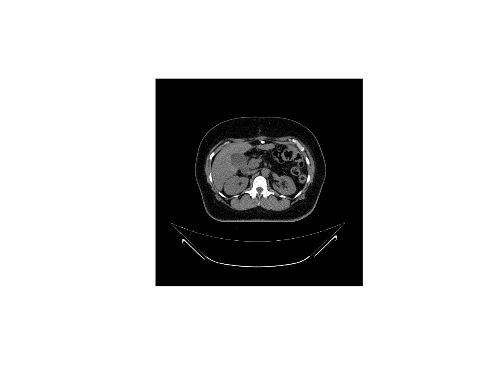
Dicom Detect with YOLOv7

Luiz Castiglioni

30/09/2022



#### Data info

* total files: 165371
* total size: 89.9GB
* format: Tomographic images and Metadata of Medical Exams in DICOM Standards Format

. Dicom Medicine Standards

<https://www.dicomstandard.org/ai>

. YOLO Algoritimo

<https://pjreddie.com/darknet/yolo/>

. Github

<https://github.com/luizcast/DCM_TCC>

list.files(include.dirs = TRUE)

## [1] "1PRIMEIRO\_SCRIPT.R" "2SEGUNDO\_SCRIPT.R"   
## [3] "3TERCEIRO\_SCRIPT.R" "4QUARTO\_SCRIPT.R"   
## [5] "5QUINTO\_SCRIPT.R" "Anot\_IMGS.R"   
## [7] "config\_nvidia.png" "dcm\_tools.Rproj"   
## [9] "IMGs" "keras.R"   
## [11] "markdown\_dicom\_yolo\_files" "markdown\_dicom\_yolo.docx"   
## [13] "markdown\_dicom\_yolo.html" "markdown\_dicom\_yolo.pdf"   
## [15] "markdown\_dicom\_yolo.Rmd" "rascunho.R"   
## [17] "README.md" "TABLEs"   
## [19] "yolo\_dicom\_v1.pt"

## Pacotes utilizados no projeto

* oro.dicom - <https://cran.r-project.org/web/packages/oro.dicom/index.html>
* dcmtk - <https://dicom.offis.de/dcmtk.php.en>
* tidyverse - <https://www.tidyverse.org/>
* tensorflow - <https://www.tensorflow.org/>
* keras - <https://keras.io/>

pacotes <- c("keras", "tensorflow", "tidyverse", "oro.dicom","dcmtk", "readxl")  
if(sum(as.numeric(!pacotes %in% installed.packages())) != 0){  
 instalador <- pacotes[!pacotes %in% installed.packages()]  
 for(i in 1:length(instalador)) {  
 install.packages(instalador, dependencies = T)  
 break()}  
 sapply(pacotes, require, character = T)   
} else {  
 sapply(pacotes, require, character = T)   
}

## Loading required package: keras

## Loading required package: tensorflow

## Loading required package: tidyverse

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4  
## ✔ tibble 3.1.8 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.1  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## Loading required package: oro.dicom  
##   
## oro.dicom 0.5.3  
##   
## Loading required package: dcmtk  
##   
## Loading required package: readxl

## keras tensorflow tidyverse oro.dicom dcmtk readxl   
## TRUE TRUE TRUE TRUE TRUE TRUE

tf\_gpu\_configured()

## Loaded Tensorflow version 2.9.2

## TensorFlow built with CUDA: TRUE   
## GPU device name: /device:GPU:0

## [1] TRUE

tf\_config()

## TensorFlow v2.9.2 (~/.local/share/r-miniconda/envs/r-reticulate/lib/python3.8/site-packages/tensorflow)  
## Python v3.8 (~/.local/share/r-miniconda/envs/r-reticulate/bin/python)

### Configurações da Máquina

neofetch --stdout

## luiz@box   
## --------   
## OS: Linux Mint 21 x86\_64   
## Host: Dell G15 5515 1.7.1   
## Kernel: 5.15.0-48-generic   
## Uptime: 4 hours, 44 mins   
## Packages: 3375 (dpkg), 9 (flatpak), 11 (snap)   
## Shell: bash 5.1.16   
## Resolution: 1920x1080   
## DE: Cinnamon   
## WM: Mutter   
## WM Theme: Mint-Y   
## Theme: Mint-Y-Dark-Purple [GTK2/3]   
## Icons: Mint-Y-Dark-Blue [GTK2/3]   
## Terminal: R   
## CPU: AMD Ryzen 7 5800H with Radeon Graphics (16) @ 4.463GHz   
## GPU: NVIDIA GeForce RTX 3060 Mobile / Max-Q   
## GPU: AMD ATI 06:00.0 Cezanne   
## Memory: 5755MiB / 15342MiB

nvidia-smi

## Fri Sep 30 20:49:59 2022   
## +-----------------------------------------------------------------------------+  
## | NVIDIA-SMI 515.65.01 Driver Version: 515.65.01 CUDA Version: 11.7 |  
## |-------------------------------+----------------------+----------------------+  
## | GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC |  
## | Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |  
## | | | MIG M. |  
## |===============================+======================+======================|  
## | 0 NVIDIA GeForce ... Off | 00000000:01:00.0 Off | N/A |  
## | N/A 53C P0 27W / N/A | 857MiB / 6144MiB | 16% Default |  
## | | | N/A |  
## +-------------------------------+----------------------+----------------------+  
##   
## +-----------------------------------------------------------------------------+  
## | Processes: |  
## | GPU GI CI PID Type Process name GPU Memory |  
## | ID ID Usage |  
## |=============================================================================|  
## | 0 N/A N/A 1292 G /usr/lib/xorg/Xorg 167MiB |  
## | 0 N/A N/A 2548 G cinnamon 37MiB |  
## | 0 N/A N/A 3267 G /usr/lib/rstudio/bin/rstudio 73MiB |  
## | 0 N/A N/A 3320 C .../lib/rstudio/bin/rsession 273MiB |  
## | 0 N/A N/A 27357 G ...090296402576475813,131072 27MiB |  
## | 0 N/A N/A 52673 C /usr/lib/R/bin/exec/R 273MiB |  
## +-----------------------------------------------------------------------------+

### Reconhecendo Padrões em imagens de Tomografia de Abdómen no formato DICOM.

Este projeto foi desenvolvido como trabalho de conclusão de curso do aluno Luiz Castiglioni no curso do MBA - Data Science and ANalytics da USP / ESALQ. Trabalho desenvolvido em parceria com o departamento de AI da Escola Paulista de Medicina UNIFESP.

Foram selecionados 200 exames de tomografia de abdomen de 200 pacientes diferentes. Desses 200 exames foram geradas anotações pelo grupo de médicos e residentes da UNIFESP sobre a presença do órgão vesícula ou a presença do clipe cirúrgico em casos de pacientes que tiveram a vesícula retirada. Surgiram 4 classes:

* Pacientes com Vesícula Normal
* Pacientes com Vesícula Hipodistendida
* Pacientes sem Vesícula e com Clipe Cirúrgico
* Pacientes sem Vesícula e sem Clipe Cirúrgico

Destas anotações surgiu a tabela **annot\_clean**

read\_excel("~/Desktop/USP/TCC/projeto\_TCC/TABLEs/annot\_clean.xlsx") %>% as\_tibble() -> annot\_clean  
annot\_clean

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

### Preparando a Base Annot\_Clean

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)`  
gsub(pattern = "Não se aplica", replacement = "NA", annot\_clean$`Corte INICIAL vesícula`) ->   
 annot\_clean$`Corte INICIAL vesícula`  
gsub(pattern = "Não se aplica", replacement = "NA", annot\_clean$`Corte FINAL vesícula`) ->   
 annot\_clean$`Corte FINAL vesícula`  
annot\_clean$`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)` <- as.factor(annot\_clean$`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`)  
annot\_clean$`Corte FINAL vesícula` <- as.numeric(annot\_clean$`Corte FINAL vesícula`)  
annot\_clean$`Corte INICIAL vesícula` <- as.numeric(annot\_clean$`Corte INICIAL vesícula`)  
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 × 10  
## Acession Num…¹ Patie…² Corte…³ Corte…⁴ Hipod…⁵ Densi…⁶ Sem c…⁷ Arter…⁸ Portal  
## <dbl> <dbl> <dbl> <dbl> <fct> <dbl> <dbl> <dbl> <dbl>  
## 1 5499993 3861927 NA NA S NA 1 1 1  
## 2 5493468 823083 NA NA C NA 1 1 1  
## 3 5480343 3586593 NA NA S NA 0 1 1  
## 4 5481425 2863787 NA NA C NA 0 0 1  
## 5 5480051 3880103 NA NA C NA 0 0 1  
## 6 5459110 3825619 NA NA S NA 0 0 1  
## 7 5465667 3132902 NA NA C NA 1 1 1  
## 8 5466142 3872936 NA NA C NA 1 0 1  
## 9 5435830 3227017 NA NA C NA 1 1 1  
## 10 5433748 870140 NA NA S NA 0 0 1  
## # … with 190 more rows, 1 more variable: Equilibrio <dbl>, and abbreviated  
## # variable names ¹​`Acession Number`, ²​`Patient ID`,  
## # ³​`Corte INICIAL vesícula`, ⁴​`Corte FINAL vesícula`,  
## # ⁵​`Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)`,  
## # ⁶​`Densidade não habitual (sim=1, não=0)`,  
## # ⁷​`Sem contraste (sim = 1; não = 0)`, ⁸​Arterial  
## # ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names

colnames(annot\_clean)

## [1] "Acession Number"   
## [2] "Patient ID"   
## [3] "Corte INICIAL vesícula"   
## [4] "Corte FINAL vesícula"   
## [5] "Hipodistendida/Vesícula normal/Clipe/sem clipe (HVSC)"  
## [6] "Densidade não habitual (sim=1, não=0)"   
## [7] "Sem contraste (sim = 1; não = 0)"   
## [8] "Arterial"   
## [9] "Portal"   
## [10] "Equilibrio"

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.