>% select(PatientID,SerieNumber,MEDIA) -> PatienIDSerieNumberMEDIA
R> bind_cols(gerador, PatienIDSerieNumberMEDIA) -> geradorPatienIDSerieNumberMEDIA
R> geradorPatienIDSerieNumberMEDIA$FINAL <-
paste("0000",geradorPatienIDSerieNumberMEDIA$PatientID...1,".dcm", sep = ")
R> geradorPatienIDSerieNumberMEDIA %>% mutate(FINAL = ifelse(str_length(FINAL) >=
20, str_sub(FINAL,start=2L), FINAL)) -> geradorPatienIDSerieNumberMEDIA
R> geradorPatienIDSerieNumberMEDIA %>% left_join(base_frames-vesicula_normal,
by=c("FINAL"="FileName")) %>% select(FilePath) -> vesicula_select
```

Como resultado a tabela `vesicula_select` selecionou 240 observações. Esse número é maior do que a quantidade total de pacientes com vesícula ( $H + V = 100$ ). Isso ocorre pois para cada paciente foram coletadas mais de uma série de imagens (`SerieNumber`).

Portanto, essas 240 imagens foram entregues aos médicos da UNIFESP que avaliaram em quais imagens selecionadas foi detectada a presença da vesícula. Após esse laudo de constatação médica, 75 imagens foram selecionadas para compor a amostra da categoria vesícula.

## 2.5. Modelo GLM e o Clipe Cirúrgico

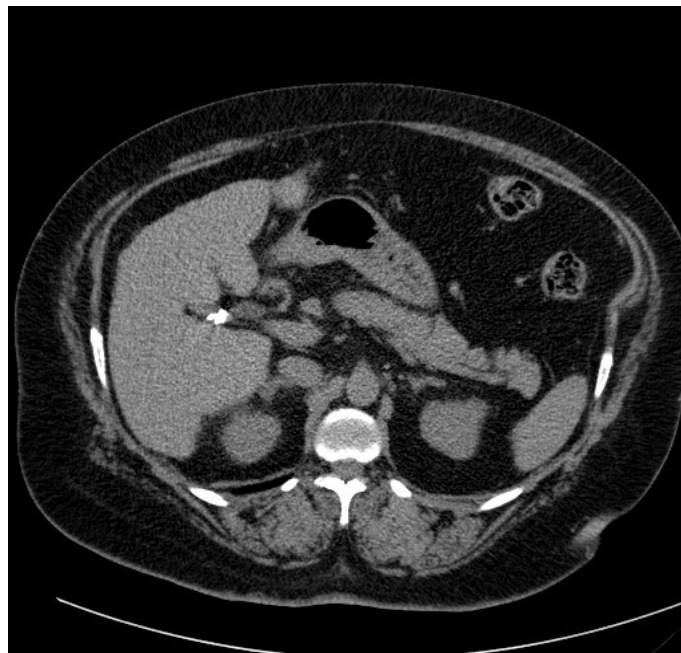


Figura 9. Exemplo Imagem Tomográfica DICOM contendo o clipe cirúrgico

Fonte: banco de Imagens DICOM

A tabela `annot_clean` foi anotada pelos médicos apontando a posição da vesícula nas imagens. Inicialmente não foram anotadas as informações referentes a posição dos cliques cirúrgicos. Portanto a criação de um modelo `glm()` será interessante para auxiliar a identificação visual das imagens com alta probabilidade de conter o clipe cirúrgico.

Os valores da variável `$posicao_interesse` possibilitam a criação do modelo `glm()` de regressão logística binária uma vez que foram observados os valores de 1 para a presença da vesícula na observação ou 0 para a não presença da vesícula na observação. O objetivo desse modelo é tentar encontrar a imagem que contém o clipe cirúrgico através da anotação da posição da vesícula.

```
R> base_tratada %>% filter(`Hipodistendida/Vesícula normal/Clique/sem clipe (HVSC)` %in%  
c('H','V')) -> tabela_GLM  
R> tabela_GLM %>% mutate(posicao_interesse = ifelse(c(InstanceNumber >= `Corte  
INICIAL vesícula` & InstanceNumber <= `Corte FINAL vesícula`), 1,0)) -> tabela_GLM  
R> glm(posicao_interesse ~ InstanceNumber + SerieNumber, family = binomial, data =  
tabela_GLM) -> modelo_glm  
R> step(modelo_glm) -> modelo_fit  
R> summary(modelo_fit)
```

Call:

```
glm(formula = posicao_interesse ~ InstanceNumber + SerieNumber,  
     family = binomial, data = tabela_GLM)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.86810	-0.36552	-0.16512	-0.03731	2.66852

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.9877638	0.0323654	-30.519	<2e-16 ***
InstanceNumber	-0.0191188	0.0003399	-56.243	<2e-16 ***
SerieNumber	0.0090062	0.0047032	1.915	0.0555 .

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 33681 on 78264 degrees of freedom

Residual deviance: 26466 on 78262 degrees of freedom

AIC: 26472

Number of Fisher Scoring iterations: 8

```
R> tabela_GLM$fit <- modelo_fit$fitted.values
```

Será gerada uma nova tabela de imagens CLIPE (C) contendo apenas as observações com valores de fit próximos aos valores encontrados nas observações anotadas com a presença da vesícula.

```
R> tabela_GLM$fit %>% summary()
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.000006557	0.001450455	0.021443340	0.055772056	0.085847624	0.313947461

```
R> base_clipe <- base_tratada %>% filter(`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)` == "C")
```

```
R> predict(object = modelo_fit, data.frame(base_clipe), type = "response") -> base_clipe$fit
```

```
R> c(as.integer(mean(base_vesicula_normal$`Corte INICIAL vesícula`))) - 10 -> ENTRADA
```

```
R> c(as.integer(mean(base_frames_vesicula_normal$`Corte FINAL vesícula`))) + 10 -> SAIDA
```

```
R> base_clipe %>% filter(fit >= 0.025 & fit <= 0.21) %>% filter(InstanceNumber >= ENTRADA & InstanceNumber <= SAIDA) -> clipe_select
```

A tabela clipe\_select contém 7.106 imagens pertencentes a 64 pacientes. Conforme protocolo de pesquisa médica, as imagens foram entregues aos médicos da UNIFESP que anotaram em quais foram detectadas a presença do clipe cirúrgico. Após o laudo de constatação médica, 31 imagens foram selecionadas para compor a amostra.

## 2.6. Amostras de treino, validação e teste.

Ao total foram selecionadas 106 imagens, sendo 75 da categoria vesícula e 31 da categoria CLIPE. Para o melhor desempenho do algoritmo, as categorias Vesícula Hipodistendida (H) e Vesícula Normal (V) serão unidas e compreendidas como sendo apenas uma categoria, pois ambas as labels tratam do mesmo objeto só que com



dimensões diferentes. E para que não haja um desbalanceamento no modelo, será reduzido o número de amostras de treino da categoria vesícula.

Tabela 1. Amostras

Amostra	Imagens Vesícula	Imagens Clipe
Treino	43	25
Validação	7	3
Teste	25	3

## 2.7. Gerador de Imagens no R – Exportando as amostras.

O gerador de imagens desenvolvido nesse projeto possibilita a preparação da amostra para servir de input para a rede neural YOLO. Foram geradas imagens com aproximadamente 50kb. O gerador de imagens também cria uma nova variável "ImagePath" na tabela de onde serão extraídas as imagens. A variável "ImagePath" indica no sistema a localização desse novo arquivo criado para cada observação.

```
R> temp_dcm_export <- function(arg1, arg2, arg3){dcmj2pnm(arg1, opt = arg2, outfile = arg3, "--use-window 1")}
```

Gerar a amostra da categoria vesícula a partir da tabela vesicula\_select:

```
R> amostra-vesicula %>% group_by(FilePath) %>% mutate(imagePath = temp_dcm_export(FilePath, "--write-jpeg", make.names(paste(str_sub(FilePath, start = 35L, end = -5L), "png", sep = "."), unique = TRUE))) %>% ungroup() -> amostra-vesicula
```

Gerar a amostra label clipe a partir da tabela clipe\_select:

```
R> clipe_select %>% group_by(FilePath) %>% mutate(imagePath = temp_dcm_export(FilePath, "write-jpeg", make.names(paste(str_sub(FilePath, start = 35L, end = -5L), "png", sep = "."), unique = TRUE))) %>% ungroup() -> clipe_select
```

As 106 imagens selecionadas foram movidas para a pasta *amostra* no sistema e catalogadas na tabela *base\_imgs\_amostra* para análise.

```
R> path3 = "/home/luiz/amostra/"
```

```
R> list.files(path = path3) %>% as_tibble() -> array_imgs_amostra
```

```
R> list.files(path = path3) %>% as_tibble() %>%
  separate(col = value, sep = "_", into = c("PatientID","SerieNumber","InstanceNumber")) ->
  imgs_amostra
R> imgs_amostra$InstanceNumber %>% str_sub(end=-5) ->
  imgs_amostra$InstanceNumber
R> imgs_amostra$PatientID %>% str_sub(start=2) -> imgs_amostra$PatientID
R> imgs_amostra$PatientID = as.numeric(imgs_amostra$PatientID)
R> imgs_amostra$SerieNumber = as.numeric(imgs_amostra$SerieNumber)
R> imgs_amostra$InstanceNumber = as.numeric(imgs_amostra$InstanceNumber)
R> base_tratada %>% group_by(PatientID) %>% right_join(imgs_amostra) ->
  base_imgs_amostra
```

```
> base_imgs_amostra
# A tibble: 106 × 9
# Groups:   PatientID [98]
  FileName PatientID SerieNumber InstanceNumber Corte INICIAL ve...1 Corte...2 Hipod...3 Densi...4 FileP...5
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <fct> <dbl> <chr>
1 0000548545_2_61.dcm 548545 2 61 NA NA C 0 /home/...
2 0000629515_2_55.dcm 629515 2 55 NA NA C 0 /home/...
3 0000629515_6_53.dcm 629515 6 53 NA NA C 0 /home/...
4 0000695874_4_43.dcm 695874 4 43 37 49 V 1 /home/...
5 0000718485_2_50.dcm 718485 2 50 41 60 V 0 /home/...
6 0000720391_2_51.dcm 720391 2 51 NA NA C 0 /home/...
7 0000823083_6_41.dcm 823083 6 41 NA NA C 0 /home/...
8 0000840771_6_44.dcm 840771 6 44 34 54 V 0 /home/...
9 0000888356_3_41.dcm 888356 3 41 NA NA C 0 /home/...
10 0001012709_2_42.dcm 1012709 2 42 36 48 V 1 /home/...
# ... with 96 more rows, and abbreviated variable names 1`Corte INICIAL vesícula`, 2`Corte FINAL vesícula`,
# 3`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`, 4`Densidade não habitual (sim=1, não=0)`,
# 5FilePath
# Use `print(n = ...)` to see more rows
```

Figura 11. Output Terminal RStudio objeto tibble base\_imgs\_amostra

Fonte: 5QUINTO\_SCRIPT.R

### 3. Implementação do Algoritmo YOLO de classificação de Imagem.

O algoritmo YOLO (You Only Look Once) foi o escolhido para esse projeto por apresentar alta acurácia e bom desempenho computacional. A Rede Neural Convolucional YOLO possui uma vasta compatibilidade entre diferentes hardwares e boa interconectividade entre plataformas. A versão v7 utilizada nesse projeto foi escrita em pytorch o que a torna mais veloz, com melhor acurácia e precisão do que as versões anteriores. Também é possível monitorar o desempenho do treinamento da rede e os resultados, deploys e dashboards do modelo. A rede neural dividi a imagem em regiões e prevê a probabilidades de existirem objetos de interesse dentro dessas regiões. Esses objetos de interesse são identificados na imagem por “bounding boxes”.

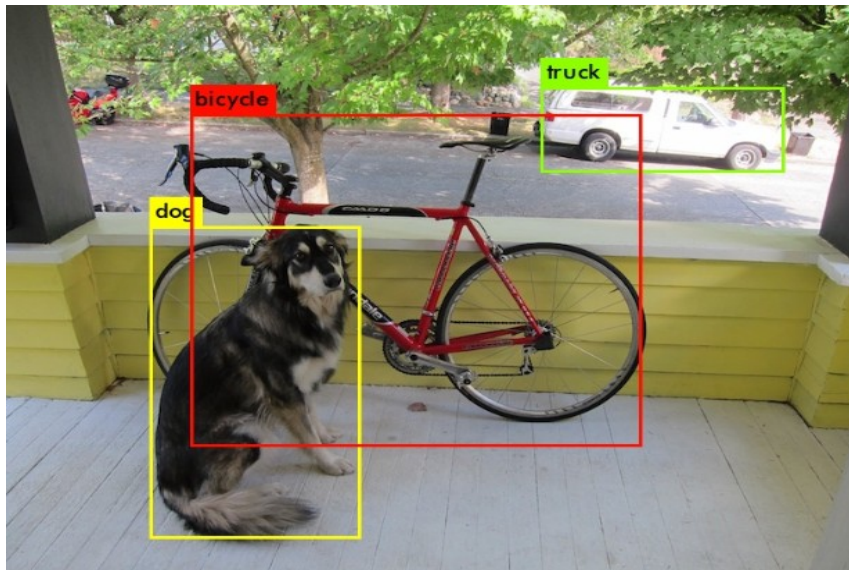


Figura 12. Imagem com Bounding Box e 3 categorias

Fonte: [https://pjreddie.com/media/files/papers/yolo\\_1.pdf](https://pjreddie.com/media/files/papers/yolo_1.pdf)

Para a implementação do Algoritmo YOLO é necessário clonar o repositório github da versão v7 e instalar todos os módulos requeridos:

```
$ git clone https://github.com/WongKinYiu/yolov7.git
```

```
$ cd yolov7
```

```
$ pip install -r requirements.txt
```

Também é necessário fazer o download do peso pré-treinado para realizar o “transfer learning” do modelo:

```
$ wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt
```

### 3.1. Criação das Classes e Bounding Boxes nas imagens.

Determinar a “Bounding Box” é determinar a área dentro de 4 coordenadas de pontos na imagem que indicam um “retângulo de recorte”. Essa etapa será realizada com o labellmg:

```
$ pip install labellmg
```

```
$ labellmg
```

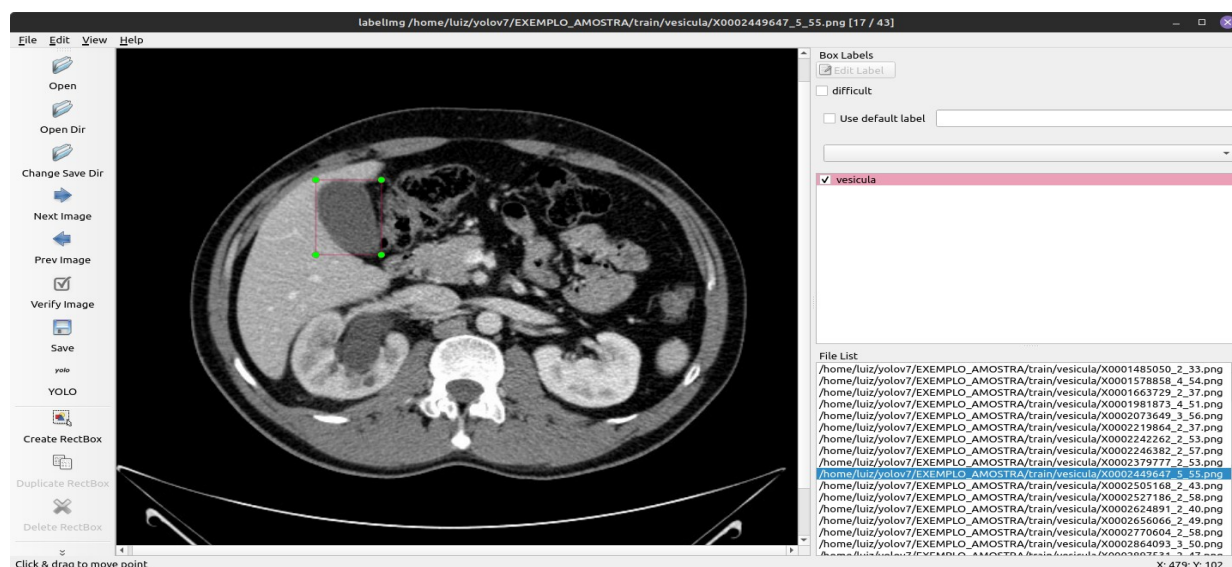


Figura 13. Exemplo do uso do labelImg para apontamento das Bounding Boxes nas Imagens

O apontamento das Bounding boxes pelo labelImg gera um arquivo .txt com 4 valores numéricos referentes as coordenadas dos pontos na imagem. Para o algoritmo YOLO treinar a rede neural é necessário que todas as amostras de treino e validação tenham suas respectivas áreas de bounding box demarcadas. Conforme o exemplo:

```
$ cat ~/yolov7/DICOM_VESICULA_YOLO/train/vesicula/X0002449647_5_55.txt
```

```
1 0.345703 0.340820 0.101562 0.138672
```

É necessário criar um arquivo '.yaml' apontando para as pastas contendo as amostras de treino, validação e teste. Especificar a quantidade de classes que a rede neural terá e o nome dado a cada classe. O arquivo '.yaml' deve ser salvo na pasta yolov7/data seguindo o exemplo:

```
$ cat data/yolo_luiz_vesicula.yaml
```

```
#database amostra
```

```
train: /home/luiz/yolov7/DICOM_VESICULA_YOLO/train/ #treino
```

```
val: /home/luiz/yolov7/DICOM_VESICULA_YOLO/val/ #validação
```

```
test: /home/luiz/yolov7/DICOM_VESICULA_YOLO/test/ #teste
```

```
nc: 2 # numero de classes
```

```
names: [ 'clipe', 'vesicula' ] # nome das classes
```

### 3.2. Treino da rede convolucional.

Os hiperparâmetros escolhidos para treinamento da rede estão detalhados no arquivo data/hyp.scratch.p5.yaml

```
$ cat /data/hyp.scratch.p5.yaml
```

```
lr0: 0.01 # initial learning rate (SGD=1E-2, Adam=1E-3)
lrf: 0.1 # final OneCycleLR learning rate (lr0 * lrf)
momentum: 0.937 # SGD momentum/Adam beta1
weight_decay: 0.0005 # optimizer weight decay 5e-4
warmup_epochs: 3.0 # warmup epochs (fractions ok)
warmup_momentum: 0.8 # warmup initial momentum
warmup_bias_lr: 0.1 # warmup initial bias lr
box: 0.05 # box loss gain
cls: 0.3 # cls loss gain
cls_pw: 1.0 # cls BCELoss positive_weight
obj: 0.7 # obj loss gain (scale with pixels)
obj_pw: 1.0 # obj BCELoss positive_weight
iou_t: 0.20 # IoU training threshold
anchor_t: 4.0 # anchor-multiple threshold
# anchors: 3 # anchors per output layer (0 to ignore)
fl_gamma: 0.0 # focal loss gamma (efficientDet default gamma=1.5)
hsv_h: 0.015 # image HSV-Hue augmentation (fraction)
hsv_s: 0.7 # image HSV-Saturation augmentation (fraction)
hsv_v: 0.4 # image HSV-Value augmentation (fraction)
degrees: 0.0 # image rotation (+/- deg)
translate: 0.2 # image translation (+/- fraction)
scale: 0.9 # image scale (+/- gain)
shear: 0.0 # image shear (+/- deg)
perspective: 0.0 # image perspective (+/- fraction), range 0-0.001
flipud: 0.0 # image flip up-down (probability)
fliplr: 0.5 # image flip left-right (probability)
mosaic: 1.0 # image mosaic (probability)
mixup: 0.15 # image mixup (probability)
copy_paste: 0.0 # image copy paste (probability)
```

```
paste_in: 0.15 # image copy paste (probability), use 0 for faster training  
loss_ota: 1 # use ComputeLossOTA, use 0 for faster training
```

O script `train.py` é o responsável por treinar a rede. As configurações escolhidas para o treinamento e os hiperparâmetros estão na estrutura da linha de comando:

```
$ python3 train.py --weights yolov7.pt --data "data/yolo_luiz_vesicula.yaml" --workers  
8 --batch-size 1 --img 512 512 --cfg cfg/training/yolov7.yaml --name yolo_tomo_v1 --hyp  
data/hyp.scratch.p5.yaml
```

Com a conclusão do treinamento da rede, o melhor peso gerado `best.pt` será usado para identificar a presença da vesícula ou do clipe nas amostras de teste que não participaram do treinamento da rede. O comando deve apontar para a pasta que contém a amostra de teste conforme o exemplo:

```
$ python detect.py --weights runs/train/yolo_tomo_v1/weights/best.pt --source  
~/yolov7/DICOM_VESICULA_YOLO/test/
```

Os resultados podem ser encontrados na pasta `yolov7/runs/detect/`

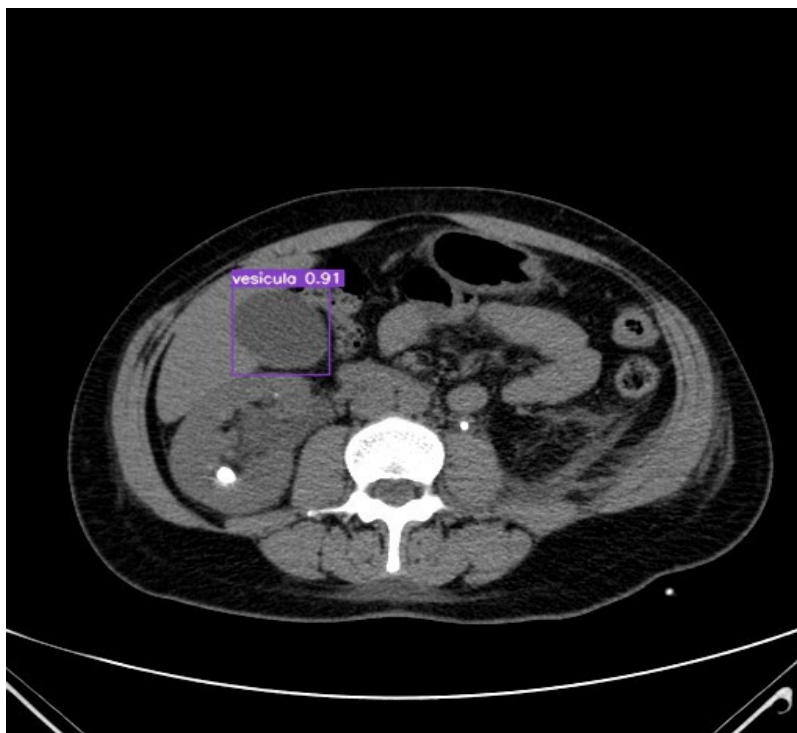


Figura 14. Resultado gerado pelo modelo `yolo_tomo_v1` na imagem `0003871186_2_64.dcm` da Tabela 2



#### 4. Resultados e Discussão

A rede treinada apresentou algumas métricas como output em forma de sumário:

```
Optimizer stripped from runs/train/yolo_tomo_v1/weights/last.pt, 74.8MB
Optimizer stripped from runs/train/yolo_tomo_v1/weights/best.pt, 74.8MB
Images sizes do not match. This will causes images to be display incorrectly in the UI.
wandb: Waiting for W&B process to finish... (success).
wandb:
wandb: Run history:
wandb:   metrics/mAP_0.5
wandb:   metrics/mAP_0.5:0.95
wandb:   metrics/precision
wandb:   metrics/recall
wandb:   train/box_loss
wandb:   train/cls_loss
wandb:   train/obj_loss
wandb:   val/box_loss
wandb:   val/cls_loss
wandb:   val/obj_loss
wandb:   x/lr0
wandb:   x/lr1
wandb:   x/lr2
wandb: Run summary:
wandb:   metrics/mAP_0.5 0.83687
wandb:   metrics/mAP_0.5:0.95 0.53136
wandb:   metrics/precision 0.86302
wandb:   metrics/recall 0.83333
wandb:   train/box_loss 0.01043
wandb:   train/cls_loss 0.001
wandb:   train/obj_loss 0.00173
wandb:   val/box_loss 0.06675
wandb:   val/cls_loss 0.00823
wandb:   val/obj_loss 0.0066
wandb:   x/lr0 0.001
wandb:   x/lr1 0.001
wandb:   x/lr2 0.001
wandb: Synced yolo_tomo_v1: https://wandb.ai/luizcast/YOLOR/runs/kwh2ji1o
wandb: Synced 6 W&B file(s), 172 media file(s), 1 artifact file(s) and 0 other file(s)
wandb: Find logs at: ./wandb/run-20220928\_140827-kwh2ji1o/logs
(base) Luiz@box:~/yolov7$
```

Figura 15. output treinamento da rede neural convolucional no Terminal

Fonte: <https://wandb.ai/luizcast/YOLOR/runs/kwh2ji1o>

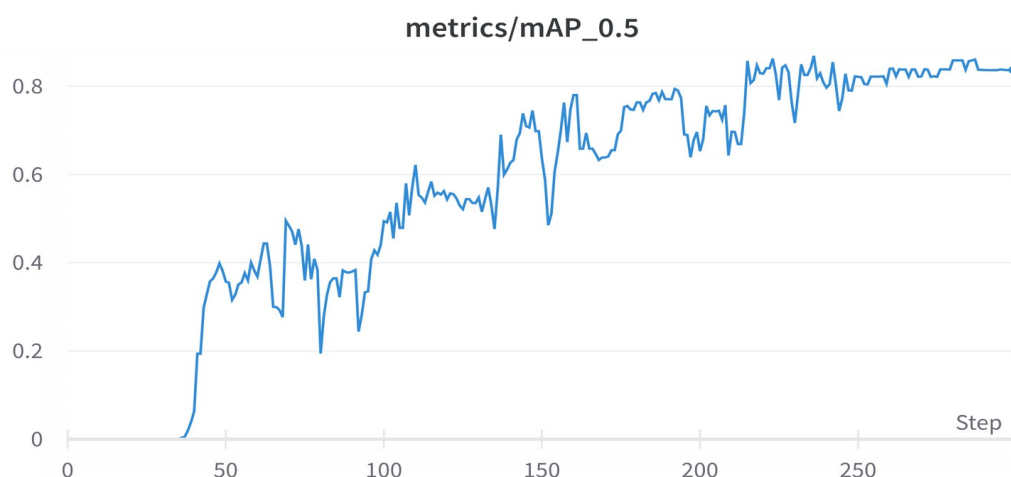


Figura 16. gráfico mAP\_0.5 do modelo yolo\_tomo\_v1 em 300 steps

Fonte: <https://wandb.ai/luizcast/YOLOR/runs/kwh2ji1o>

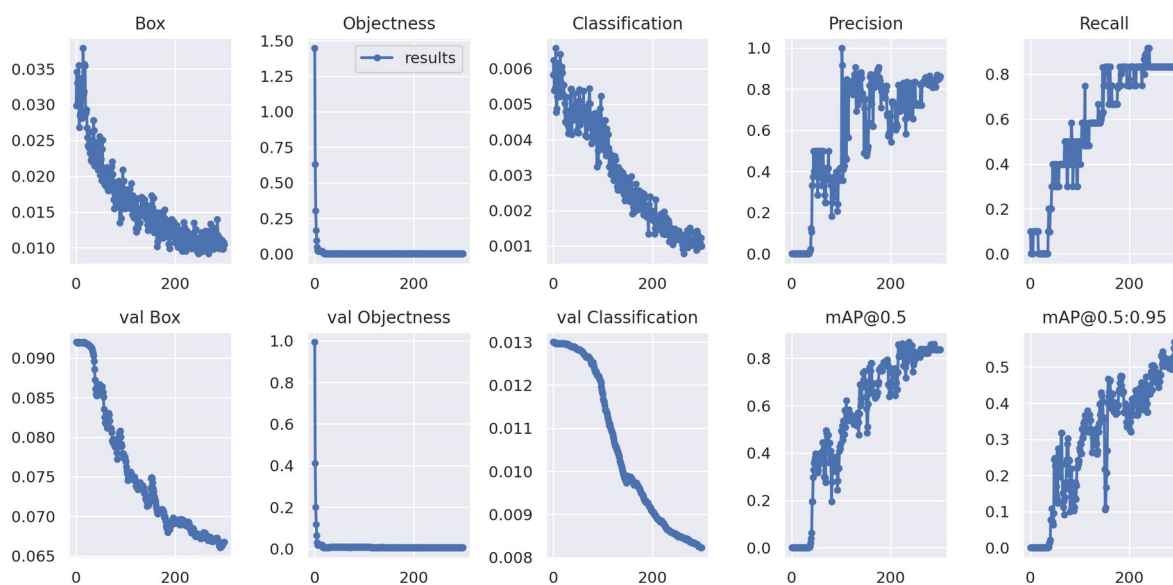


Figura 17. Métricas de treinamento da rede neural convolucional yolo\_tomo\_v1

Fonte: <https://wandb.ai/luizcast/YOLOR/runs/kwh2ji1o>

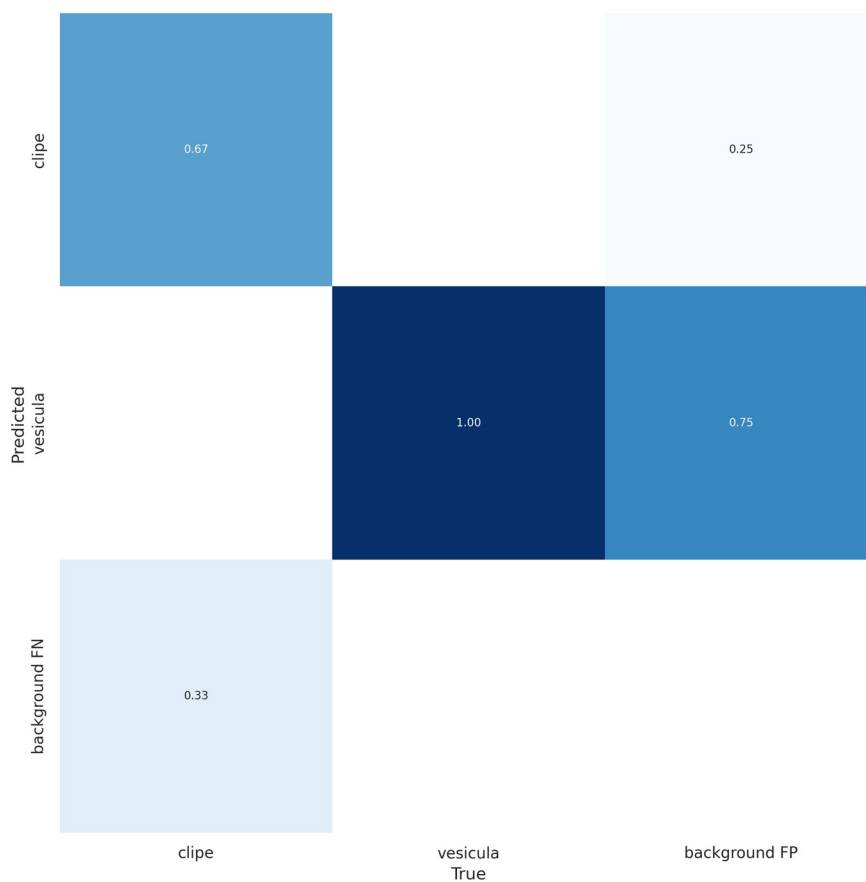


Figura 18. Matriz de Confusão do modelo yolo\_tomo\_v1

Fonte: <https://wandb.ai/luizcast/YOLOR/runs/kwh2ji1o>



Das 28 imagens que foram usadas como amostra de teste, 27 foram classificadas corretamente pelo algoritmo.

Tabela 2. Resultados Amostra Teste

FileName	Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)	Objeto Identificado Corretamente	Acurácia
0003655566_2_63.dcm	V	TRUE	0.95
0003660351_3_42.dcm	V	TRUE	0.94
0003662481_2_46.dcm	V	TRUE	0.89
0003664496_2_52.dcm	V	TRUE	0.91
0003735879_2_48.dcm	C	TRUE	0.94
0003770320_2_42.dcm	C	TRUE	0.68
0003770320_4_46.dcm	C	TRUE	0.79
0003775180_7_59.dcm	V	TRUE	0.60
0003782194_2_68.dcm	V	TRUE	0.97
0003823903_2_60.dcm	V	TRUE	0.88
0003827068_6_83.dcm	H	FALSE	NA
0003845225_2_47.dcm	V	TRUE	0.93
0003852776_2_69.dcm	V	TRUE	0.78
0003856451_2_45.dcm	V	TRUE	0.89
0003870806_2_62.dcm	V	TRUE	0.86
0003871186_2_64.dcm	V	TRUE	0.91
0003873742_2_48.dcm	V	TRUE	0.96
0003877229_2_69.dcm	V	TRUE	0.93
0003879500_2_46.dcm	V	TRUE	0.94
0003886379_2_49.dcm	V	TRUE	0.89
0003886702_2_43.dcm	V	TRUE	0.96
0003888331_2_68.dcm	V	TRUE	0.69
0003889073_2_129.dcm	V	TRUE	0.69
0003889831_2_65.dcm	V	TRUE	0.88
0003890200_2_112.dcm	V	TRUE	0.98
0003897044_2_41.dcm	V	TRUE	0.90
0003897387_2_49.dcm	V	TRUE	0.71
0003897980_2_31.dcm	V	TRUE	0.94

Tabela 3. Sumário da Acurácia do Modelo

Valor Mínimo	Primeiro Quartil	Mediana	Média	Terceiro Quartil	Valor Máximo
0.6	0.825	0.9	0.8663	0.94	0.98

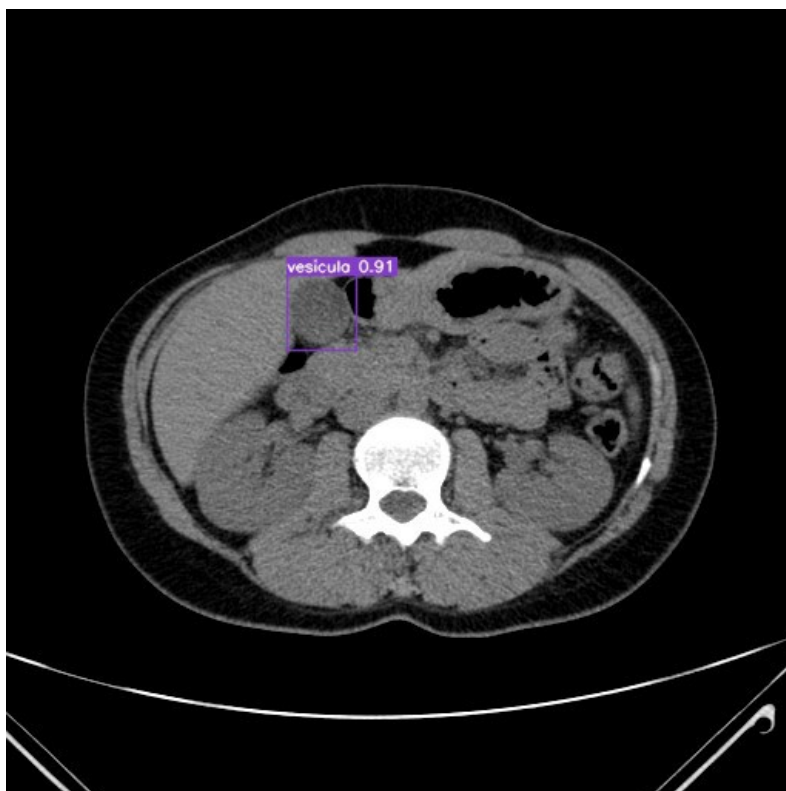


Figura 19. Resultado gerado pelo modelo yolo\_tomo\_v1 na imagem 0003664496\_2\_52.dcm da Tabela 2

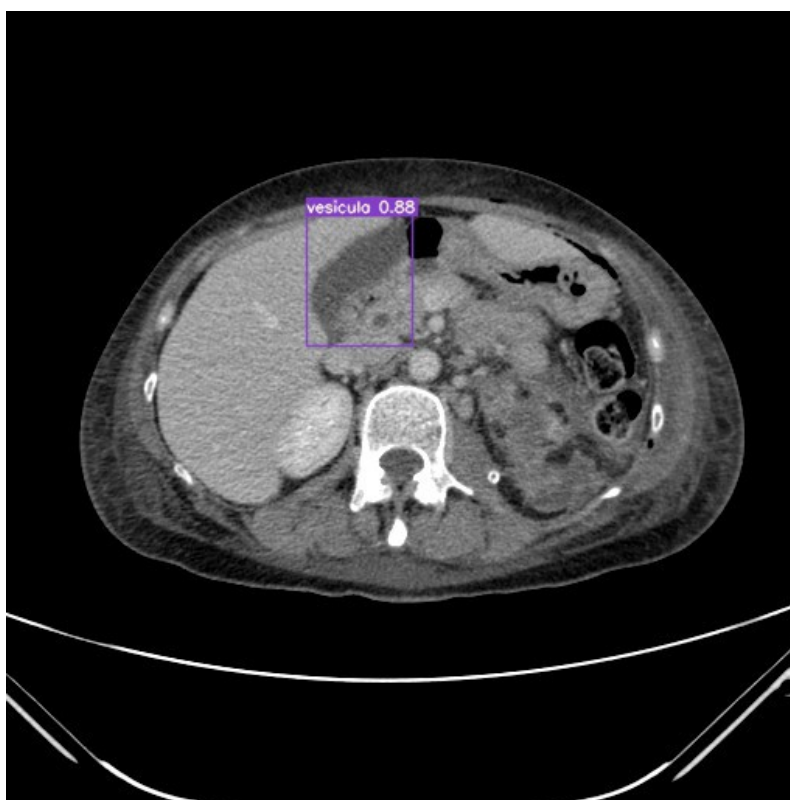


Figura 20. Resultado gerado pelo modelo yolo\_tomo\_v1 na imagem 0003823903\_2\_60.dcm da Tabela 2

Como resultado o modelo cria uma cópia de cada imagem com o objeto de classificação contornado com a bounding box. Acima da região identificada pela bounding box é criada uma legenda contendo o nome da classe referente a identificação e a acurácia encontrada pelo modelo. O modelo yolo\_tomo\_v1 apresentou uma acurácia interessante como resultado.

As ferramentas de Inteligência Artificial têm se mostrado como sendo as principais ferramentas de transformação da sociedade moderna. Modelos que auxiliam o diagnóstico e tratamento correto de pacientes podem gerar um benefício enorme a sociedade. Esta rede neural convolucional por exemplo, tem o potencial de ajudar o médico a tomar decisões melhores. Além disso, embasá-las em um modelo matemático eficaz que reforce seu diagnóstico. Num futuro, essa rede tem o potencial de emitir um certificado de qualidade ou de inferir diferentes possíveis diagnósticos ou quem sabe sugerir tratamentos, medicamentos. Talvez prever o tempo de vida de um paciente em sofrimento, ou até identificar doenças de difícil visibilidade. Essa rede pode ser usada para ajudar a formar os futuros médicos, treinar peritos e catalogar doenças. Todas essas inferências antes eram impossíveis em grande volumes de dados apenas com o olho humano e com softwares simples de observação de exames. Agora, é importante discutirmos os benefícios da I.A. para auxiliar o desenvolvimento e avanço social.

Outro ponto importante a ser levantado é sobre o futuro do arquivo DICOM. O arquivo DICOM é a segurança e a garantia da não-variabilidade entre diferentes amostras de imagens geradas por diferentes aparelhos tomográficos. Atualmente, a medicina aceita dois formatos como material de laudo: O DICOM impresso e o DICOM JPEG2000. O JPEG2000 está sendo discutido como um bom representante do DICOM para laudos não-presenciais via telemedicina devido as características de alta qualidade, taxa de compressão, interoperacionalidade e compatibilidade. Nesse sentido, também é necessário estabelecer um protocolo médico rígido em termos de definir qual será o formato aceito pela medicina como “input” para treinamento de uma rede neural.

Curiosamente, esse modelo yolo\_tomo\_v1.pt treinado pelo projeto possui características de transfer learning, sendo passível de melhorias e aprimoramentos. Outras classificações podem ser implementadas e as classificações existentes podem ser reforçadas. Como conclusão, esse projeto disponibiliza o repositório <[https://github.com/luizcast/DCM\\_TCC](https://github.com/luizcast/DCM_TCC)> junto com todos os arquivos e tabelas usados para a realização desse projeto e todas as 28 imagens geradas como resultado. As métricas estão detalhadas no site <<https://wandb.ai/luizcast/YOLOR/runs/kwh2ji1o>>. O arquivo yolo\_DICOM\_v1.pt é apresentado por esse projeto como modelo final contendo o melhor peso gerado pelo treinamento da rede neural e que pode servir para predir resultados e

identificar a presença da vesícula ou do clipe em outras imagens DICOM. Este modelo pré-treinado yolo\_DICOM\_v1.pt também pode servir como ponto de partida para novas redes neurais convolucionais em outros bancos de imagem DICOM.

## **5. Agradecimentos**

Dedico esse projeto ao meu pai, o Médico e Professor Dr. Mario Luiz Castiglioni, por dividir comigo a alegria e o interesse em criar esse projeto e minha família por me apoiarem nessa e em tantas outras jornadas. Gostaria de agradecer a todos os médicos do departamento de Inteligência Artificial da Escola Paulista de Medicina principalmente ao Médico e Diretor Dr. Nitamar Abdala pela tutela e os conselhos, Dr. Felipe Campos Kitamura, Dr. Igor Rafael Martins dos Santos e todos os envolvidos na coleta e disponibilização dos dados que compuseram esse projeto. Ao meu orientador Professor Rodrigo Zanin por caminhar comigo nessa etapa e a todos os professores do MBA por compartilhar comigo o conhecimento sobre essa coisa tão difícil e bonita que é a Ciência de Dados. Obrigado pela ajuda de todos, pela paciência, pela disponibilidade e acolhimento.

## **6. Referências**

Zefan Liu, Guannan Zhu, Xian Jiang, Yunuo Zhao, Hao Zeng, Jing Jing, Xuele Ma, 2020, Survival Prediction in Gallbladder Cancer Using CT Based Machine Learning, Frontiers In Oncology 10, Artigo 604288.

Young-Sik Jeong, Houcine Hassan, Arun Kumar Sangaiah, 2019, Machine learning on big data for future computing, The Journal of Supercomputing 75:2925–2929

Younbeom Jeong, Jung Hoon Kim, Hee-DongChae, Sae-Jin Park, Jae Seok Bae, Ijin Joo, Joon Koo Han, 2020, Deep learning-based decision support system for the diagnosis of neoplastic gallbladder polyps on ultrasonography: Preliminary Results, Scientific Reports 10:7700.

Kiana Ehsani, Hessam Bagherinezhad, Joseph Redmon, Roozbeh Mottaghi, Ali Farhadi, 2018, Who Let The Dogs Out? Modeling Dog Behavior From Visual Data. Conference on Computer Vision and Pattern Recognition, 2018, Salt Lake City, Utah, Estados Unidos, <<https://arxiv.org/abs/1803.10827>>. Acesso em: 18 out. 2022.

Joseph Redmon, Ali Farhadi, 2018, YOLOv3: An Incremental Improvement, University of Washington, <<https://arxiv.org/abs/1804.02767>>. Acesso em: 18 out. 2022.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, Ali Farhadi, 2018, IQA: Visual Answering in Interactive Environments. Conference on Computer Vision and Pattern Recognition, 2018, Salt Lake City, Utah, Estados Unidos, <<https://arxiv.org/abs/1712.03316>>. Acesso em: 18 out. 2022.

Joseph Redmon, Ali Farhadi, 2017, YOLO9000: Better, Faster, Stronger. Conference on Computer Vision and Pattern Recognition, 2017, Honolulu, Hawaii, Estados Unidos, <<https://arxiv.org/abs/1612.08242>>. Acesso em: 18 out. 2022.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi, 2016, XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. European Conference on Computer Vision, 2016, Amsterdam, Netherlands. <<https://arxiv.org/abs/1603.05279>>. Acesso em: 18 out. 2022.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, 2016, You Only Look Once: Unified, Real-Time Object Detection. Conference on Computer Vision and Pattern Recognition, 2016, Las Vegas, Nevada, Estados Unidos. <<https://arxiv.org/abs/1506.02640>>. Acesso em: 18 out. 2022.