

Análise de Imagens e Visão Computacional

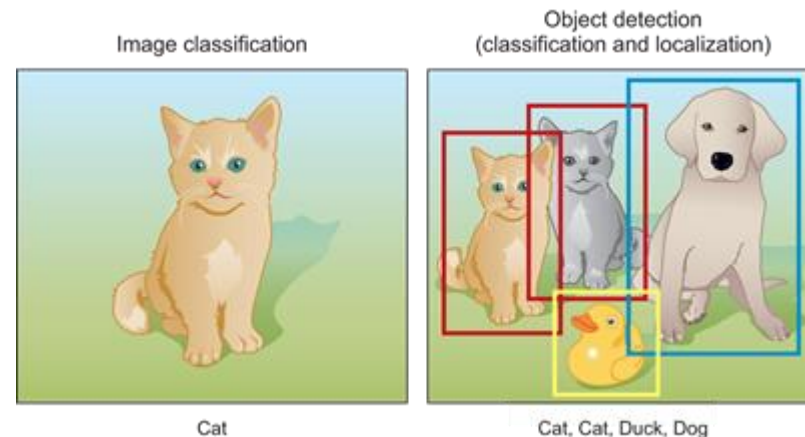
Prof. Henrique Batista da Silva

Introdução a Detecção de Objetos

Object Detection

Em muitas situações, existem vários alvos na imagem que nos interessam. Não queremos apenas classificá-los, mas também queremos obter suas posições específicas na imagem. Em visão computacional, nos referimos a tarefas como detecção de objetos.

Veja a Figura: diferença entre as tarefas de classificação de imagens e detecção de objetos.



Classificação: o classificador fornece a probabilidade da classe (gato)
Detecção de objeto: o detector fornece as coordenadas do bounding box (4 bbox neste exemplo) e as classes previstas (2 gatos + pato + cachorro).

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Object Detection

Existem muitas aplicações: veículos autônomos (localizar outros veículos, pedestres, rodovias e obstáculos); robôs (detecção de um objeto alvo/interesse); segurança (detecção de intrusos ou bombas)

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Componentes de um modelo de detecção de objetos

Object Detection

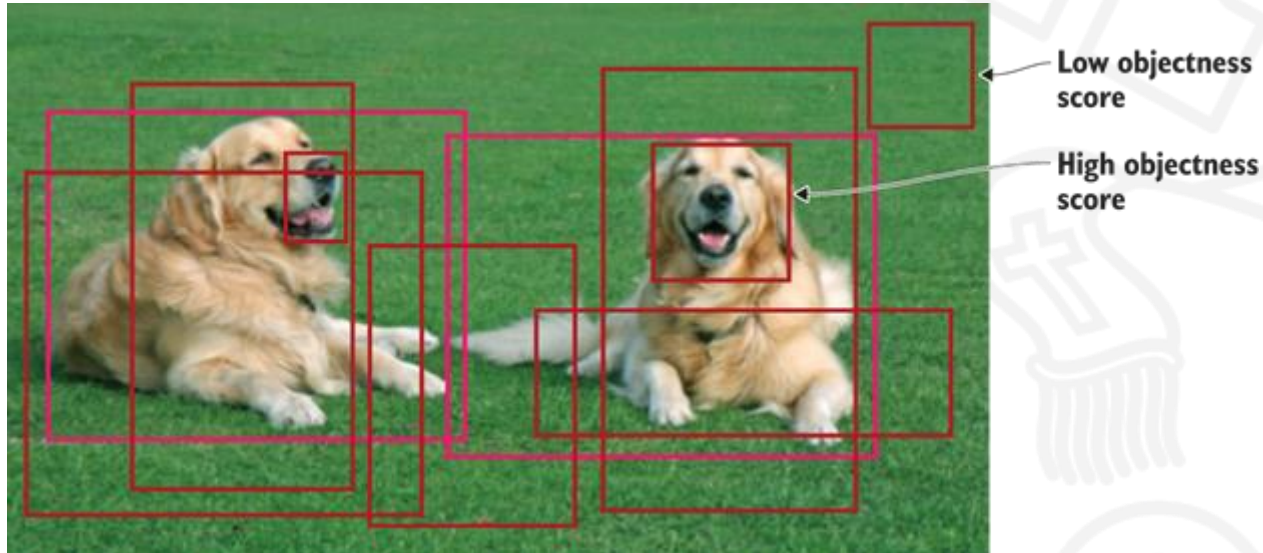
Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Proposta de região - um modelo é usado para gerar regiões de interesse a ser processada posteriormente pelo sistema. São regiões que a rede acredita que podem conter um objeto. Bounding boxes (bbox) são então passadas ao longo das camadas de rede para processamento posterior.

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Proposta de região



Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

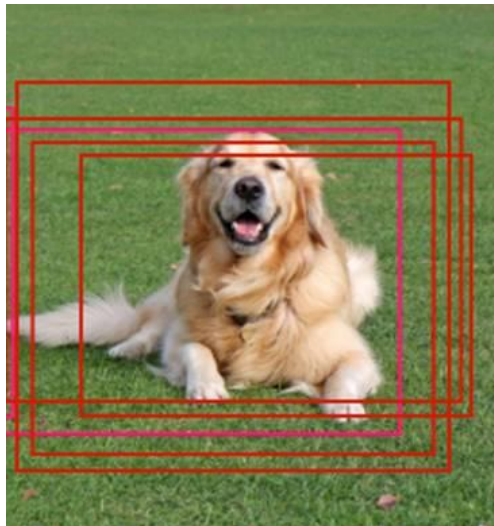
- Feature extraction e network predictions – features são extraídas para cada um dos bbox, elas são avaliadas e é determinado se e quais objetos estão presentes. Normalmente, utiliza-se modelos pré-treinados de classificação para extrair features, pois eles tendem a generalizar muito bem (ImageNet).

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Feature extraction e network predictions

Detector de objetos prevendo 5
bbox para o cachorro na imagem



As coordenadas do bbox são representadas como a seguinte tupla (x, y, w, h) . Onde “x” e “y” são as coordenadas do ponto central do bbox e “w” e “h” são a largura e a altura.

Predição de classe: esta é a função softmax clássica que prevê a probabilidade de classe para cada objeto

Object Detection

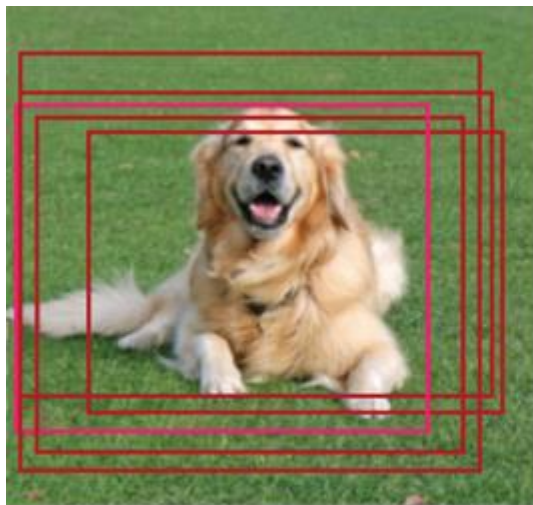
Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Non-maximum suppression (NMS) – nesta etapa, o modelo provavelmente encontrou vários bbox para o mesmo objeto. NMS ajuda a evitar a detecção repetida da mesma instância, combinando a sobreposição em um único bbox para cada objeto.

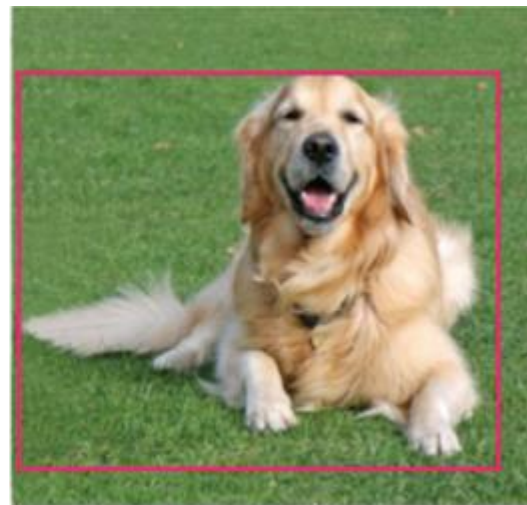
Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Non-maximum suppression (NMS)



Predictions before NMS



After applying non-maximum suppression

Vários bbox detectam o cão na imagem. Após NMS, apenas o bbox que melhor se ajusta ao objeto permanece e os demais são ignorados, pois têm grandes sobreposições com o bbox selecionado. NMS busca o bbox com a predição **máxima** e **suprime** os demais.

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Métricas de avaliação - Mean average precision (mAP), curva de recall e precisão (curva PR) e interseção sobre união (IoU).
- Ao avaliar o desempenho de um detector, usamos duas métricas de avaliação principais: 1) FPS (quadro por segundo) para medir a velocidade de detecção da rede e 2) mAP para medir a precisão da rede.

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Curva de recall e precisão (curva PR) e interseção sobre união (IoU): É uma medida que avalia a sobreposição entre dois bbox: o bbox do **groundtruth** e o bbox previsto pela rede. Ao aplicar o IoU, podemos dizer se uma detecção é válida (Verdadeiro Positivo) ou não (Falso Positivo).

Object Detection

Veja no link abaixo como é a produção de **groundtruth**, procedimento chamado de anotação de imagens, para object detection.

<https://github.com/tzutalin/labelImg>

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Object Detection


Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Curva de recall e precisão (curva PR) e interseção sobre união (IoU):



Predicted person bounding box

Ground truth person bounding box

$$\text{Score} = \frac{\text{Area of overlap}}{\text{Area of Union}}$$


Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Curva de recall e precisão (curva PR) e interseção sobre união (IoU):



Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Qual é o limite de sobreposição pode ser considerada válida: Este limite é um valor ajustável dependendo do desafio, mas 0,5 é um valor padrão. Isso significa que se o IoU estiver acima desse limite é considerado um **verdadeiro positivo (TP)** e se estiver abaixo, é considerado um **falso positivo (FP)**.

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Calculamos a precisão e recall da seguinte forma:

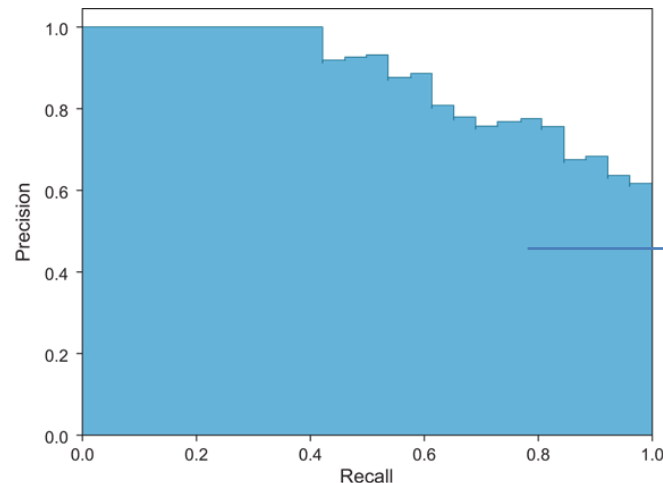
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

Object Detection

Normalmente, existem quatro componentes de uma estrutura de detecção de objetos:

- Depois de calcular a precisão e o recall para todas as classes, a Curva PR é então plotada da seguinte forma:



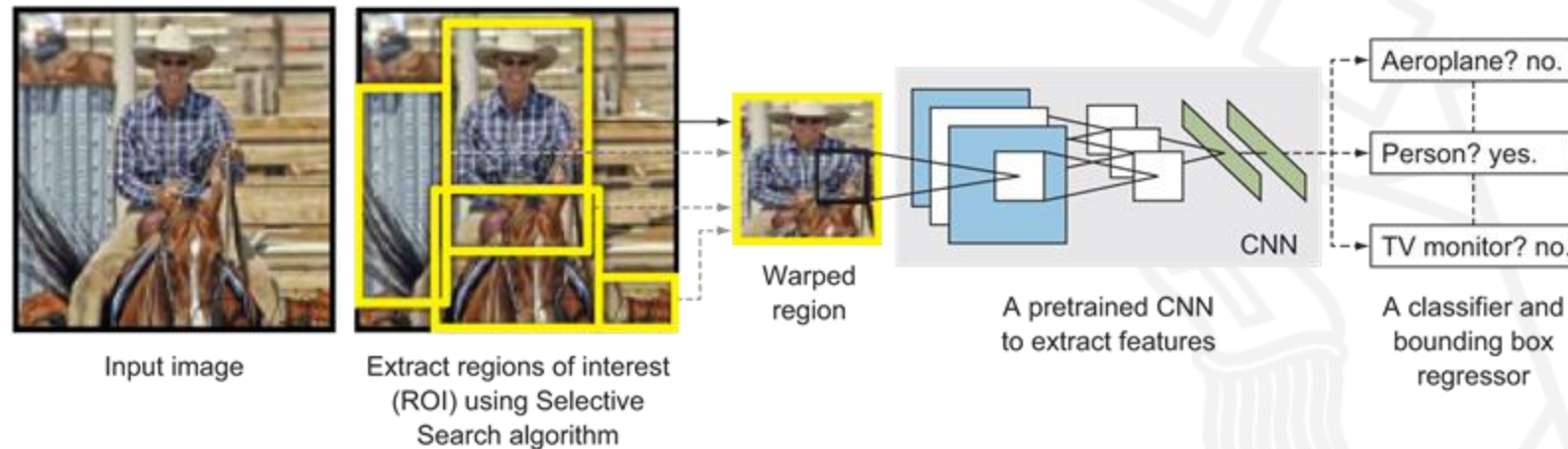
Average Precision (AP):
área sob a curva para
cada classe de objetos.
E Mean Average Precision
(mAP) é a média para
todas as classes.

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Region-Based Convolutional Neural Networks (R-CNNs)

R-CNNs

Técnica de detecção de objetos desenvolvida por Ross Girshick et al em 2014: “Rich feature hierarchies for accurate object detection and semantic segmentation”



Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

R-CNNs

Baseda em quatro componentes:

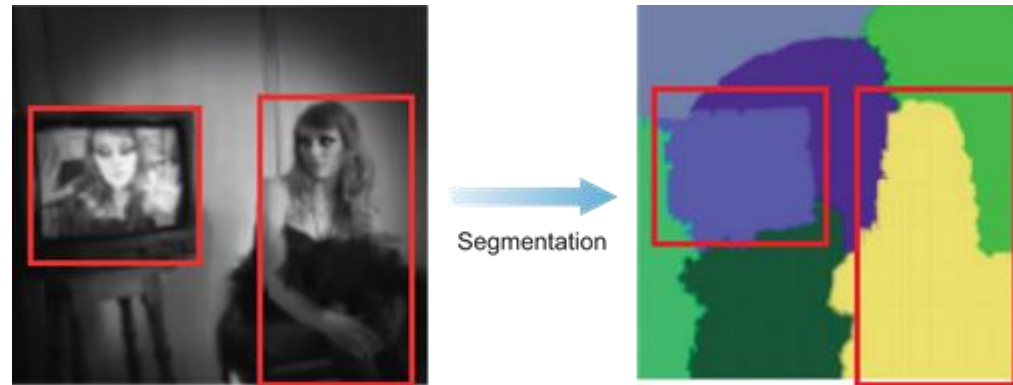
- Extract regions of interest (RoI) - regiões que têm alta probabilidade de conter um objeto. Utiliza o algoritmo de Selective Search. As regiões de interesse são distorcidas para ter um tamanho fixo (as CNNs exigem um tamanho de imagem de entrada fixo).

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

R-CNNs

Baseda em quatro componentes:

- Extract regions of interest (RoI) - O algoritmo de Selective Search é baseado na ideia de segmentação de imagens para encontrar regiões que possam conter objetos.



Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

R-CNNs

Baseda em quatro componentes:

- Feature Extraction - rede convolucional pré-treinada no topo para extrair features de cada região candidata.
- Módulo de classificação - treine um classificador como o Support Vector Machine (SVM) para classificar as detecções de candidatos com base nas features.

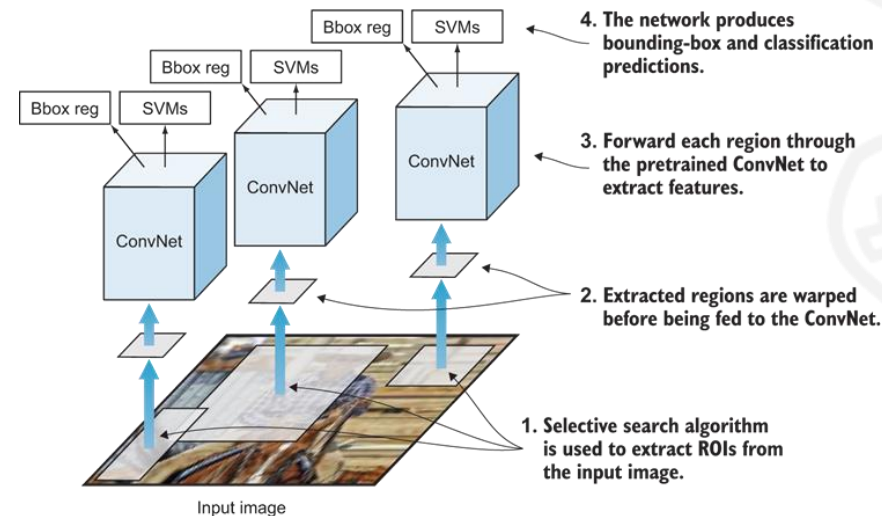
Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

R-CNNs

Baseada em quatro componentes:

- Localização: predição (por meio de regressão) das coordenadas “x” e “y” e das larguras “w” e “h” de cada bbox

As etapas 2, 3 e 4 são treináveis (R-CNN). A etapa 1 não é baseada em aprendizado.



Utiliza-se técnicas de classificação para predição das classes e de regressão para a predição das coordenadas (quadro valores)

R-CNNs

Desvantagens das R-CNNs:

- É um método muito lento para detecção de objetos (algoritmo selective search gera muitos objetos candidatos).
- O pipeline de treinamento contém múltiplos estágios: extração de features, classificação (classes) e regressão (coordenadas dos bbox).
- O treinamento é caro computacionalmente (memória e CPU-GPU).

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Fast R-CNN

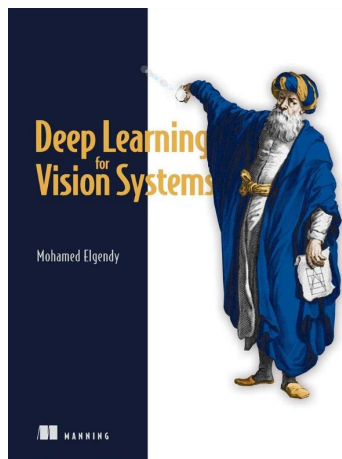
Evolução da R-CNN proposta em 2015.

- Agora, aplica-se a etapa de feature extraction primeiro, e depois a detecção de regiões. Assim, a ConvNet é executada em toda a imagem ao invés de executar em cada região de interesse (antes chegava a 2000 regiões)
- Elimina o uso de SVM para classificação e executa Softmax.

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Fast R-CNN

Para mais detalhes sobre a família de técnicas R-CNN para object detection, veja capítulo 7 da referência:



Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Single Shot Detection (SSD)

Single Shot Detection (SSD)

Proposta em 2016 no paper: SSD: Single Shot MultiBox Detector de C. Szegedy et. al.

A R-CNN é composta de múltiplos estágios, conforme vimos anteriormente. Detectores como o SSD (estágio único) as camadas convolucionais fazem ambas as predições diretamente (one shot).

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Single Shot Detection (SSD)

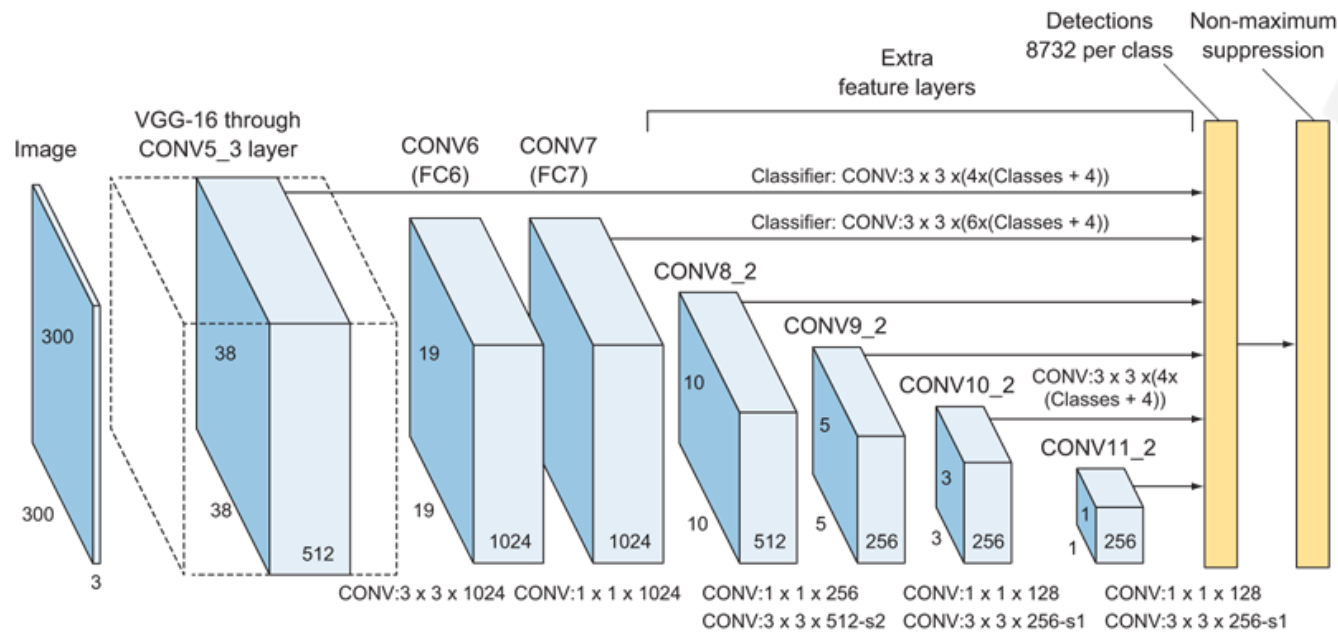
A arquitetura do modelo SSD é composta por três partes principais:

- Base network para extrair feature maps: uma rede pré-treinada (rede VGG16). Outras redes como VGG19 e ResNet podem ser usadas.
- Multi-scale feature layers: Essas camadas diminuem de tamanho progressivamente para permitir previsões de detecções em várias escalas.
- Non-maximum suppression: eliminar bbox sobrepostos

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Single Shot Detection (SSD)

A arquitetura do modelo SSD é composta por três partes principais:



Observe que as camadas de convolução 7, 8, 9, 10 e 11 fazem previsões que são inseridas diretamente na camada NMS

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Single Shot Detection (SSD)

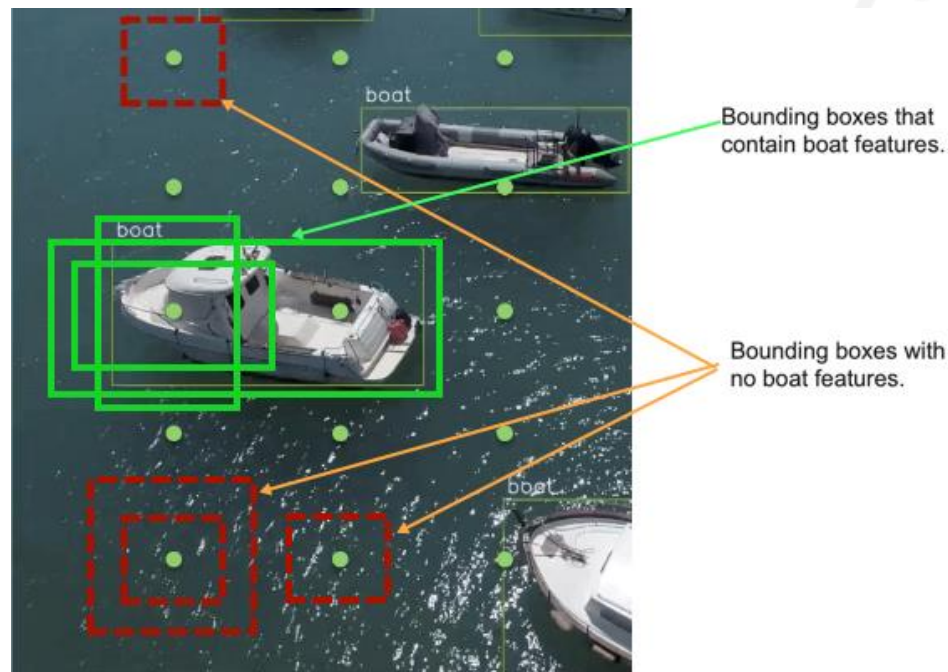
Como a rede faz previsões:

- SSD sobrepõe âncoras ao redor da imagem e, para cada âncora, a rede criará bbox. No SSD, as âncoras são chamadas de Priors.
- A rede analisará cada bbox como uma imagem separada e buscará por features das classes.
- Quando features de alguma classe é encontrada, ela envia as coordenadas e a classificação para a cama NMS.
- A camada NMS irá eliminar todos excessos de bbox com maior overlap com o groundthruth.

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Single Shot Detection (SSD)

Como a rede faz previsões:



Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

You Only Look Once (YOLO)

You Only Look Once (YOLO)

Existem várias versões, mas a versão mais recente é a YOLOv4

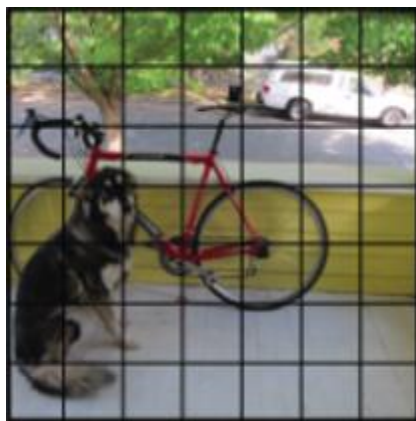
Uma das técnicas mais rápidas para detecção de objetos. Ideal para detecção em tempo real.

YOLO não utiliza o conceito de região da R-CNN, ao invés, divide a entrada em um grid e faz previsões apenas de um número limitado de bbox.

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

You Only Look Once (YOLO)

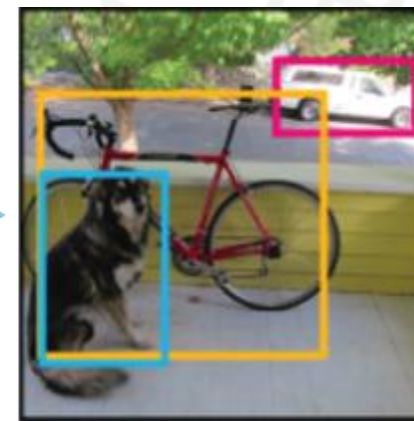
O YOLO divide a imagem em grid, prevê objetos para cada célula e usa NMS para finalizar as previsões.



Split the image into grids



Predict bounding boxes and classifications



Final predictions after non-maximum suppression

Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

You Only Look Once (YOLO)

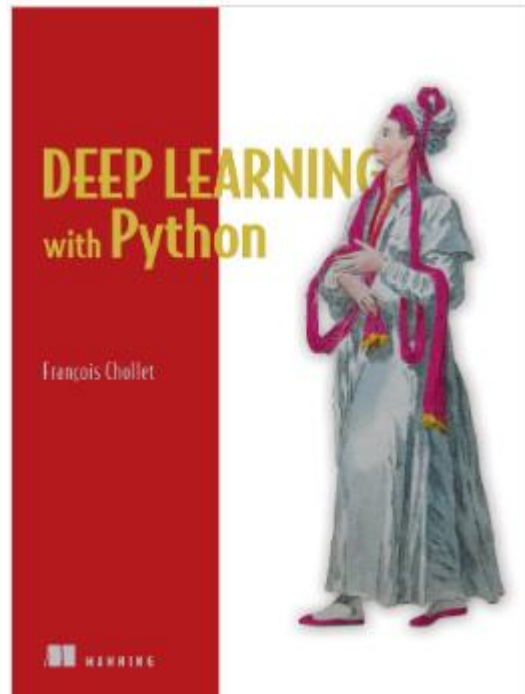
Mais detalhes sobre YOLO e implementação:

<https://arxiv.org/abs/2004.10934>

<https://github.com/Tianxiaomo/pytorch-YOLOv4>

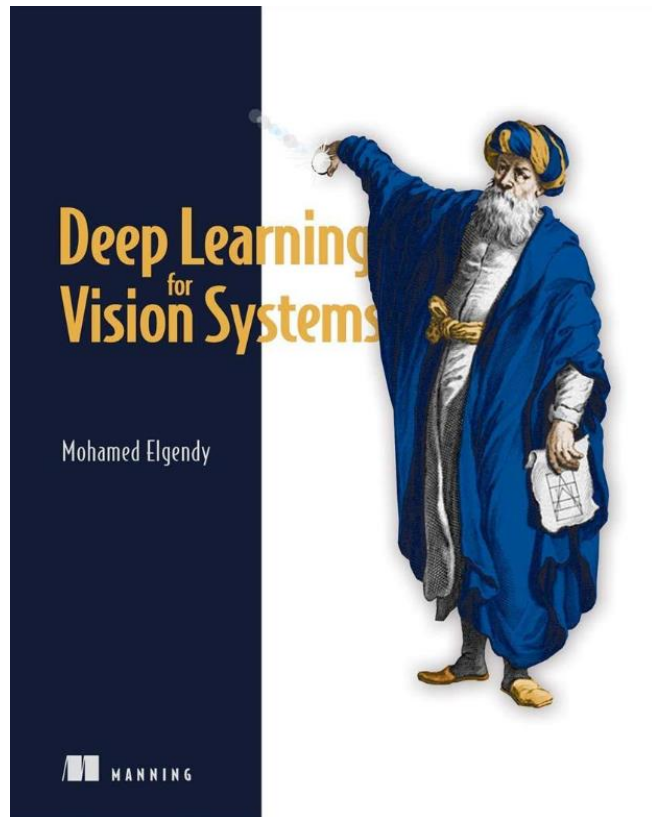
Ref.: Mohamed Elgendy. **Deep Learning for Vision Systems**. 2020

Principais Referências



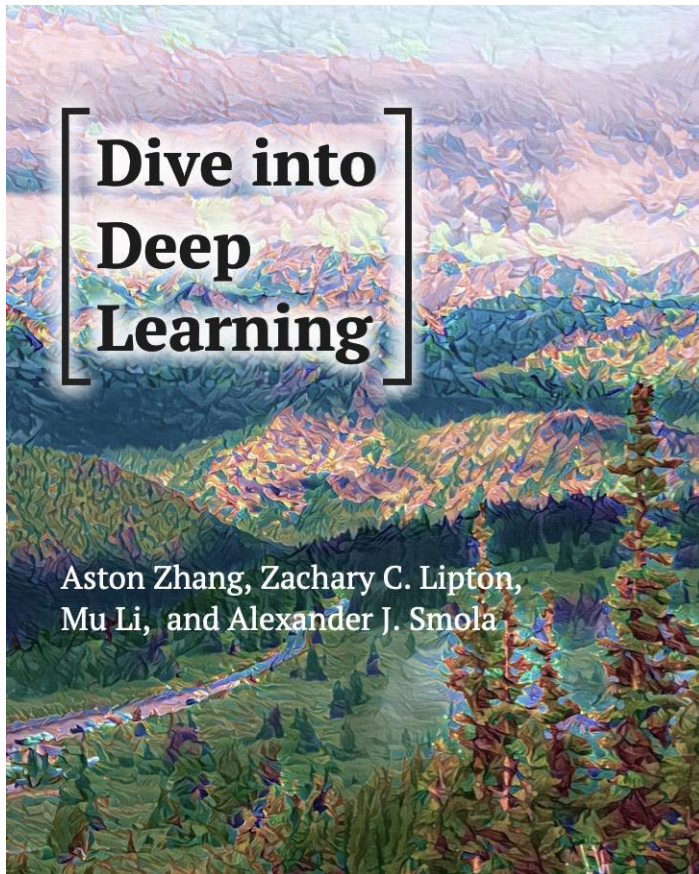
François Chollet. **Deep Learning with Python.** November 2017 ISBN 9781617294433 384 pages

Principais Referências



Mohamed Elgandy. **Deep Learning for Vision Systems**. 2020
(estimated)
ISBN 9781617296192 410 pages

Principais Referências



Aston Zhang; Zack C. Lipton; Mu Li;
Alex J. Smola. **Dive into Deep
Learning**. <http://numpy.d2l.ai/>