

# **Detecção de Personagens em Animação utilizando YOLOv7**

**Marcos Raphael Bernardi de Souza**

[marcos.rafael@usp.br](mailto:marcos.rafael@usp.br)

*Atenção: este documento foi gerado como parte de avaliação na disciplina SCC5920 - Mineração de Dados Não Estruturados, visando avaliar os conhecimentos teóricos do(a) estudante a respeito do tema selecionado na disciplina (mineração de textos ou imagens). Não foi desenvolvido com o objetivo de publicação em conferências ou periódicos.*

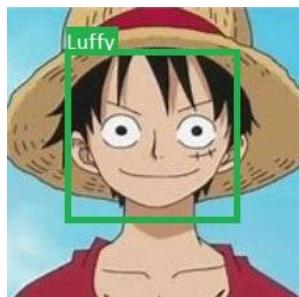
**Resumo.** Este projeto descreve o processo de treino de um conjunto de dados personalizado para a detecção e reconhecimento de personagens em animações. Esta tarefa é relevante para lidar com a sumarização baseada em personagens ou identificação de uso impróprio de direitos autorais. Foi utilizada uma base de dados personalizada, criada a partir de imagens disponíveis no google e na kaggle. Foram feitos dois métodos de treino e utilização do LabelImg para anotações do dataset. O modelo escolhido para detecção foi o YOLOv7 que mostrou-se adequado para atingir o objetivo deste projeto.

## **1. Definição do Problema**

É perceptível o aumento do volume de dados multimídia produzidos e acessíveis. O crescente volume de dados gerados pelo meio digital trouxe à tona áreas de pesquisas focadas em analisar, extrair, recomendar e/ou sumarizar as informações provenientes desses dados. Na última década uma das áreas com bastante demanda foi o reconhecimento, em especial, o facial, devido ao número crescente de aplicações na área de segurança [1]. Também devemos citar a detecção de Objetos de Interesse (OOI - Objects of Interest). Detectar o OOI em uma série de imagens já é uma técnica madura em vigilância e segmentação de objetos [2].

Dado que, como entrada, um conjunto de vídeos de determinada animação, é necessário realizar seleção de segmentações utilizando como um dos critérios a presença ou não de personagens específicos. Neste caso, a utilização de técnicas para reconhecimento facial humano dificilmente pode ser aplicada para cartoons, animes, mangas, hqs e games em geral. Para detecção de faces de personagens de animes é comum encontrar utilização do *Haar-cascade* no *OpenCV*, porém treinar uma base de dados específica por esse método necessita de imagens ‘boas’ e ‘ruins’, o que influencia significativamente no treino.

Dentro desse contexto, utilizamos técnicas usuais para detecção de objetos com aplicação em identificação de personagens. Para exemplificar, neste projeto, usamos a base de dados *One Piece image classifier* disponibilizada na plataforma Kaggle e imagens que buscamos no google.



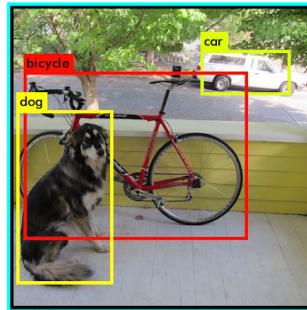
**Figura 1. Exemplo de detecção e identificação na imagem**

## **2. Pré-processamento**

Nossa base de dados é composta por imagens do personagem Luffy e de imagens aleatórias salvas através do resultado da pesquisa ‘one piece’ no google. Nesse conjunto são 654 imagens somente do Luffy e 118 da pesquisa do google com outros personagens ou cenários expandidos.

O método escolhido foi o YOLOv7 (You Only Look Once), pois a detecção de objetos feita pelo yolo identifica a posição e classe. Além disso, o Yolo mostra caixas delimitadoras nas imagens e vídeos que estão sendo analisados. Para utilização da base de dados selecionada com esse método é necessário o seguinte fluxo de trabalho:

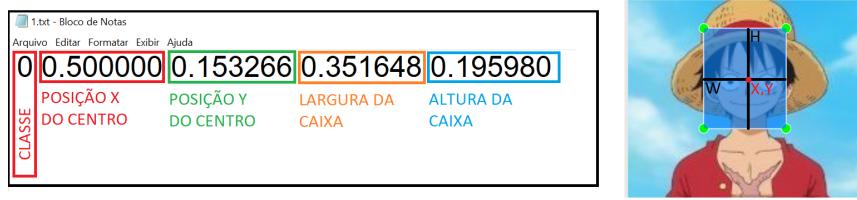
1. Criar uma nova pasta chamada *dataset*;
2. Criar 3 sub-pastas: *train* e *valid*;
3. Dentro de cada sub-pasta criar 2 sub-pastas: *images* e *labels*.
4. Salvar dentro de *train* as imagens que serão utilizadas para treino.
5. Salvar dentro de *valid* as imagens que serão utilizadas para validação.



**Figura 2. Exemplo de detecção e classificação do YOLOv7.**

A divisão de imagens de treino e validação é na proporção de 80% para treino e 20% para validação. Para realização do treino é necessário ter anotações dos objetos de interesse em cada imagem em *train* e *test* seguindo o padrão em arquivo texto com coordenadas e sua classificação. No YOLOv7 as anotações para realizar o treino são geralmente feitas utilizando imagens com cenário completo e identificando as regiões de interesse. Essas anotações podem ser feitas através de softwares de catalogação disponíveis, neste caso foi utilizado o LabelImg.

O formato de anotação para o YOLOv7 é salvo em um arquivo texto (.txt) e que cada linha contém uma marcação. Cada linha é composta por até 5 campos separados por espaço. O primeiro campo é a classe da caixa anotada, o segundo e terceiro é a coordenada relativa (X,Y) do centro da caixa e os últimos dois campos são as dimensões relativas. Exemplo na figura 3:



**Figura 3. Exemplo do padrão de anotação para o YOLOv7**

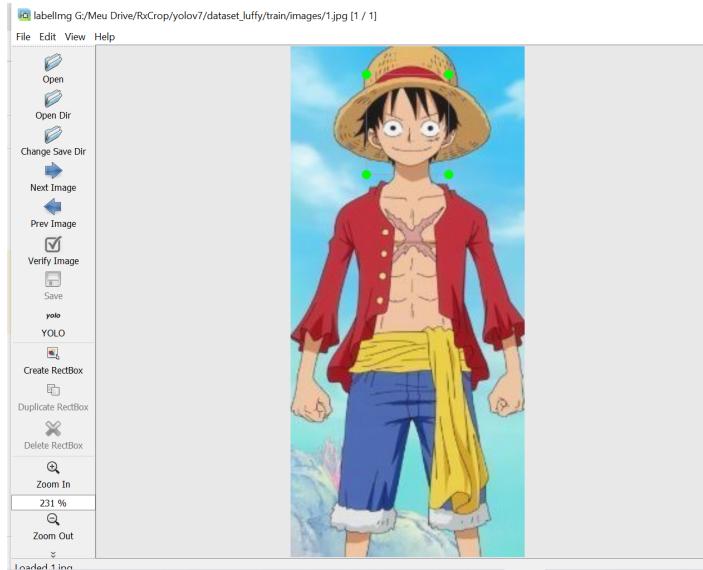
Para o primeiro treino foi considerado que 90% de qualquer imagem do dataset utilizado faz parte da nossa região de interesse, pois as imagens utilizadas contém apenas o personagem. Na figura 4 vemos um exemplo da diferença entre o tipo de anotação de um *dataset* usual para YOLO que faz anotação da região de interesse (ROI) com cenário e o que estamos utilizando. A vantagem para tal consideração é que não foi necessário fazer anotações das imagens manualmente, economizando muito tempo, neste caso. Foram utilizadas para treinamento 524 imagens *train* e 130 imagens *valid*.



**Figura 4. Diferença de tipo de dataset**

O segundo treino foi feito com as anotações do *LabelImg*. Em cada imagem do *dataset* fizemos anotações apenas do rosto do personagem Luffy. As anotações foram feitas apenas nas imagens ‘normais’ sem inversão de cor ou rotações diversas. Além disso, também consideramos imagens compostas com

outros personagens e selecionamos especificamente a região do personagem de interesse. Foram utilizadas 165 imagens *train* e 86 *valid*, não seguindo a proporção 80/20 do primeiro treino.



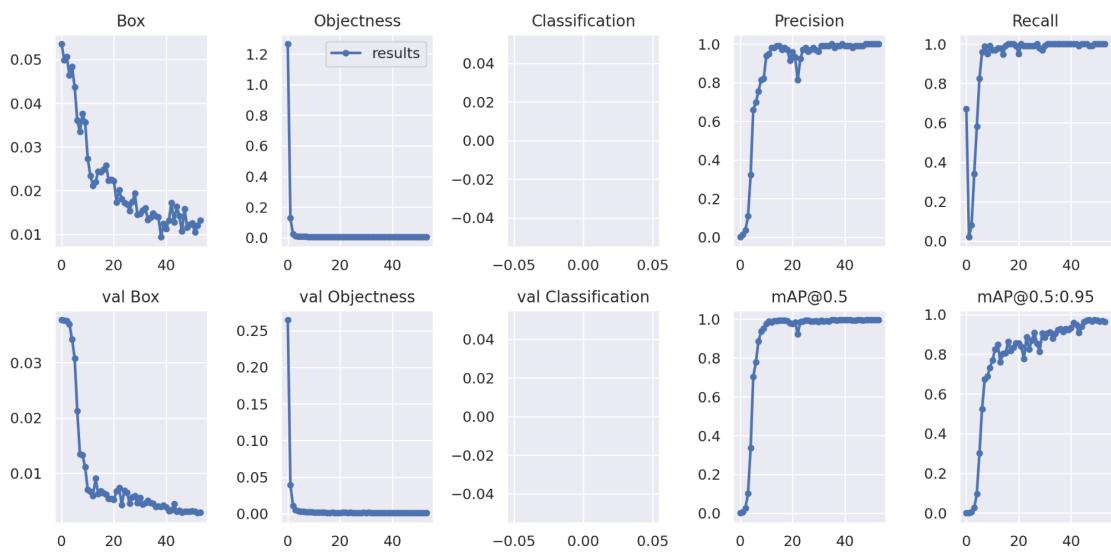
**Figura 5. Tela do LabelImg**

### 3. Extração de Padrões

Nosso projeto consiste numa tarefa de detecção, ou seja, extrair características de uma imagem. Essas características da imagem podem ser objetos, cantos ou pessoas, isso dependerá da aplicação. Ao invés de prever uma classe presente numa imagem, como a classificação, a detecção identifica a posição e a classe. Neste projeto é tratado especificamente para o caso de um personagem da animação One Piece para exemplificação, porém o método aqui proposto poderá ser aproveitado em diversas outras aplicações.

### 4. Pós-Processamento

O treino do YOLOv7 com *dataset* personalizado gera uma pasta *train* que contém o arquivo *Pytorch* do modelo. Além disso, a pasta tem alguns gráficos mostrando o resultado da classificação, confiança obtida nas detecções de validação e precisão durante as épocas do treino.



**Figura 6. Resultados gráficos do treino 1**

Nos gráficos da Figura 6 podemos verificar que durante o primeiro treino após, aproximadamente, 10 épocas a precisão das detecções ficou acima de 80% e se manteve na maioria das épocas seguintes acima de 95%. Isso mostra que o conjunto de treino convergiu rapidamente, olhando o gráfico *Recall* também verifica-se que após 10 épocas já estava quase 100%. Contudo, vemos que nas imagens de validação, Figura 7, o YOLOv7 interpretou a detecção de maneira distinta da esperada.

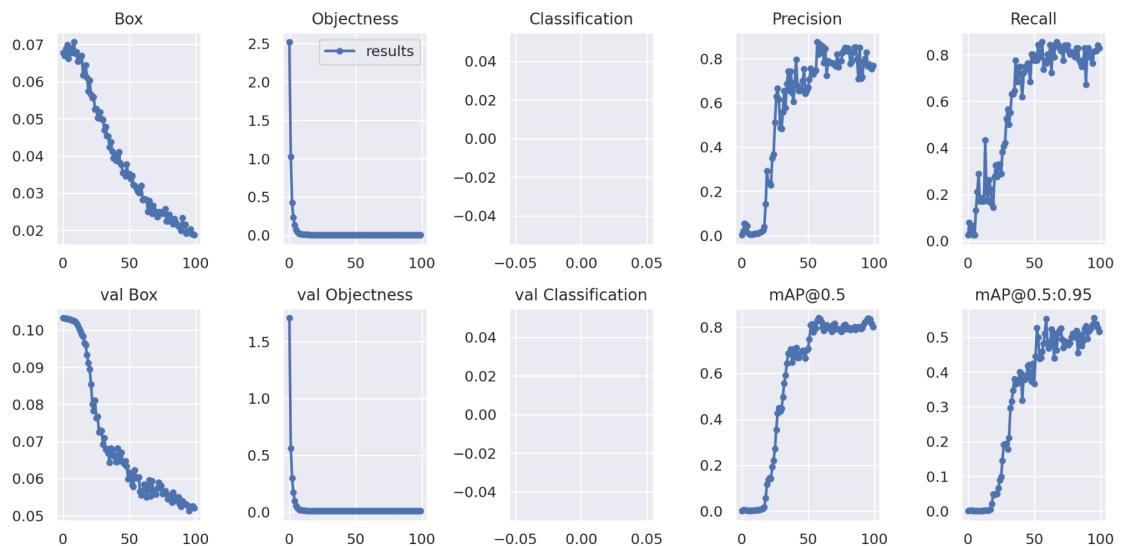


**Figura 7. Detecção no conjunto validação do treino 1**

A estratégia adotada como tentativa de fazer anotações de maneira automática do *dataset* utilizado não funcionou para o treino do YOLOv7.

O segundo treino esteve com confiança entre 75-85% após 50 épocas. Conforme pode ser observado na figura 8, pode-se diminuir para aproximadamente 75 o número de

épocas do treino, mas 100 épocas teve um resultado satisfatório. Neste conjunto o modelo de detecção se comportou conforme o esperado e localizou a posição e classificou especificamente o personagem Luffy, Figura 9.



**Figura 8. Resultados do treino 2**



**Figura 9. Detecção no conjunto validação do treino 2**

Utilizando o arquivo *Pytorch* podemos fazer a inferência do modelo em imagens e vídeos. O documento de saída das detecções realizadas pelo YOLOv7 são no formato texto (.txt). De forma análoga ao formato das anotações, cada linha do arquivo é uma caixa de detecção e os 5 primeiros campos são: classe, posição relativa X do centro, posição relativa Y do centro, comprimento da caixa e altura da caixa. Além disso, tem um campo adicional que é a confiança na detecção.

CLASSE	RESULTADO DA REGIÃO DE DETECÇÃO	CONFIANÇA DA DETECCÃO
0	0.47338 0.437934 0.881944 0.389757	0.691468

**Figura 10. Exemplo do padrão de anotação da inferência**

## 5. Uso de Conhecimento

A identificação de personagens animados pode facilitar a sumarização baseada em personagens, verificação do uso de imagem de maneira imprópria e sistemas de recomendação dessas obras.

## 6. Considerações Finais

O objetivo do projeto foi alcançado com a utilização do YOLOv7, obtendo sucesso no treinamento do *dataset* personalizado para detecção do personagem Luffy em imagens e vídeos. Ao utilizar um *dataset* personalizado é necessário fazer anotações para o treino. A ferramenta LabelImg facilita esse processo, porém ainda necessita de tempo e atenção para essa tarefa.

Em algumas imagens houveram mais de uma detecção e sendo algumas dessas um falso positivo. Outros personagens foram detectados e classificados de maneira equivocada, nesse projeto não tratamos desses casos, porém alguns caminhos seriam possíveis no pós-processamento. Também pode-se realizar treinamento com mais imagens e outras classes/personagens.

## Referências

- [1] Galindo, T. P. S. (2018) “Análise de desempenho de YOLOv3 quanto a variação de fatores da cena em imagens para detecção de personagens em vídeo”, Repositório UFOP, João Pessoa.
- [2] Adarsh Kowdle1, Kuo-Wei Chang2, Tsuhan Chen1 “Video categorization using object of interest detection”.