Reconhecimento de objetos em Imagens: Uma Abordagem de Aprendizado Profundo

Jamil Anderson Mansur Willy Thiago Gutsche

I. Introdução e Motivação

O reconhecimento de objetos em imagens é uma tarefa essencial em várias aplicações de visão computacional e inteligência artificial. Com o avanço tecnológico e o crescente volume de dados visuais disponíveis, tornou-se necessário desenvolver métodos eficazes para automatizar o processo de identificação e classificação de objetos em imagens [1]. Nesse contexto, a aplicação de técnicas de aprendizado profundo tem se mostrado promissora, proporcionando resultados significativos e impulsionando o desenvolvimento de sistemas de reconhecimento de objetos mais precisos e robustos.

Esta pesquisa tem como objetivo explorar o campo do reconhecimento de objetos em imagens por meio do uso de técnicas de inteligência artificial baseadas em aprendizado profundo. O aprendizado profundo é uma área da inteligência artificial que se baseia em redes neurais artificiais profundas, capazes de aprender representações complexas dos dados de entrada.

A motivação acerca do tema teve origem durante a observação de tecnologias de segurança como: Sistemas de vigilância por vídeo inteligente. Sistemas que utilizam câmeras de segurança equipadas com algoritmos de reconhecimento de objetos e pessoas para monitorar e analisar imagens em tempo real [2], sistemas de detecção de incêndio ou queimadas onde é alertado se está ocorrendo um incêndio [3].

Além do aplicativo "Google Lens" da Google, uma ferramenta com inteligência artificial capaz de reconhecer o conteúdo de imagens. Pode reconhecer raças de cães, espécies de plantas e até pontos turísticos de uma cidade, além disso é capaz de traduzir textos em imagens, placas e outros tipos de sinalização.

II. Conceitos Fundamentais

O desenvolvimento deste trabalho se dá em torno do aprendizado de máquina e do deep learning. Este que é uma subárea da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos que permitem que sistemas computacionais aprendam e melhorem automaticamente a partir de dados, sem serem explicitamente programados [4]. No aprendizado de máquina, em vez de escrever regras ou instruções específicas para realizar uma tarefa, o sistema é alimentado com um grande conjunto de dados (previamente tratado) e é capaz de aprender padrões e fazer previsões ou tomar decisões com base nesses dados.

Entre as etapas que envolvem o machine learning o tratamento de dados é essencial, um dos casos de tratamento de dados mal executado foi uma inteligência que tinha por objetivo detectar câncer de pele em pessoas. O banco de dados usado foi todo retirado de imagens da internet, porém, grande parte das imagens que continham a doença eram acompanhadas de uma régua para medir o tamanho das manchas, com isso o modelo entendeu que réguas causam doenças [5]. A partir deste exemplo é possível compreender a importância de um banco de dados completo, diverso e filtrado. Os dados funcionam como a alimentação do algoritmo, e o objetivo é montar o "cardápio" mais nutritivo possível.

Dessa forma, o próximo passo é a definição do algoritmo. Para este trabalho a abordagem será a do deep learning, que está intrinsecamente ligada às redes neurais. Estas que são modelos computacionais inspirados no funcionamento do cérebro humano. Essas redes consistem em unidades de processamento interconectadas, chamadas de neurônios artificiais ou perceptrons, que colaboram para realizar tarefas de processamento de informações. As redes neurais são compostas por camadas, sendo a camada de entrada responsável por receber os dados de entrada, as camadas ocultas realizam o processamento intermediário e a camada de saída fornece os resultados finais.

A. A arquitetura YOLO

O motivo da ascensão da YOLO se dá por conta da união entre precisão e agilidade. Tendo isto em vista é necessário detalhar o funcionamento por trás disso. A partir da versão 3.

A estrutura técnica da arquitetura YOLO envolve um backbone (espinha dorsal) baseado em uma CNN e camadas adicionais para detecção de objetos em imagens. O motivo de sua agilidade é por conta da inexistência de camadas densas, durante o processo de detecção, apenas no final para unir as informações coletadas; não necessariamente os neurônios de uma camada estão ligados a todos os outros de outra camada, possuindo no seu intermédio as camadas convolucionais

Backbone (Espinha dorsal):

- O backbone é responsável por extrair recursos (features) de nível mais baixo e médio da imagem de entrada.
- Geralmente é utilizado um modelo pré-treinado, como o DarkNet-53, que é uma rede neural convolucional profunda.
- O DarkNet-53 é composto por camadas convolucionais e de pooling, seguidas por camadas residuais (residual layers) que ajudam a capturar informações de alto nível.

Camadas intermediárias:

- Após o backbone, são adicionadas camadas intermediárias para combinar os recursos de diferentes escalas.
- Nesta camada a imagem é redimensionada, aumentando gradativamente seu tamanho. Assim, diferentes características podem ser observadas pelo algoritmo.
- Fazendo uso do FPN que incorpora camadas convolucionais adicionais e usa conexões de resíduos (skip connections) para combinar características de diferentes níveis da rede.

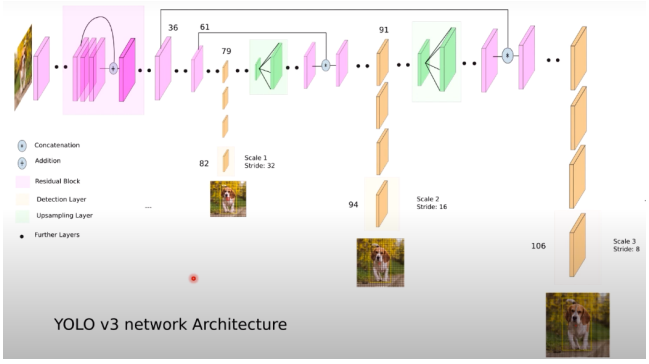
Camadas de detecção:

- Após as camadas intermediárias, são adicionadas camadas de detecção para gerar as bounding boxes (caixas delimitadoras) e as probabilidades de classe.
- Cada célula na grade de detecção é responsável por prever múltiplas caixas delimitadoras.
- Cada caixa delimitadora é representada por um vetor que contém coordenadas (x, y) do centro, largura (w) e altura (h) da caixa, confiança (score) da detecção e probabilidades de classe para cada objeto.

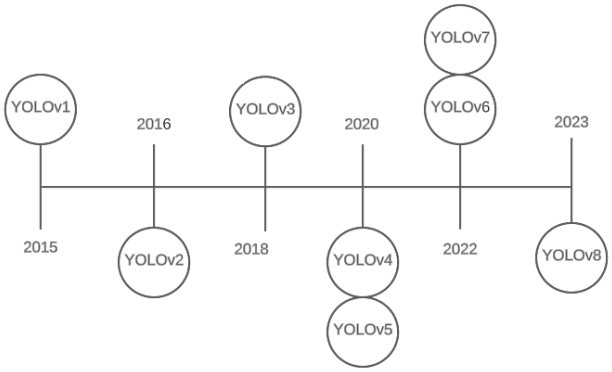
Camada de saída:

- A camada de saída recebe as predições das caixas delimitadoras e realiza o pós-processamento para obter as detecções finais dos objetos.
- Isso inclui filtrar caixas redundantes e aplicar técnicas como a supressão não máxima (non-maximum suppression) para selecionar as melhores caixas delimitadoras.
- As detecções resultantes incluem as classes dos objetos e suas localizações na imagem.

No geral, a estrutura da arquitetura YOLO envolve um backbone para a extração de recursos, camadas intermediárias para combinar características de diferentes escalas, camadas de detecção para prever as caixas delimitadoras e probabilidades de classe, e uma camada de saída para o pós-processamento das detecções. Essa estrutura permite que o YOLO detecte objetos em tempo real e com alta precisão.



B. Explicação e linha do tempo: Arquitetura YOLO



- YOLOv1: Desenvolvido na universidade de Washington rapidamente ganhou popularidade devido sua alta velocidade e precisão [14]
- YOLOv2: Melhorias em relação ao modelo original, incorporando Batch Normalization, Anchor Box e Dimension Cluster [15]
- YOLOv3: Melhorou ainda mais o desempenho usando um backbone mais eficiente, adicionando a pirâmide de características e fazendo uso da perda focal (Tecnologias que se popularizaram em 2017 com a RetinaNet) [16]
- YOLOv4: Foi implementada uma nova função de perda, o Mosaic Data augmentation e o Bag of Freebies [17]
- YOLOv5: Lançado pela Ultralytics, apresentou melhorias de desempenho tornando a detecção mais rápida, além disso foi implementada a segmentação panóptica e rastreamento de objetos. Também difere dos anteriores por ser baseado em Pytorch ao invés de Darknet [18]
- YOLOv6: Versão otimizada do YOLOv5, foi desenvolvido com foco em aplicações industriais, não considerado parte da série oficial YOLO já que foi desenvolvido pela empresa chinesa Meituan [19]
- YOLOv7: Superou todos os modelos anteriores, usa como base a arquitetura do YOLOv4, aprimorou ainda mais o Bag of Freebies. A significativa melhora se deu por conta de mudanças na arquitetura da rede, entre elas: a Agregação da camada eficiente estendida (E-ELAN), Técnicas de escala de modelo(Model Scaling Techniques) e o planejamento de reparametrização (Reparameterization Planning) [20]
- YOLOv8: Também desenvolvido pela Ultralytics (autores do YOLOv5), porém foi considerado pela comunidade como uma versão oficial. Possui resultados similares ao YOLOv7, porém mais estável e de fácil implementação. Implementada uma nova rede de Backbone, novo cabeçote de deteção sem âncora (anchor-free detection head) e nova função de perda

C. Termos e técnicas

- Batch Normalization: é uma técnica que normaliza os valores de entrada em cada lote durante o treinamento de redes neurais, melhorando a estabilidade, acelerando o treinamento e melhorando o desempenho do modelo.

- Anchor Boxes: gera várias caixas delimitadoras com diferentes tamanhos e proporções de aspecto, enquanto é centralizado em cada pixel. Essas caixas delimitadoras são chamadas de caixas de âncora. Durante o treinamento, cada objeto real presente na imagem de treinamento é associado a uma ou mais anchor boxes. Essa associação é feita com base na sobreposição entre as caixas delimitadoras dos objetos e as anchor boxes. Se uma anchor box tiver uma sobreposição significativa com um objeto real, ela é considerada como uma candidata a detectar esse objeto.
- Dimension Cluster: A Dimension Cluster (ou Clustering de Dimensões) é uma abordagem que visa determinar as dimensões (largura e altura) ideais das anchor boxes com base nas características dos objetos presentes nos dados de treinamento.
- Backbone: o termo Backbone refere-se à parte principal de uma arquitetura de rede neural convolucional (CNN) que é responsável pela extração de características de uma entrada. este geralmente consiste em várias camadas convolucionais e de pooling, que são empilhadas para formar uma hierarquia de representações visuais. Essas camadas convolucionais são projetadas para aprender recursos de baixo nível, como bordas e texturas, e recursos de nível mais alto, como formas e padrões complexos. Seu nome se dá pois serve como a ”espinha dorsal” da rede, fornecendo as características básicas que são posteriormente utilizadas por módulos ou camadas adicionais para tarefas específicas, como classificação, detecção de objetos, segmentação semântica, entre outras.
- Pirâmide de Características: Técnica usada para detectar e reconhecer objetos em diferentes escalas e níveis de detalhe. A construção da pirâmide se dá pelos seguintes passos: Reescala da imagem, onde a mesma é redimensionada para várias escalas diferentes aumentando ou diminuindo a resolução da imagem. Filtragem: Em cada escala são aplicados vários filtros para extrair características da imagem. Concatenação: As características extraídas de cada escala da pirâmide são combinadas e concatenadas para formar uma representação unificada das características da imagem em várias escalas.
- Perda Focal: Função de perda especializada em casos de desequilíbrio entre as classes de objetos, especialmente em casos de desequilíbrio entre as classes de objetos. Esta atribui pesos diferentes para cada parte da imagem, a ideia principal é reduzir a contribuição das amostras fáceis (background) e dar mais ênfase às amostras difíceis (objetos de interesse). Sendo assim os possíveis objetos têm um valor maior que o fundo da imagem, o que encoraja o algoritmo a buscar as regiões mais valiosas.
- Mosaic Data Augmentation: É uma técnica usada para aumentar o conjunto de dados de treinamento, envolve a combinação de múltiplas imagens em um único exemplo de treinamento. Essas imagens são selecionadas aleatoriamente do conjunto de dados original, formando um mosaico ou colagem de imagens. O mosaico criado combina informações de várias imagens, o que pode melhorar a robustez do modelo em relação a diferentes condições de iluminação, fundos e oclusões.
- Bag of Freebies: É uma coleção de técnicas adicionais que podem ser aplicadas para melhorar o desempenho de modelos de deep learning sem aumentar significativamente a carga computacional ou o custo de treinamento. Dentre estas técnicas temos: Aumento de dados (data augmentation), Pré-processamento de dados, Inicialização de pesos (weight initialization) entre outras técnicas.
- Agregação da camada eficiente estendida: A ELAN considera projetar uma rede eficiente controlando o caminho de gradiente mais curto e mais longo, para que redes mais profundas possam convergir e aprender de forma eficaz.
- Técnicas de escala de modelo: As técnicas de escala de modelo são abordagens usadas para ajustar o tamanho ou a complexidade de uma rede neural, a fim de melhorar seu desempenho ou eficiência. Essas técnicas podem ser aplicadas em várias dimensões do modelo, como o número de camadas, o número de unidades em cada camada, a resolução de entrada ou outros parâmetros relacionados. Às técnicas mais comuns são: Escala de largura (Width scaling), Escala de profundidade (Depth scaling), Escala de resolução (Resolution scaling) e Escala de hiperparâmetros (Hyper Parameter scaling)
- Planejamento de reparametrização: É uma técnica de natureza estatística que visa otimizar a reparametrização de uma distribuição paramétrica, permitindo uma inferência mais eficiente e precisa.

III. Trabalhos relacionados

- ”You Only Look Once: Unified, Real-Time Object Detection” por Joseph Redmon, Santosh Divvala, Ross Girshick e Ali Farhadi. Este trabalho apresenta o YOLO original, uma abordagem unificada para detecção de objetos em tempo real. O artigo descreve a arquitetura do YOLO, que realiza a detecção de objetos em uma única passagem pela rede neural. Ele discute a geração de bounding boxes (caixas delimitadoras) e a atribuição de probabilidades de classe para cada objeto detectado. O YOLO demonstrou um equilíbrio entre velocidade e precisão, tornando-se uma referência no campo da detecção em tempo real [6].
- ”Mask R-CNN” por Kaiming He, Georgia Gkioxari, Piotr Dollár e Ross Girshick. Este paper introduziu o Mask R-CNN, uma extensão do Faster R-CNN que adiciona a capacidade de segmentação em nível de pixel para o reconhecimento de objetos. Essa arquitetura permite a detecção, classificação e segmentação de objetos em imagens [7].
- ”Focal Loss for Dense Object Detection” por Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He e Piotr Dollár. Este estudo introduz o RetinaNet, uma arquitetura de detecção de objetos baseada em redes neurais convolucionais. O artigo destaca a função de perda chamada Focal Loss, que é projetada para lidar com o desequilíbrio de classes na tarefa de detecção de objetos. O RetinaNet

é capaz de detectar objetos em uma ampla variedade de escalas e obteve resultados superiores em desafios de detecção [8].

- ”ImageNet Classification with Deep Convolutional Neural Networks” por Alex Krizhevsky, Ilya Sutskever e Geoffrey E. Hinton. Este trabalho foi pioneiro no uso de redes neurais convolucionais profundas para a classificação de imagens em grande escala. Os autores propuseram uma arquitetura chamada AlexNet, que consiste em várias camadas convolucionais e de pooling, seguidas por camadas totalmente conectadas. O artigo demonstrou que o uso de redes neurais profundas com aprendizado supervisionado em grandes conjuntos de dados, como o ImageNet, levou a resultados notáveis na classificação de imagens [9].

IV. Objetivo

A principal contribuição deste trabalho é a investigação e implementação de uma abordagem de aprendizado profundo para o reconhecimento de objetos em imagens. Serão exploradas técnicas de pré-processamento de imagens, treinamento e otimização de modelos de CNN, bem como a utilização de conjuntos de dados rotulados para garantir a qualidade e a generalização do sistema de reconhecimento de objetos.

Dentre as arquiteturas amplamente utilizadas, as escolhidas e testadas para este projeto são: RetinaNet [8] e seu método “Focal Loss” que dá maior importância aos exemplos de treinamento mais difíceis, Mask R-CNN [7] que trabalha a segmentação semântica junto com a classificação para gerar resultados precisos sobre o objeto e sua localização na imagem e YOLO (You Only Look Once) [6] conhecida por sua velocidade realiza a detecção de objetos com apenas uma passada pela rede. Com a descoberta da arquitetura YOLO está tem tido um domínio no campo industrial e acadêmico, portanto será nosso principal alvo de estudo.

V. metodologia experimental

O primeiro passo de nosso projeto é fazer o tratamento dos dados, este que será feito selecionando as melhores imagens para o treinamento escolhido, é necessário ter cuidado com o fundo das imagens (estes devem ser diferentes para não criar um “vício” no algoritmo), além disso uma diversidade no design dos objetos é de extrema importância (no caso de reconhecer pessoas não é recomendável ter fotos apenas de uma pessoa, mesmo que seus arredores mudem). Estes dados serão, parcialmente, coletados da internet e posteriormente incrementados com fotos e vídeos pessoais.

Em seguida essas imagens serão divididas em treino e teste e então utilizadas no algoritmo para fazer a predição do que for enviado a ele. Para isso diversas bibliotecas serão usadas, dentre elas a Ultralytics: Responsável por modelar e modernizar a arquitetura YOLO, sendo a autora de diversas versões como a mais recente YOLOv8. Cv2 [10] (responsável pelo tratamento das imagens, desenvolvida pela Intel possui diversas tecnologias relacionadas à visão computacional), entre outras que podem ser necessárias.

A arquitetura de estudo, como dito anteriormente, será a YOLO, está enquanto arquitetura é extremamente ágil, sendo a mais usada para detecções ao vivo (câmeras de segurança ou outros tipos de monitoramento e até pilotagem de veículos), sua velocidade de detecção foi testada em monitoramentos feitos por drone e teve resultados satisfatórios [11]. Em sua versão 3 um sistema foi desenvolvido para que pudesse detectar figuras de diversos tamanho em uma mesma imagem, algo similar a RetinaNet pois trabalha de maneira mais elaborada a profundidade da imagem, sendo usada para detectar múltiplos objetos (mesmo que sobrepostos, ou algo como um carro na frente do outro) ou cenários monótonos sem deixar de lado a velocidade. Já na versão 5, foi introduzida tecnologia de rastreamento e a segmentação semântica foi de fato implementada a arquitetura, assemelhando-se a Mask R-CNN que até então era amplamente utilizada na medicina/biomedicina por conta da sua precisão em demarcação, marcando exatamente a silhueta do objeto desejado, no contexto médico um grande uso é na detecção de câncer de mama [12]. E mais recentemente foi lançada a versão 8, que trabalha com classificação, detecção, segmentação, rastreamento e estimativa de pose. É possível observar que, rapidamente, a arquitetura YOLO vem incorporando as tecnologias de outros, sem perder sua característica de agilidade que dá seu nome You Only Look Once [13].

Ao final será feita uma comparação entre as versões para saber qual possui a melhor eficiência na relação tempo/desempenho e será avaliado se as versões foram de fato se superando.

A. Base de Dados

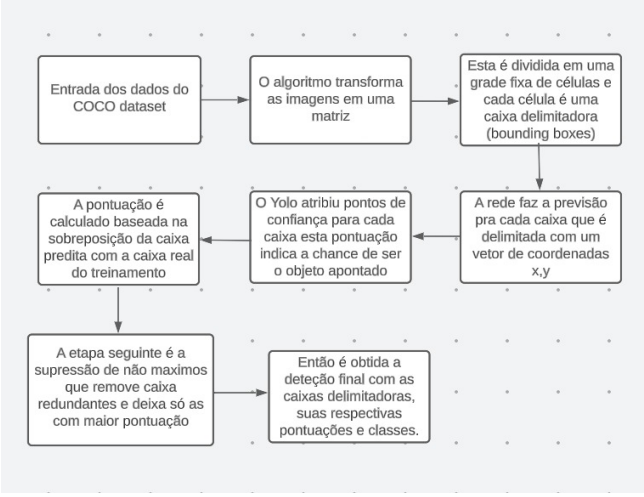
Para o treinamento do modelo o Dataset COCO (Common objects in context) será usado. O COCO é um dos maiores conjuntos de imagens anotadas e foi criado para obter o estado da arte dos algoritmos de Reconhecimento de Objetos. Todas as suas imagens estão classificadas e os objetos segmentados, facilitando o seu uso para o desenvolvedor e sendo extremamente úteis para a leitura do algoritmo.

O COCO tem mais de 200 mil imagens anotadas permitindo que algoritmos de Deep Learning tenham maior eficiência, visto que são “gulosos”, consomem muita informação para ter um bom rendimento. uma descrição de como o COCO funciona pode ser feita a partir dos seguintes passos: primeiro, uma tag que classifica a imagem como um todo (gato, carro, ...); segundo, uma caixa delimitadora (normalmente quadrado) selecionando o objeto; terceiro, um polígono que segmenta a imagem detectando os objetos; quarto, descrição escrita da imagem (“gato deitado no sofá”).

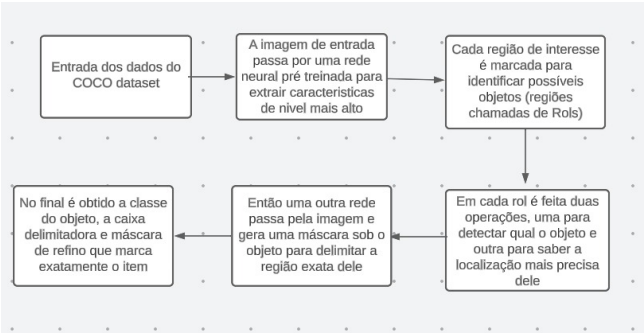
B. Protocolo de validação

O método Hold-out será utilizado devido a grande gama de informações que temos, assim tornando seu treinamento mais rápido, exigindo menos capacidade computacional e tendo sua eficácia comparável a de métodos de Cross-Validation.

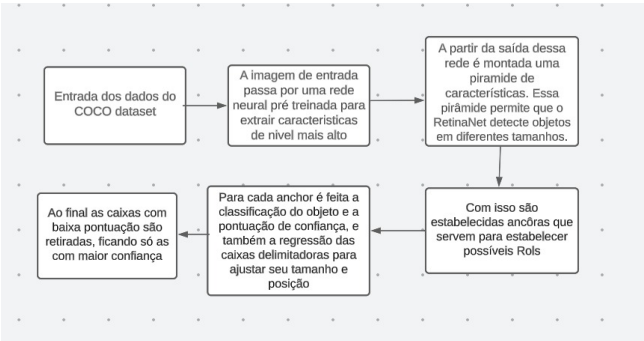
C. Yolo



D. Mask R-CNN

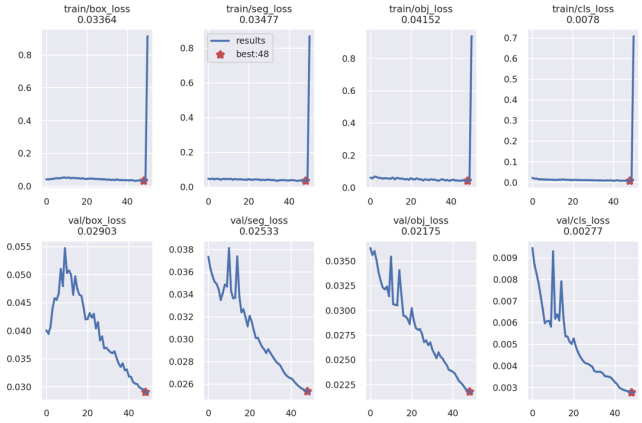


E. RetinaNet

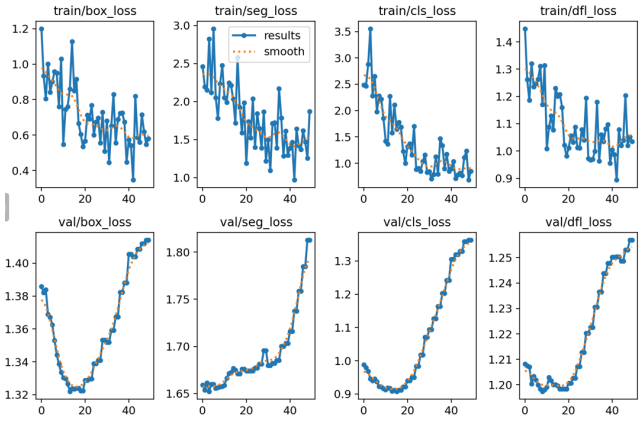


VI. Resultados e Discussão

A. YOLOv5



B. YOLOv8



VII. Final

Como resultado final é esperado poder classificar quais objetos estão presentes na mídia proposta, seja ela foto ou vídeo, e a posição em que ele se encontra. Espera-se um mínimo de 60% de confiabilidade para qualquer objeto aparente. Os principais elementos alvo são: pessoas, garrafas, crachás e mochilas.

Além disso é necessário que o código seja escrito de forma clara e comentada para compreensão de terceiros, servindo como base e material de estudo para outros que desejarem aprimorar suas habilidades no assunto, assim como o paper deve possuir todas as informações necessárias para compreensão do que foi feito.

Além de toda parte didática que deseja-se passar, o intuito é mostrar como a arquitetura YOLO foi inovadora no campo de detecção de objetos e como suas versões foram incorporando funções de seus "concorrentes" sem perder suas características primárias de agilidade e eficiência.

References

[1] O. S. Lucena, "Estudo e Implementação de Técnicas para Reconhecimento de Objetos em Imagens", UFCG

[2] F. Majeed, Investigating the efficiency of deep learning based security system in a real-time environment using YOLOv5, Science Direct

[3] B. Kim, A Video-Based Fire Detection Using Deep Learning Models, Chonbuk National University, Jeonju 54896, Korea

[4] Y. LeCun, Deep learning, Nature, 2015
[5] B. Christian, The Alignment Problem: Machine Learning and Human Values, W. W. Norton Company, 2021
[6] J. Redmon, S. Divvala, R. Girshick e A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, IEEE, 2016
[7] K. He, G. Gkioxari, P. Dollár e R. Girshick, Mask R-CNN, Cornell University, 2017
[8] T. Lin, P. Goyal, R. Girshick, K. He e P. Dollár, Focal Loss for Dense Object Detection, Cornell University, 2018
[9] A. Krizhevsky, I. Sutskever e G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, University of Toronto
[10] G.R. Bradski, Intel's Computer Vision Library: applications in calibration, stereo segmentation, tracking, gesture, face and object recognition, IEEE
[11] C. Jiang, Object detection from UAV thermal infrared images and videos using YOLO models, Science Direct, 2022
[12] J. Ying, Detection and classification the breast tumors using mask R-CNN on sonograms, Medicine (Baltimore)
[13] X. Cheng, RetinaNet With Difference Channel Attention and Adaptively Spatial Feature Fusion for Steel Surface Defect Detection, IEEE, 2020
[14] J. Redmon, S. Divvala, R. Girshick e A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, IEEE, 2016
[15] J. Redmon, S. Divvala e A. Farhadi, YOLO9000: Better, Faster, Stronger, Cornell University, 2016
[16] J. Redmon, S. Divvala e A. Farhadi, YOLOv3: An Incremental Improvement, Cornell University, 2018
[17] C. Y. Wang, YOLOv4: Optimal Speed and Accuracy of Object Detection, Cornell University, 2020
[18] M. Karthi, Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset, IEEE, 2021
[19] L. Chuyi, YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications, Cornell University, 2022
[20] B. Alexey, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, Cornell University, 2022