Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Bacharelado em Ciências de Computação – 2023.1
SCC0910 – Tópicos Avançados em Ciências de Computação I

Docente: Prof Dr Fernando Pereira dos Santos

Análise de Desempenho YOLOv3 e YOLOv5

You Only Look Once

Diógenes Silva Pedro	11883476
Guilherme Lourenço de Toledo	11795811
Pedro Augusto Ribeiro Gomes	11819125

São Carlos, 23 de Junho de 2023

1. Resumo

Este trabalho apresenta uma análise comparativa abrangente entre YOLOv3 e YOLOv5, dois algoritmos de ponta para detecção de objetos em tempo real. O estudo concentra-se em avaliar as diferenças dos modelos considerando as métricas de Precision e Recall, bem como as matrizes de confusão de cada um dos modelos. Devido aos recursos computacionais limitados, foram utilizados a versão YOLOv5s do modelo YOLOv5. Em resumo, ao decidir entre esses modelos (e suas versões de diferentes tamanhos disponíveis) , é necessário ponderar cuidadosamente as necessidades e prioridades específicas do projeto.

2. Introdução

Anteriormente, foi implementada a rede neural YOLO (You Only Look Once) para detecção de objetos com o objetivo de entender sua estrutura intrínseca e compreender como é feita a tarefa de multidetecção e classificação de objetos. Agora, o objetivo é usar duas versões pré-treinadas distintas desse mesmo modelo - YOLOv3 e YOLOv5 - e analisar as diferenças entre eles.

Primeiramente, será explicado o dataset COCO, o qual foi utilizado para testar os diferentes modelos. Em seguida, serão apresentados os aspectos teóricos que distinguem as duas versões do modelo e por fim os resultados das medidas e testes feitos por cada uma das versões sobre o mesmo dataset.

O código fonte do trabalho pode ser encontrado neste repositório.

3. Dataset Para Validação

O conjunto de dados COCO (Common Objects in Context) é uma coleção extensa de imagens usado para tarefas de reconhecimento de objetos, segmentação e descrição de cenas. Com mais de 330.000 imagens anotadas, cada uma contendo 80 categorias de objetos e 5 legendas descritivas, o COCO é amplamente utilizado em pesquisas de visão computacional.

Ele fornece informações detalhadas, como coordenadas de *bounding boxes*, máscaras de segmentação, pontos-chave e suas posições (se disponíveis), para ajudar a treinar e avaliar modelos de detecção e segmentação de objetos. Além disso, o COCO também abrange a segmentação de elementos de plano de fundo e anotações de pontos-chave para pessoas.

O conjunto possui duas categorias principais: "coisas" e "elementos". As "coisas" são objetos facilmente manipulados, como pessoas, bicicletas, carros e motocicletas. Já os "elementos" são itens de fundo ou ambientais, como céu, árvores e estradas. No entanto, o COCO sofre de um desequilíbrio de classes, onde algumas classes possuem um número muito maior de imagens do que outras. Esse desequilíbrio pode causar viés no treinamento de modelos de aprendizado de máquina, afetando seu desempenho em classes menos frequentes. Para mitigar esse problema, são utilizadas técnicas como oversampling, undersampling e geração de dados sintéticos.

4. Diferenças dos Modelos:

4.1. YOLOv3

Assim como foi visto com a implementação do YOLOv1, o YOLOv3 utiliza uma arquitetura de rede neural convolucional para realizar a detecção de objetos em tempo real. A rede é dividida em duas partes principais: uma parte de convolução que extrai características das imagens de entrada e uma parte de detecção que prevê os retângulos delimitadores (*bounding boxes*) e as classes dos objetos presentes na imagem.

A rede de convolução é responsável por extrair características hierárquicas da imagem em diferentes níveis de escala. Isso permite que o YOLOv3 detecte objetos de diferentes tamanhos e resoluções. A parte de detecção é baseada em um mapa de características de alta resolução e realiza a predição dos retângulos delimitadores e das classes dos objetos usando convoluções adicionais.

O YOLOv3 foi feito utilizando Darknet, um framework de aprendizado profundo escrito em Liguagem C e CUDA.

Uma das principais melhorias do YOLOv3 em relação às versões anteriores é a utilização de múltiplas escalas de detecção. Ele divide a imagem de entrada em grades (grids) e gera caixas delimitadoras em diferentes escalas em cada grid. Isso aumenta a capacidade do modelo de detectar objetos em várias resoluções e melhora a precisão da detecção.

O YOLOv3 também utiliza um mecanismo chamado "*non-maximum suppression*" para eliminar detecções redundantes. Após a detecção inicial dos objetos, esse mecanismo seleciona as melhores detecções, removendo aquelas que têm sobreposição significativa com outras caixas delimitadoras de maior confiança.

4.2. YOLOv5

Uma das principais mudanças no YOLOv5 é a arquitetura da rede neural convolucional. Ele utiliza uma arquitetura baseada em EfficientNet, uma família de arquiteturas de redes neurais conhecida por manter um equilíbrio entre desempenho e eficiência computacional.

Ao contrário do YOLOv3, o YOLOv5 opera em uma única escala fixa para a detecção de objetos. No entanto, ele utiliza técnicas de *data augmentation*, como redimensionamento aleatório, para melhorar a capacidade do modelo em lidar com objetos de diferentes tamanhos. Essa abordagem de escala única simplifica o modelo e o torna mais eficiente.

O YOLOv5 alcança uma boa precisão na detecção de objetos, embora os resultados possam variar dependendo do conjunto de dados e da configuração do modelo. Além disso, o YOLOv5 oferece recursos adicionais, como detecção em 3D e suporte para dispositivos móveis, por meio da versão YOLOv5-Nano.

Uma vantagem do YOLOv5 é que ele possui uma implementação em código aberto e uma API amigável, o que facilita sua utilização e personalização para diferentes aplicações. Isso permite que os desenvolvedores e pesquisadores possam adaptar o modelo de acordo com suas necessidades específicas.

Há uma variedade de modelos YOLOv5, com diferentes tamanhos para abrangerem diferentes hardwares:

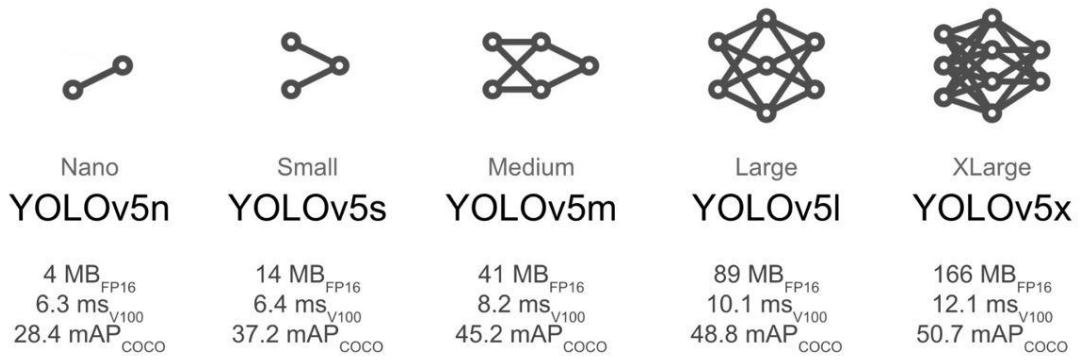

Figura1: Diferentes versões do YOLOv5

5. Resultados

Veremos a seguir uma comparação das métricas utilizadas para os modelos YOLOv3 e YOLOv5s. Para a nossa comparação, usaremos a média das métricas *Precision* e *Recall*, estas que são relevantes para a tarefa de detecção de objetos. Além disso, mostraremos as matrizes de confusão de cada modelo, estas que revelam uma eficiência muito boa para ambos os modelos.

Uma das métricas utilizadas para avaliar a performance dos modelos de detecção de objetos é a *Precision*. Ela mede a proporção de objetos corretamente identificados em relação ao total de objetos detectados. Uma alta taxa de *Precision* indica que o modelo apresenta poucos falsos positivos, ou seja, poucos objetos são erroneamente identificados como objetos de interesse. Temos para a YOLOv3 e YOLOv5s, respectivamente, 0.507 e 0.411. Ao comparar YOLOv5s e YOLOv3, observa-se que YOLOv5s demonstra uma taxa de *Precision* ligeiramente inferior. Isso significa que a YOLOv3 é um pouco mais precisa na identificação dos objetos de interesse, reduzindo a ocorrência de falsos positivos em relação ao YOLOv5s.

Outra métrica relevante é o *Recall*, que mede a proporção de objetos corretamente identificados em relação ao número total de objetos existentes. Um alto *Recall* indica que o modelo é capaz de detectar a maioria dos objetos presentes na imagem, minimizando os falsos negativos, ou seja, objetos que são erroneamente classificados como negativos ou não detectados. Nesse quesito, a YOLOv3 apresenta um desempenho um pouco superior, com uma taxa de *Recall* um pouco maior em comparação com a YOLOv5s. Isso significa que a YOLOv3 consegue identificar um maior número de objetos de interesse em relação à YOLOv5s.

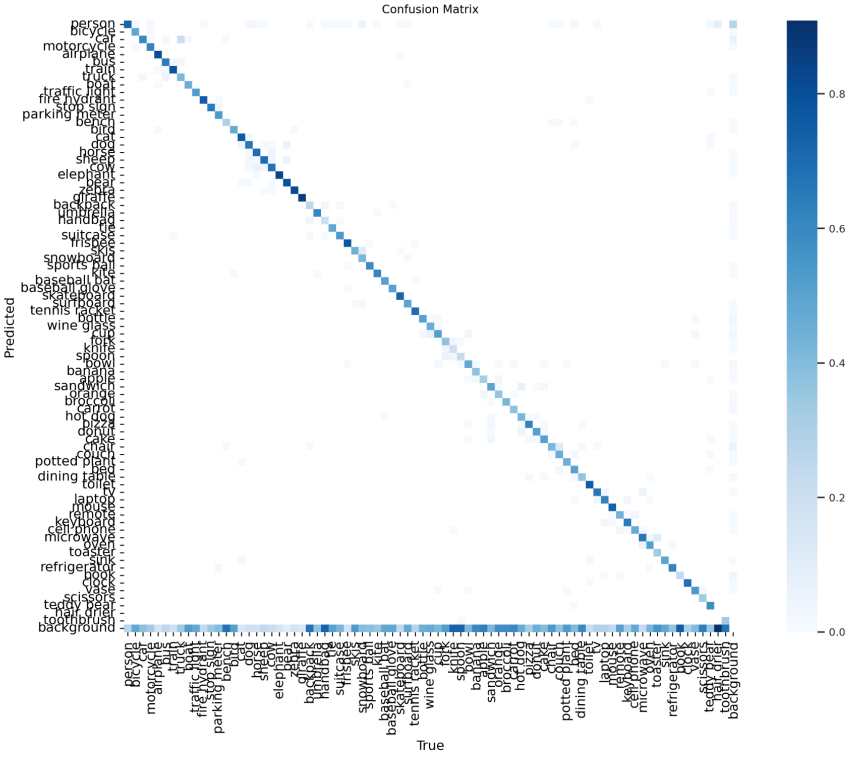

Figura 2: Matriz de Confusão da YOLOv5s

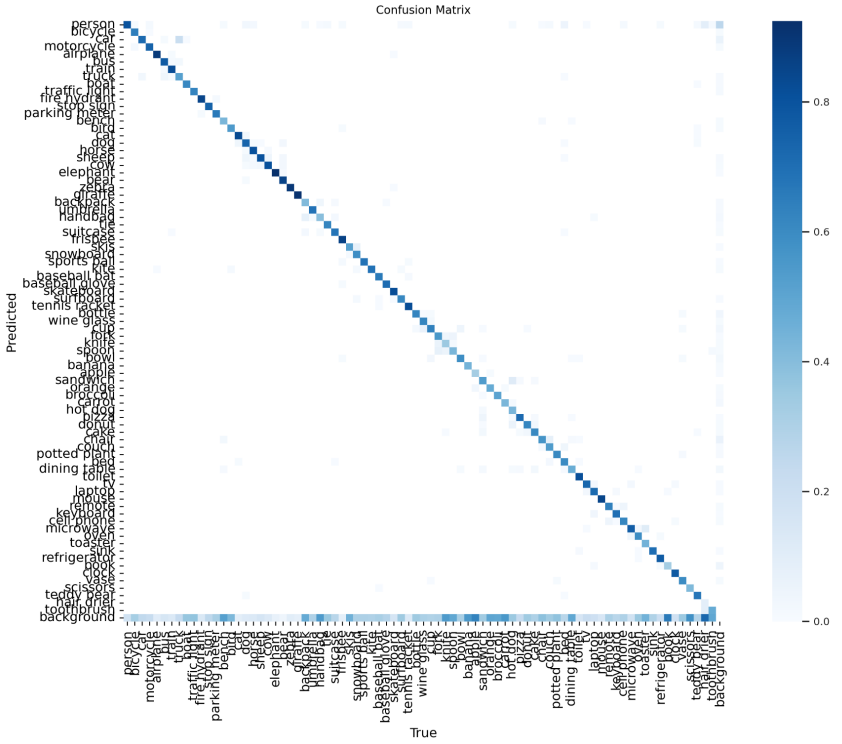

Figura 3: Matriz de Confusão da YOLOv3

Ao avaliar as matrizes de confusão de cada modelo, que fornecem uma visão detalhada do desempenho em relação às diferentes classes de objetos, é possível observar como cada modelo se comporta individualmente. Podemos perceber que a YOLOv3 acerta mais verdadeiros positivos do que a YOLOv5s.

A seguir, veremos os gráficos de *Precision-Recall* de cada um modelos:

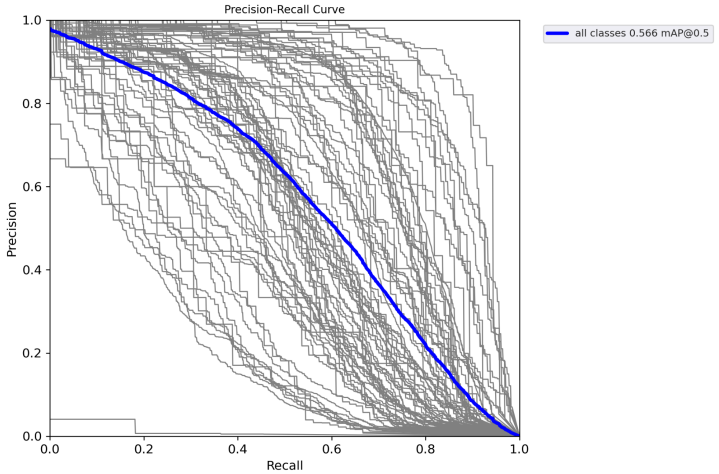

Figura 4: Gráfico Precision-Recall YOLOv5s

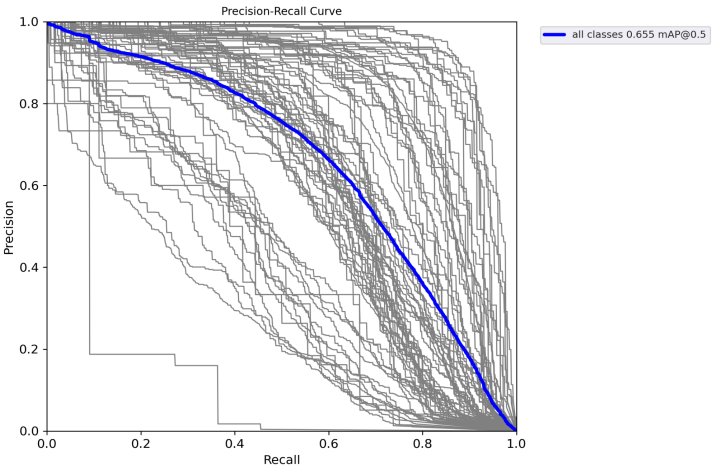

Figura 5: Gráfico Precision-Recall YOLOv3

Ao analisar um gráfico Precision-Recall, você pode observar como essas duas métricas se relacionam entre si. Um modelo ideal teria tanto uma Precision quanto um Recall de 1, indicando que todos os objetos de interesse foram detectados corretamente e sem falsos positivos. No entanto, na prática, há um compromisso entre essas duas métricas, e é comum que aumentar a *Precision* resulte em uma diminuição do *Recall* e vice-versa.

Em um gráfico Precision-Recall, você verá uma curva que conecta diferentes pontos que representam diferentes configurações ou resultados de um modelo. Idealmente, você deseja que essa curva esteja o mais próximo possível do canto superior direito do gráfico, indicando alta *Precision* e alto *Recall*. Isso significa que o modelo é capaz de detectar a maioria dos objetos corretamente e minimizar os falsos positivos.

A interpretação do gráfico *Precision-Recall* depende do objetivo específico do modelo. Se você estiver focado em maximizar a detecção de objetos, um modelo com uma curva Precision-Recall mais próxima do canto superior direito seria preferível, mesmo que a Precision seja um pouco menor. Por outro lado, se a minimização de falsos positivos for mais importante, um modelo com uma curva *Precision-Recall* mais próxima do canto inferior esquerdo seria mais adequado, mesmo que o Recall seja um pouco menor.

Podemos perceber então que a YOLOv3 é superior a YOLOv5s nessa análise, uma vez que a sua curva se aproxima mais do canto superior direito.

Ambos os modelos são notáveis em suas capacidades de detecção de objetos em tempo real e oferecem resultados impressionantes. A escolha entre YOLOv5s e YOLOv3 dependerá das necessidades e prioridades específicas de cada projeto, levando em consideração fatores como as métricas apresentadas, capacidade de processamento e memória.

6. Conclusão

Ambos os modelos, YOLOv5s e YOLOv3, possuem méritos distintos que devem ser considerados ao escolher entre eles. A YOLOv5s se destaca por sua eficiência em termos de processamento e uso de memória. Essa arquitetura foi projetada para ser mais leve e ágil, permitindo a detecção de objetos em tempo real com requisitos de hardware menos exigentes. Isso é particularmente vantajoso em cenários onde a velocidade de processamento é crucial, como em sistemas de vigilância, veículos autônomos ou aplicações em tempo real. Além disso, é importante mencionar que, ao utilizar a versão "small" da YOLOv5, estamos empregando uma variante otimizada e compacta do modelo, que possui recursos computacionais ainda mais reduzidos.

No entanto, embora a YOLOv5s tenha essa vantagem em eficiência, é importante ressaltar que a YOLOv3 apresentou um desempenho um pouco melhor nas métricas de Precision e Recall. Portanto, ao decidir entre esses modelos, é necessário ponderar cuidadosamente as necessidades e prioridades específicas do projeto. Se o foco principal for a precisão na detecção de objetos, mesmo que isso resulte em requisitos mais altos de processamento e memória, a YOLOv3 pode ser a escolha mais apropriada.

É importante ressaltar que, naturalmente, uma versão maior da YOLOv5 apresentaria um desempenho e uma eficiência melhores do que a YOLOv3. No entanto, como já mencionado acima, foi utilizada nas simulações, a versão YOLOv5s, a qual é possível de ser alocada em memória RAM de um computador pessoal para que os experimentos pudessem ser feitos com os recursos disponíveis.

7. Referências

[1] Joseph Redmon, Ali Farhadi: YOLOv3: An Incremental Improvement
Disponível em: https://arxiv.org/abs/1804.02767

[2] Ultralytics: YOLOv5
Disponível em: https://github.com/ultralytics/yolov5

[3] Aladdin Persson: YOLOv3 from Scratch
Disponível em: https://www.youtube.com/watch?v=Grir6TZbc1M