

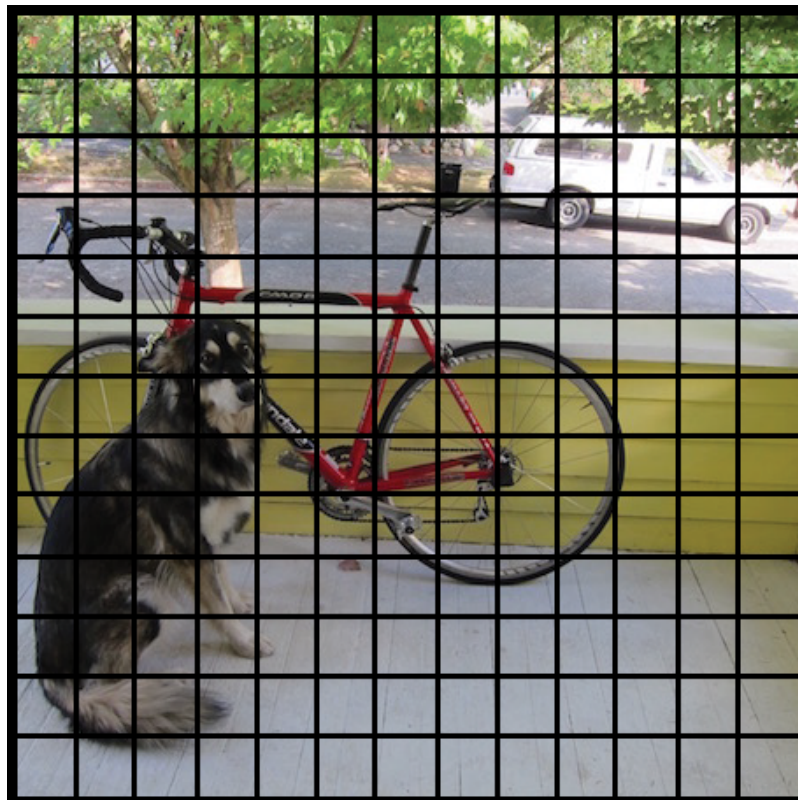
Visão Computacional

YOLO - You Only Look Once (Você Só Olha Uma Vez)

Detector e Classificador de objetos

- Usa Convolutional Neural Network - CNN, que é um ótimo classificador de imagens
- Criado em Darknet, o framework criado por seus desenvolvedores
- É capaz de detectar e classificar objetos em tempo real

YOLO divide a imagem numa grid de 13 por 13 células:



Cada uma dessas células são responsáveis por fazer a predição de 5 caixas delimitadoras. Uma caixa delimitadora descreve um retângulo que engloba um objeto.

YOLO também exibe um score de confiança que informa o quão certo ele está de que dentro da caixa delimitadora existe um objeto.

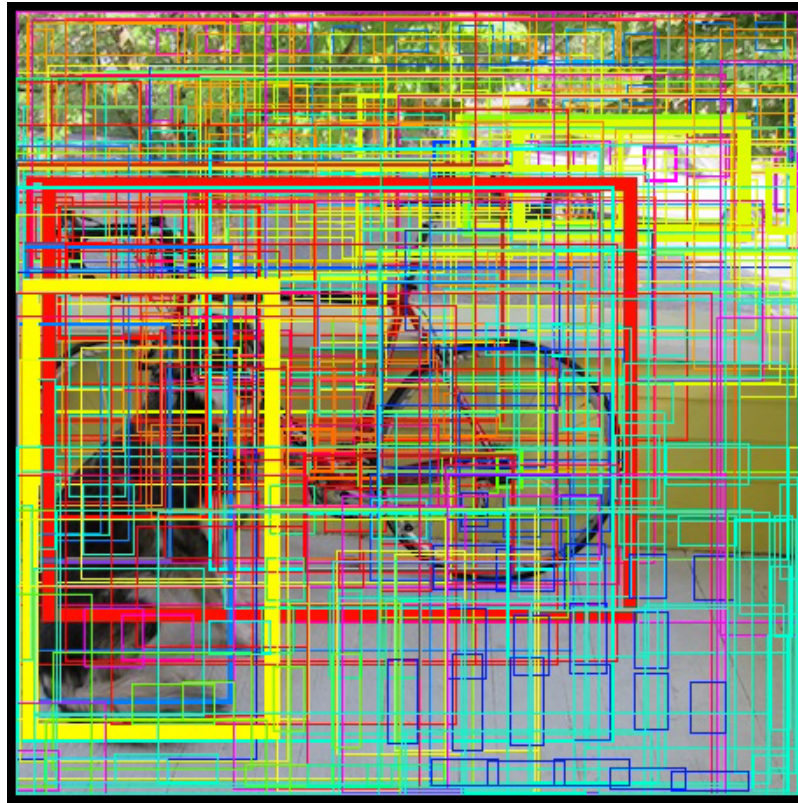


Para cada caixa delimitadora, a célula também prever uma Classe. Essa parte funciona exatamente como qualquer outro Classificador: gera a probabilidade de que seja uma Classe do universo de classes que ele conhece.

Classes como:

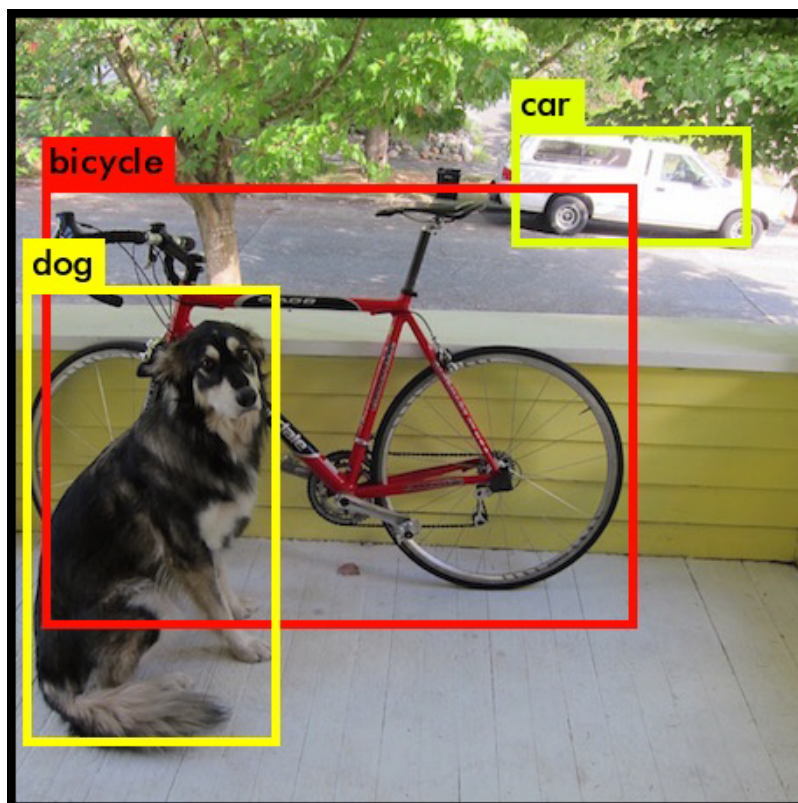
- bicicleta
- barco
- carro
- gato
- cachorro
- pessoa
- e assim por diante...

O score de confiança para uma caixa delimitadora e a predição da classe são combinados num único score final que nos diz a probabilidade de que uma caixa delimitadora contém um objeto de um tipo específico, por exemplo "cachorro":

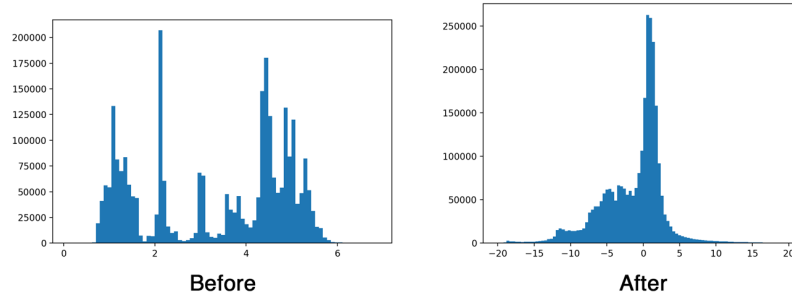


Uma vez que existe uma grid de $13 \times 13 = 169$ células, e cada uma das células fazem a predição de 5 caixas delimitadoras, obtém-se 845 caixas delimitadoras no total. E naturalmente a grande maioria dessas caixas terão um score de confiança muito pequeno, de forma que as únicas caixas que permanecem são as que possuem 30% ou mais de score de confiança (é possível modificar esse limite dependendo de quão acurado você deseja que o detector seja).

A predição final é:



YOLO usa uma tecnica de normalização chamada de *batch normalization* depois das camadas do modelo Convolutacional. A figura a seguir é um exemplo real onde a primeira imagem é da primeira camada convolutacional sem *batch normalization* e a segunda com:



Downloads:

- <http://bit.ly/yolov3weights> (<http://bit.ly/yolov3weights>).
- http://bit.ly/yolov3_projeto (http://bit.ly/yolov3_projeto).

Fontes:

- <https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/> (<https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/>).
- <https://pjreddie.com/darknet/yolo/> (<https://pjreddie.com/darknet/yolo/>).
- <http://machinethink.net/blog/object-detection-with-yolo/> (<http://machinethink.net/blog/object-detection-with-yolo/>).
- <https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection> (<https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection>).
- <https://medium.com/@xslittlegrass/almost-real-time-vehicle-detection-using-yolo-da0f016b43de> (<https://medium.com/@xslittlegrass/almost-real-time-vehicle-detection-using-yolo-da0f016b43de>).
- <https://github.com/KleinYuan/easy-yolo> (<https://github.com/KleinYuan/easy-yolo>).
- <https://medium.com/diaryofawannapreneur/yolo-you-only-look-once-for-object-detection-explained-6f80ea7aaa1e> (<https://medium.com/diaryofawannapreneur/yolo-you-only-look-once-for-object-detection-explained-6f80ea7aaa1e>).
- https://www.youtube.com/watch?v=TgeX-AF7_DE (https://www.youtube.com/watch?v=TgeX-AF7_DE).
- <https://www.youtube.com/watch?v=HbD9e6-qzko> (<https://www.youtube.com/watch?v=HbD9e6-qzko>).
- <https://www.youtube.com/watch?v=SO4tjl43Ob4> (<https://www.youtube.com/watch?v=SO4tjl43Ob4>).