**O uso de Redes Neurais Convolucionais em Imagens DICOM de Tomografia para Identificar a Vesícula Biliar**

Luiz Fernando Pereira Castiglioni, Rodrigo Zanin

**O uso de Redes Neurais Convolucionais em Imagens DICOM de Tomografia para Identificar a Vesícula Biliar**

**Resumo**

O treinamento de Redes Neurais Convolucionais para detectar a presença ou a ausência da vesícula biliar em exames de imagens no formato DICOM.

**Palavras-chave:** CNN, Detecção de Objetos, Visão Computacional, DICOM, YOLO.

**1. Introdução**

Cada vez mais o auxilio das máquinas nos ajudam a tomar decisões baseadas em dados. Na medicina diagnóstica um grande avanço ocasionado pelo uso dos dados tem sido a criação de novas ferramentas que auxiliam 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. O aprendizado de máquina nos ajuda a resolver problemas complexos antes sem solução, ou, com uma solução de difícil aplicabilidade. É o caso do objeto de estudo desse projeto.

Apesar da evolução da Tomografia como método de imagem para diagnóstico e identificação de lesões em vários órgãos e doenças, alguns pacientes que passaram pela cirurgia de colecistectomia, ou seja, a retirada da vesícula biliar, recebem diagnóstico da presença desse órgão no corpo. O contrário também pode ocorrer, onde o paciente que não foi submetido a colecistectomia, recebe o laudo anotando a ausência desse órgão. Isso ocorre por diversos fatores que podem estar relacionados desde à aquisição das imagens no aparelho tomografia (qualidade do aparelho utilizado, protocolo de aquisição utilizado), ao processamento dessas imagens (protocolos de reconstrução), a "expertise" do médico que lauda o exame e a complexidade da identificação desse órgão. 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 resultados de exames aliados às técnicas de Aprendizado de Máquina e Inteligência Artificial, tem se tornado as soluções preferíveis em diferentes campos da medicina como Inteligência Artificial e Análises de larga escala (Young‐Sik Jeong et al., 2019).

A vesícula biliar é um órgão em forma de saco, parecida com uma pêra, 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. Muitas pessoas têm a vesícula biliar removida e tem uma vida normal.

A tomografia é um exame de imagem que apresenta o corpo humano em fatias, através de raio-X das partes internas do corpo. Essas imagens fatiadas do corpo variam entre 1 milímetro de espessura das fatias. Como resultado, a tomografia apresenta fotos em escalas de cinza que contém os órgãos, tecidos e ossos, para cada parte específica facilitando a identificação visual individual de cada elemento que compõe a anatomia humana. A tomografia é um exame que comumente é feito como um recorte horizontal do corpo, mas pode ser feito em 360 graus, sendo que em alguns casos o exame tomográfico em corte horizontal é apresentado junto com o corte em plano transversal. O exame é realizado em uma máquina chamada tomógrafo onde o paciente se deita em uma maca e o aparelho captura essas diversas imagens. Como uma máquina fotográfica em Raio X. Uma das grandes vantagens é que o exame de tomografia pode armazenar várias imagens no mesmo exame exibindo-as lado a lado ou permitindo que cada imagem seja analisada individualmente. Com os tomógrafos mais modernos é possível realizar imagens em 3D, porém para pesquisa deste projeto serão utilizadas apenas imagens em 2D.

Os laudos de detecção da vesícula na tomografia estão sujeitos a algumas falhas específicas devido às particularidades desse órgão. 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 de 7 a 10 centímetros, tornando possível identificá-la com maior facilidade. Porém, em casos de emergência, quando o paciente sofre um trauma e chega na UTI do hospital 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 e 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 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 contém 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 biblioteca “dcmtk” e “labelImg”, e o repositório github do pacote “yolov7” desenvolvido em python com o pacote “pytorch”.

**2.1. Análise Inicial dos arquivos DICOM**

Figura 1. Tomografia Dicom da região do abdômen

O aquivo 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 daquela 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`

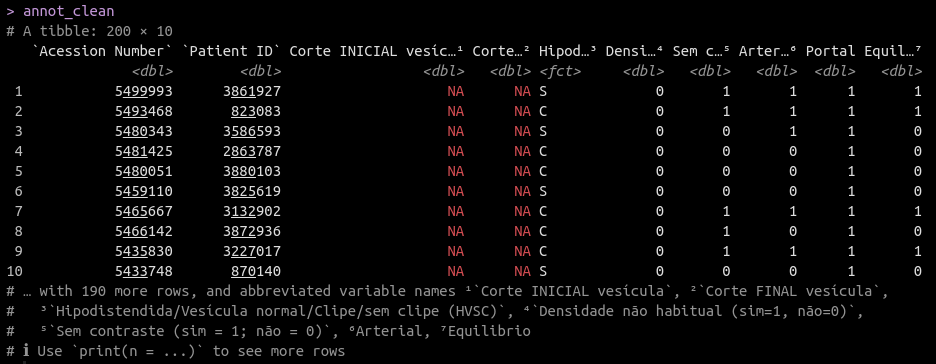
R> gsub(pattern = "Não se aplica", replacement = "NA", annot\_clean$`Corte FINAL vesícula`) -> annot\_clean$`Corte FINAL vesícula`

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)`)

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 imagens com 3 distintas tonalidades de contraste.

“Arterial” – Esta variável categórica binária indica verdadeiro ou falso se o arquivo refere-se 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 refere-se 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 refere-se 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 formam desenvolvidos ao longo dos anos para acomodar melhor as análises e processamentos. (Brandon Whitcher, Volker J. Schmid, Andrew Thornton, 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 processo foi possível modelar os dados contidos no arquivo DICOM. A partir desse ponto, a estrutura de um arquivo DICOM passa a ser interpretada pelo R como uma “lista” contendo dois objetos.

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 processamento 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.

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

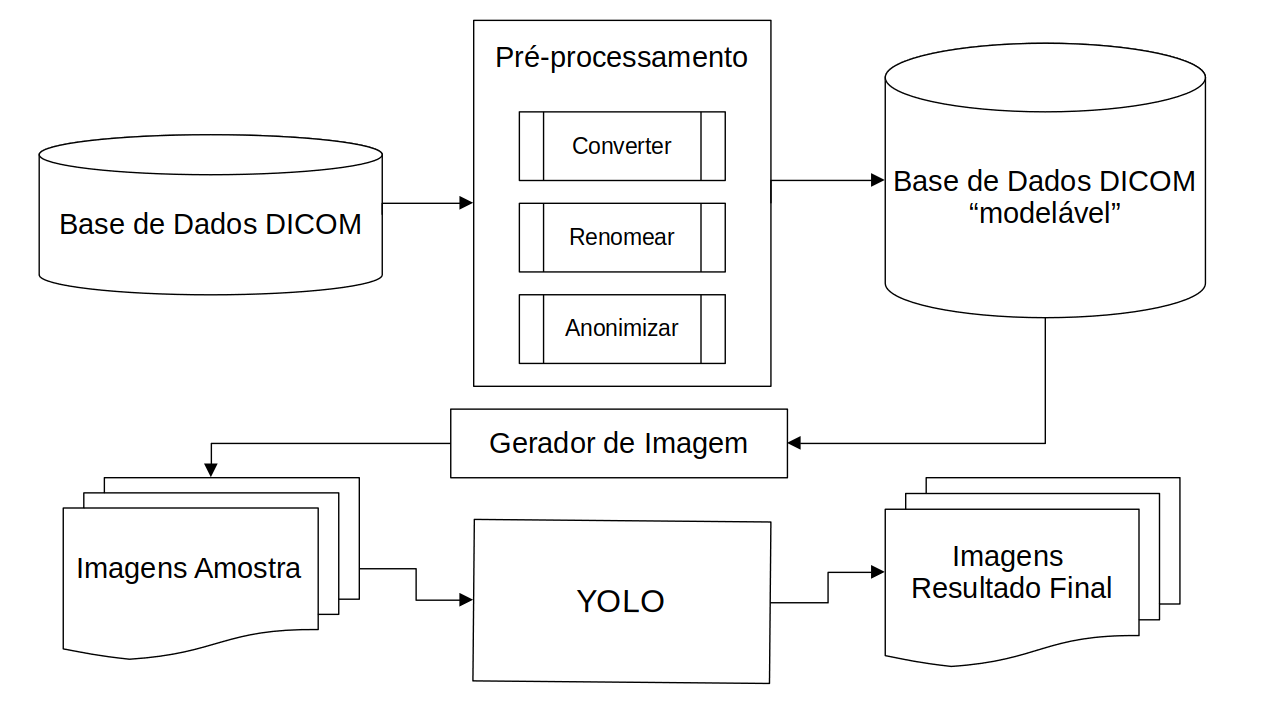
R> paste(round(c(41\*165794/630)), "segundos") # tempo total estimado em segundos de processamento da base com 165794 arquivos DICOM

[1] "10790 segundos"

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

[1] "3 horas"

Para resolução desses problemas iniciais, foi estabelecida uma etapa de Pré-processamento dos arquivos. O Pré-processamento foi dividido em 3 etapas (subprocessos) que conforme apresentados no floxograma são: Converter, Renomear e Anonimizar. A fase de Pré-processamento 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 dcmtk pelo Terminal, substituindo “path” pela localização dos arquivos no sistema:

$ sudo apt install dcmtk

$ 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. Esse processo será realizado 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() no seguinte script:

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)

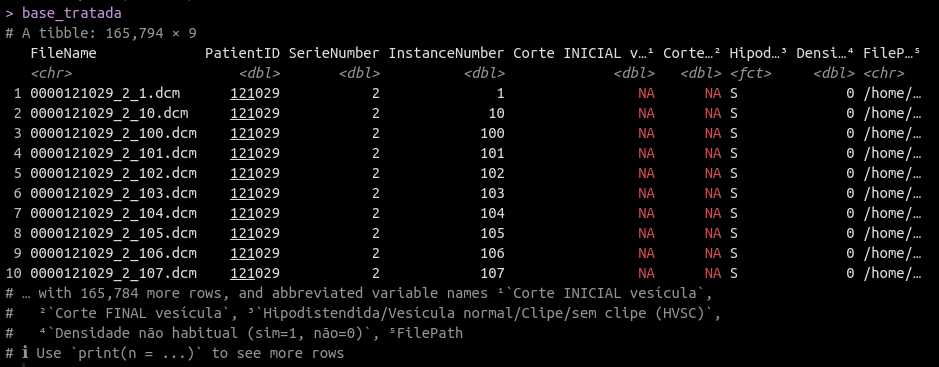
R> index\_dicom$InstanceNumber <- as.numeric(index\_dicom$InstanceNumber)

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

Figura 5. Output Terminal RStudio objeto tibble base\_tratada

Fonte: 5QUINTO\_SCRIPT.R

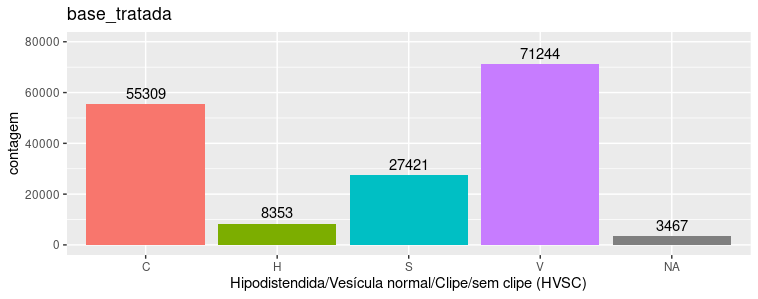
Do total de 165.794 observações presentes na tabela, observa-se:

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

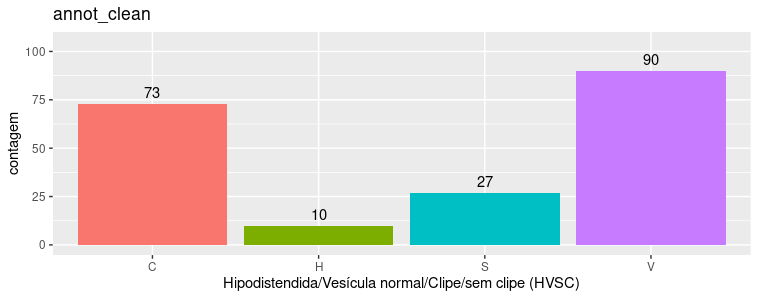
Comparativo com as frequências da tabela annot\_clean:

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")

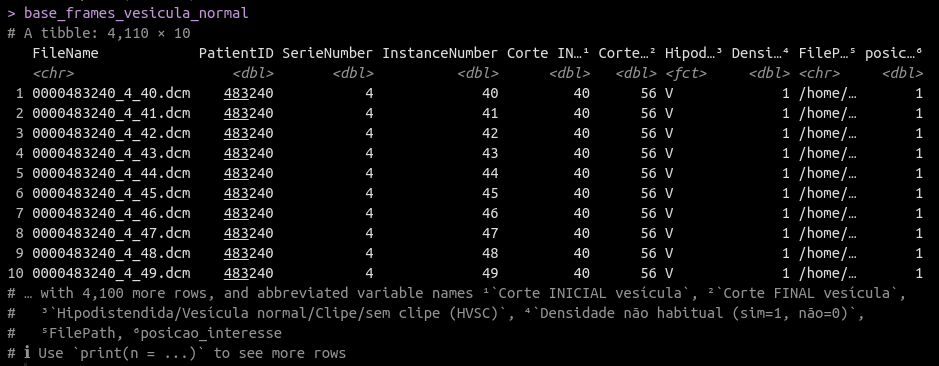
Figura 7. Tabela de frequência das categorias da tibble annot\_clean

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

Figura 8. Output Terminal RStudio objeto tibble base\_tratada

Fonte: 5QUINTO\_SCRIPT.R

Em 4.110 arquivos 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

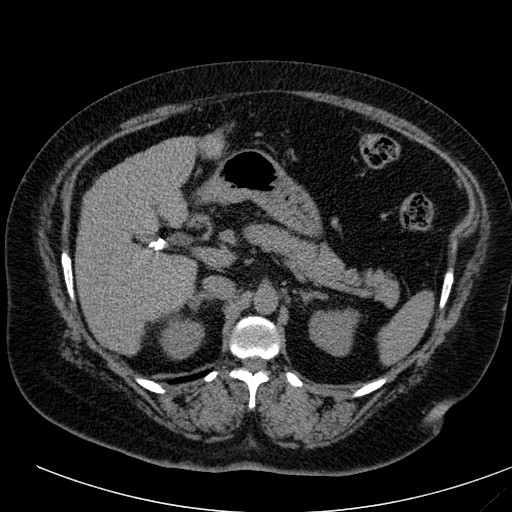
R> geradorPatienIDSerieNumberMEDIA$FINAL <- paste("0000",geradorPatienIDSerieNumberMEDIA$PatientID...1,".dcm", sep = '')

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 apontou para 240 imagens. 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 detectadas 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. Inicialente não foram anotadas as informações referentes a posição dos clipes 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/Clipe/sem clipe (HVSC)` %in% c('H','V')) %>% filter(str\_length(FileName) <= 21)-> 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. Ao total de duas categorias de classificação para o modelo: vesícula e clipe. Para o melhor desempenho do algoritmo, as categorias 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.Tabela 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 reduzidas em 20x a qualidade original. Essa medida foi determinada com intuito de pesquisa desse projeto que tem como uma das principais premissas tornar a análise e armazenamento DICOM simples, leve, rápido e intuitivo. O gerador de imagens também cria uma nova variável ImagePath na tabela de onde serão extraídas as imagens. A variável ImagePath indica a posição no sistema 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 label 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

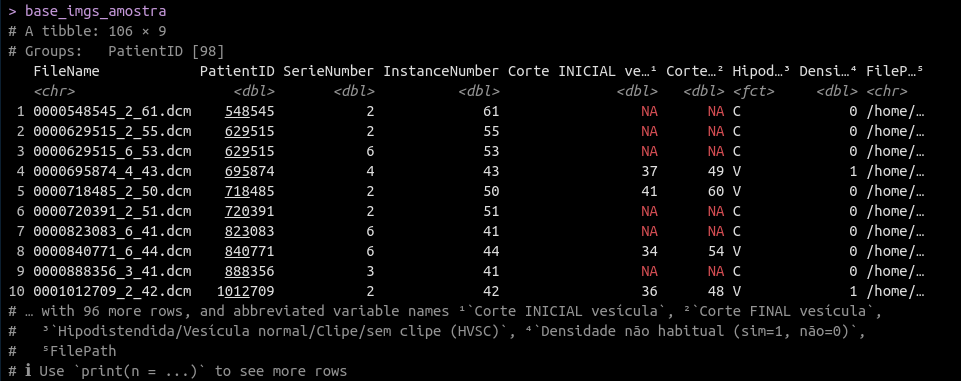
R> imgs\_amostra$PatientID %>% str\_sub(start=2) -> imgs\_amostra$PatientID

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)

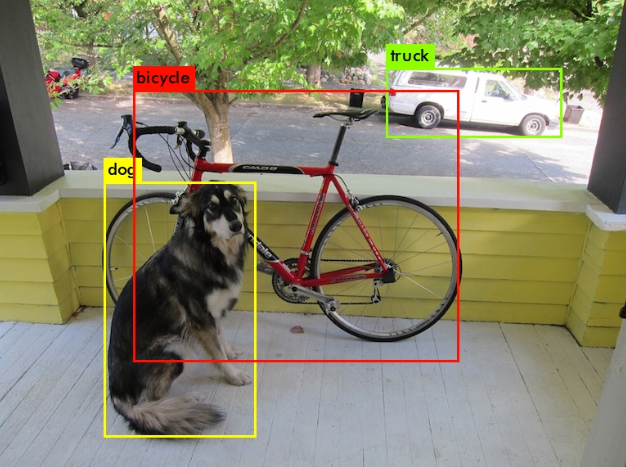
R> base\_tratada %>% group\_by(PatientID) %>% right\_join(imgs\_amostra) -> base\_imgs\_amostra

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

Fonte: 5QUINTO\_SCRIPT.R

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

O algoritimo YOLO (You Only Look Once) foi o escolhido para esse projeto por apresentar alta acurácia e bom desempenho computacional. A Rede Neural YOLO possui uma vasta compatibilidade com 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

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 baixar o peso pré-treinado para realização do transfer learning do modelo:

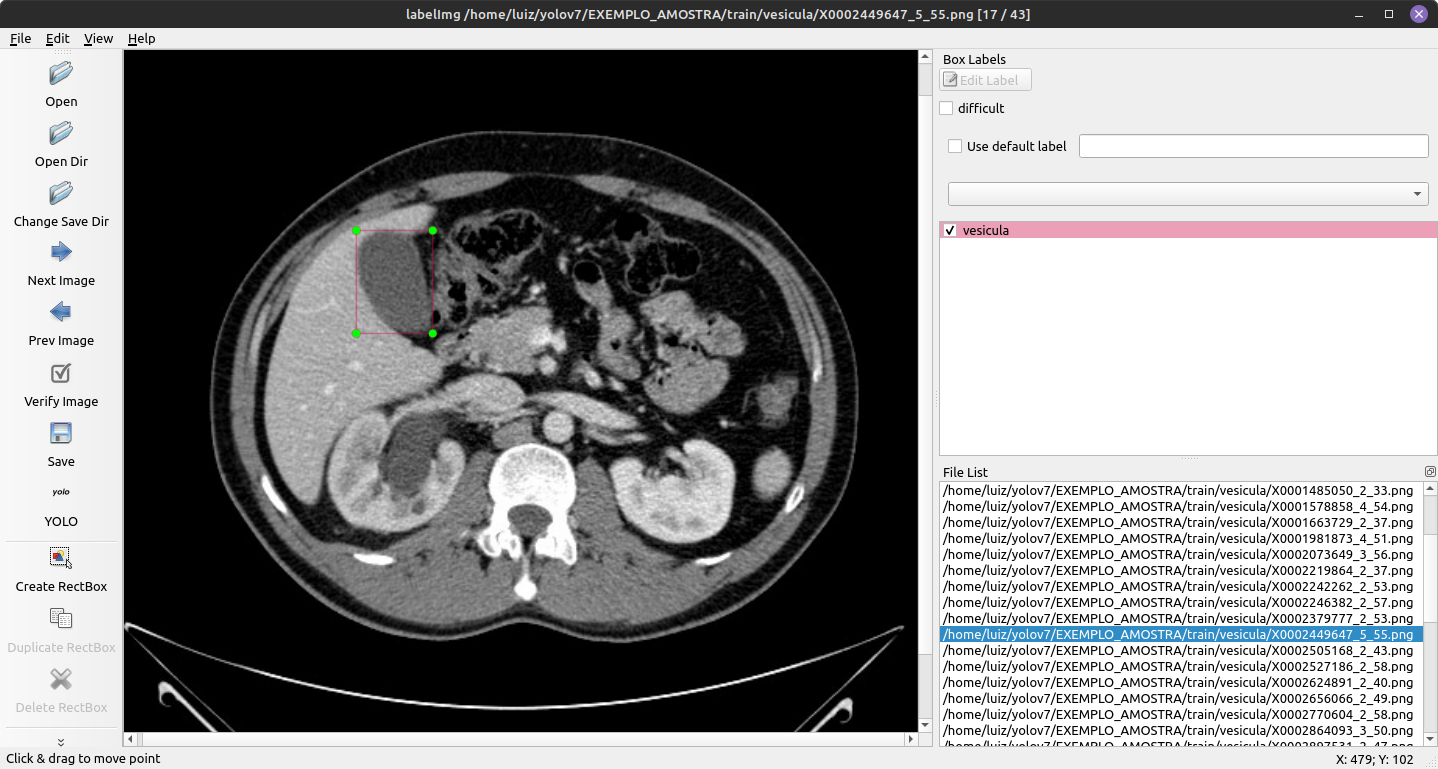
$wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt

**3.1. Criação das Labels e bounding boxes nas imagens.**

Para ensinar o modelo que será construido a interpretar as labels vesícula e clipe, é necessário 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 a biblioteca labelImg:

$ pip install labelImg

$ labelImg

Figura 13. Software 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 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 hyperparametros 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 hyperparametros 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

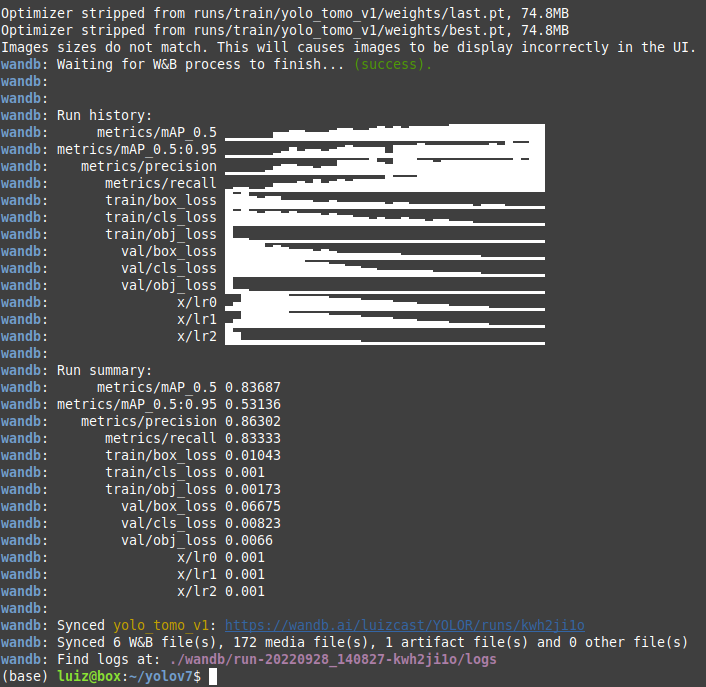
Com a conclusão do treinamento da rede, o melhor peso gerado best.pt será usado para identificar a presença da vesícula ou do clipe nas amostras de teste que não participaram do treinamento da rede. O comando deve apontar para a pasta que contém a amostra de teste conforme o exemplo:

$ python detect.py --weights runs/train/yolo\_tomo\_v1/weights/best.pt --source ~/yolov7/DICOM\_VESICULA\_YOLO/test/

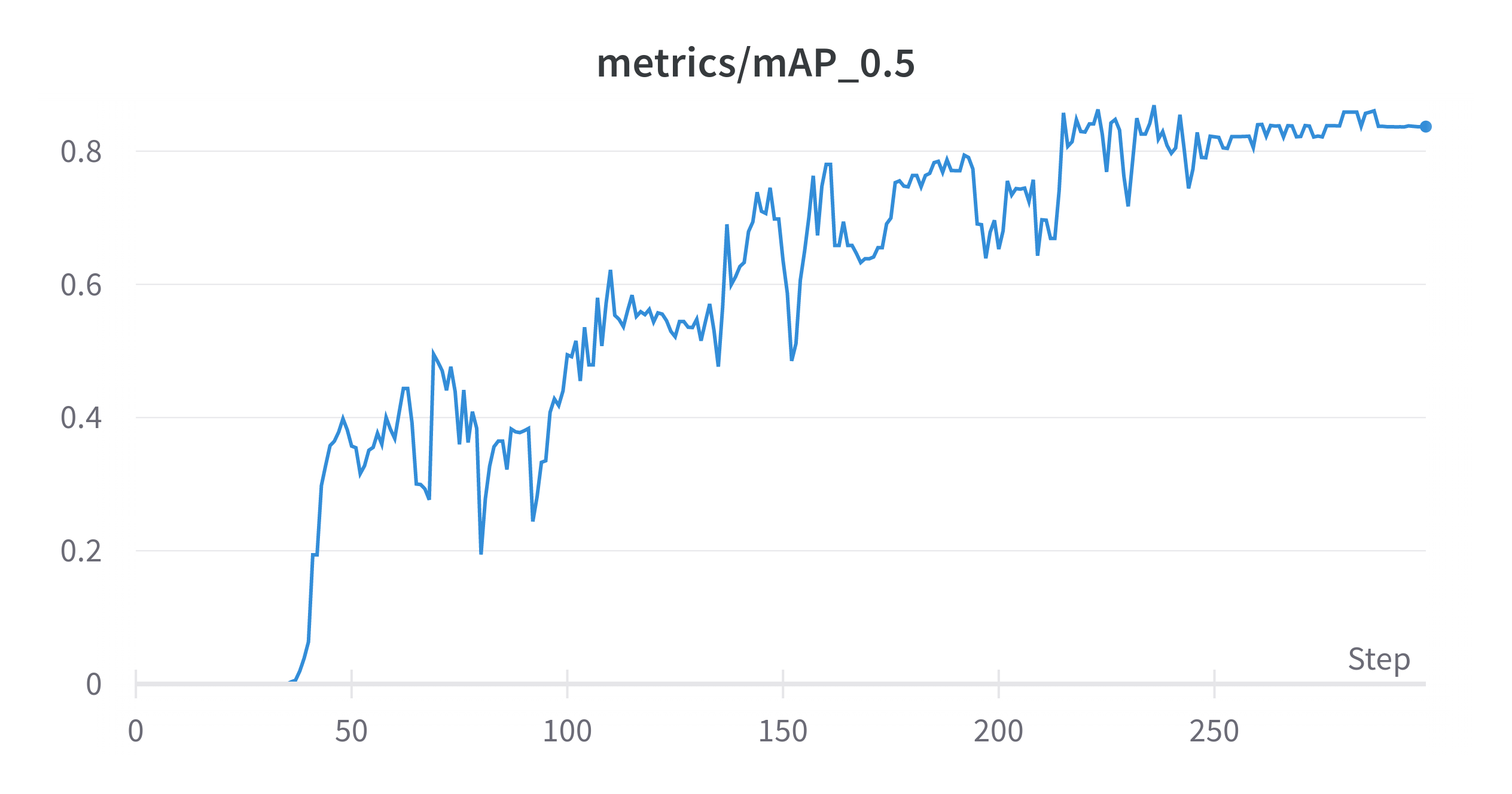
Os resultados podem ser encontrados na pasta yolov7/runs/detect/

**4. Resultados e Discussão**

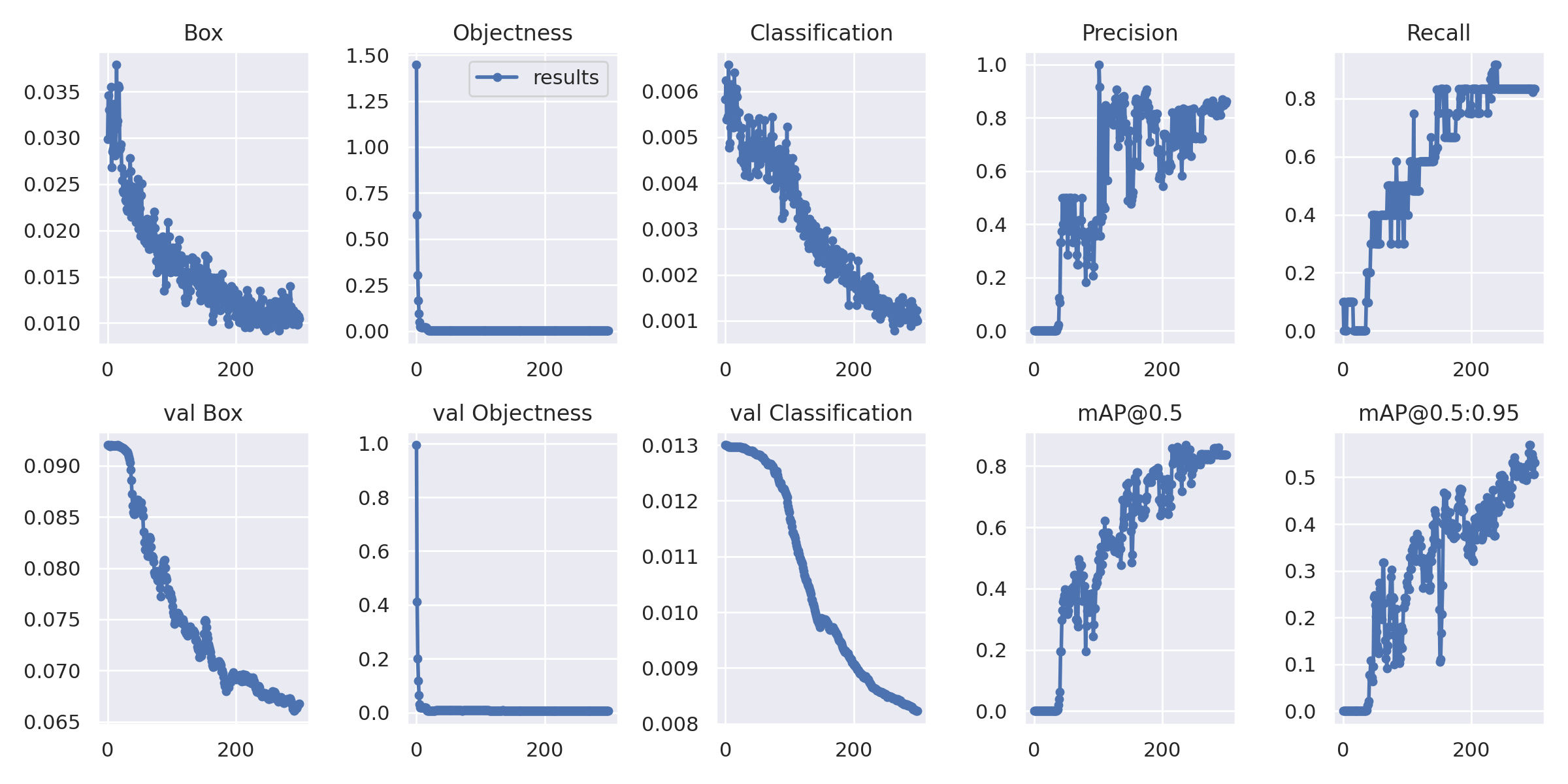
A rede treinada apresentou algumas métricas como output em forma de sumário:

Figura 15. output treinamento da rede neural convulacional no Terminal

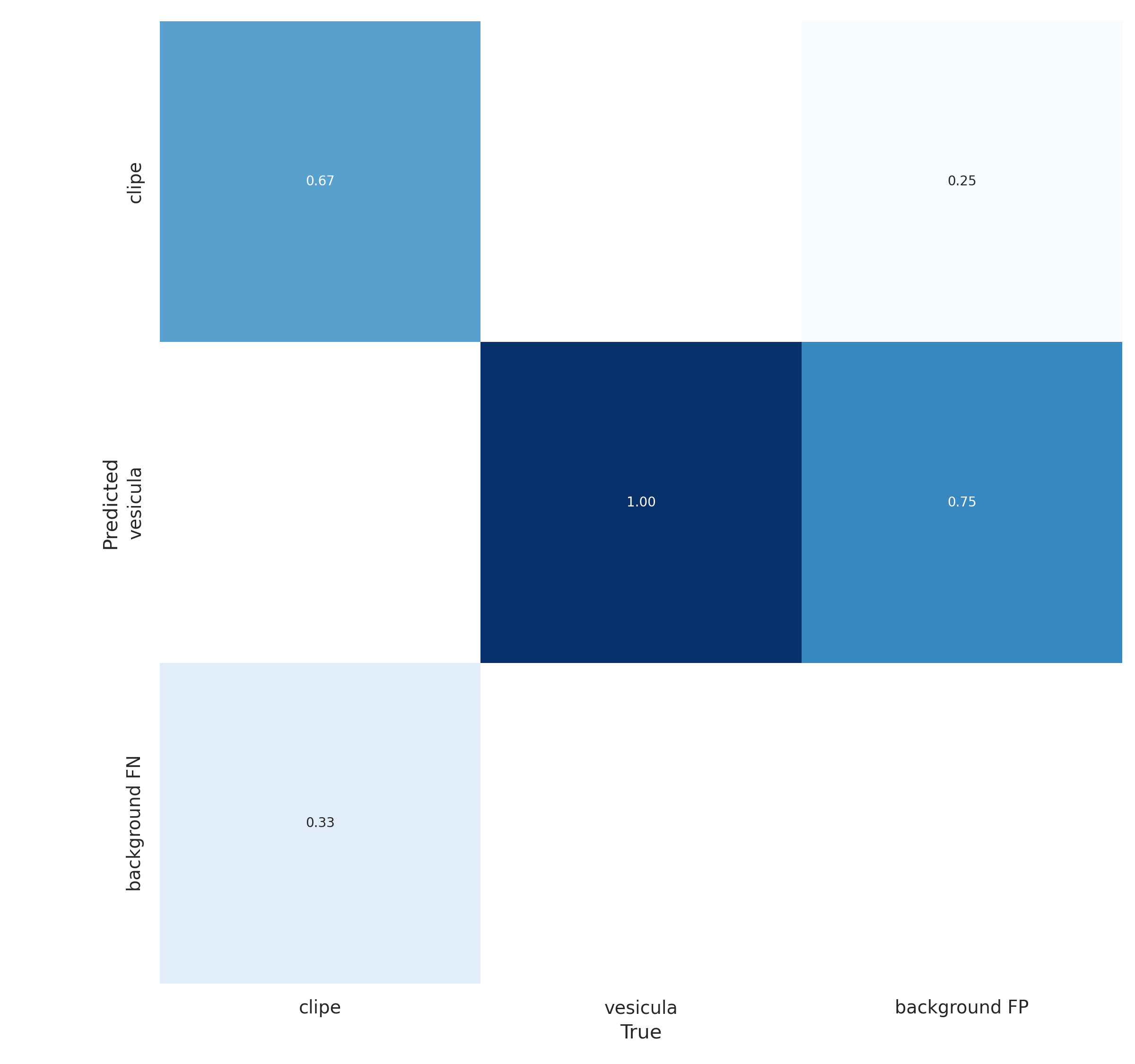
Fonte: https://wandb.ai/luizcast/YOLOR/runs/kwh2ji1o

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

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

Figura 17. output treinamento da rede neural convulacional yolo\_tomo\_v1 - Terminal

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

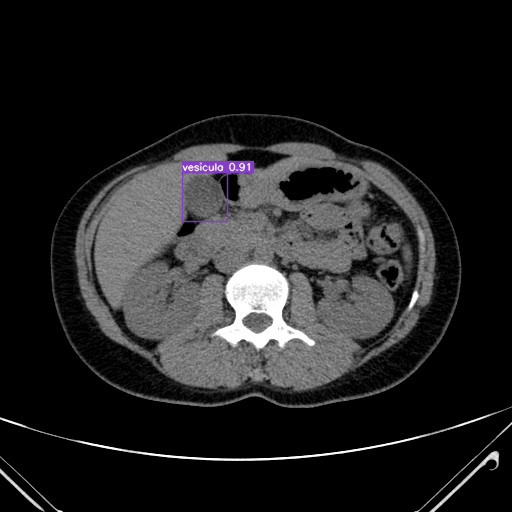
Das 28 imagens que foram usadas como amostra de teste, 27 foram classificadas corretamente pelo algoritmo.

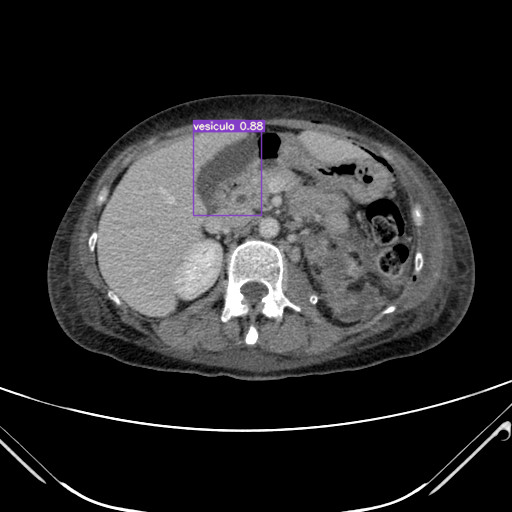
Tabela 2. Resultados Amostra Teste

|  |  |  |  |
| --- | --- | --- | --- |
| 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 apresentou uma acurácia interessante como resultado.

As ferramentas de Inteligência Artificial tem se mostrado como sendo as principais ferramentas de transformação da sociedade moderna. Modelos que auxiliam o diagnóstico e tratamento correto de pacientes podem gerar benefícios enorme a sociedade. Esta rede neural convolucional por exemplo, tem o potencial de ajudar o médico a tomar decisões melhores e 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 visualização. Essa rede pode ser usada para ajudar a formar os futuros médicos, treinar peritos, e catalogar doenças. Todas essas inferências antes eram impossíveis 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 tele-medicina 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 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 reposítório <<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 preditar 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 bancos de imagem DICOM.

**5. Agradecimentos**

Dedico esse projeto ao Médico e Professor Dr. Mario Castiglioni, meu pai, 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, ao Dr. Felipe Kitamura, ao Dr Igor 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 compartilharem comigo o conhecimento dessa coisa tão difícil e bonita que é a Ciência de Dados. Obrigado pela ajuda de todos, pela paciência, pela disponibilidade e principalmente pela diversão proporcionada.

**6. Referências**

Zefan Liu, Guannan Zhu, Xian Jiang, Yunuo Zhao, Hao Zeng, Jing Jing, Xuelei 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 Ansewering in Interactive Enviroments. 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.