Aula 08 – Detecção e reconhecimento de objetos

Prof. João Fernando Mari

joaofmari.github.io

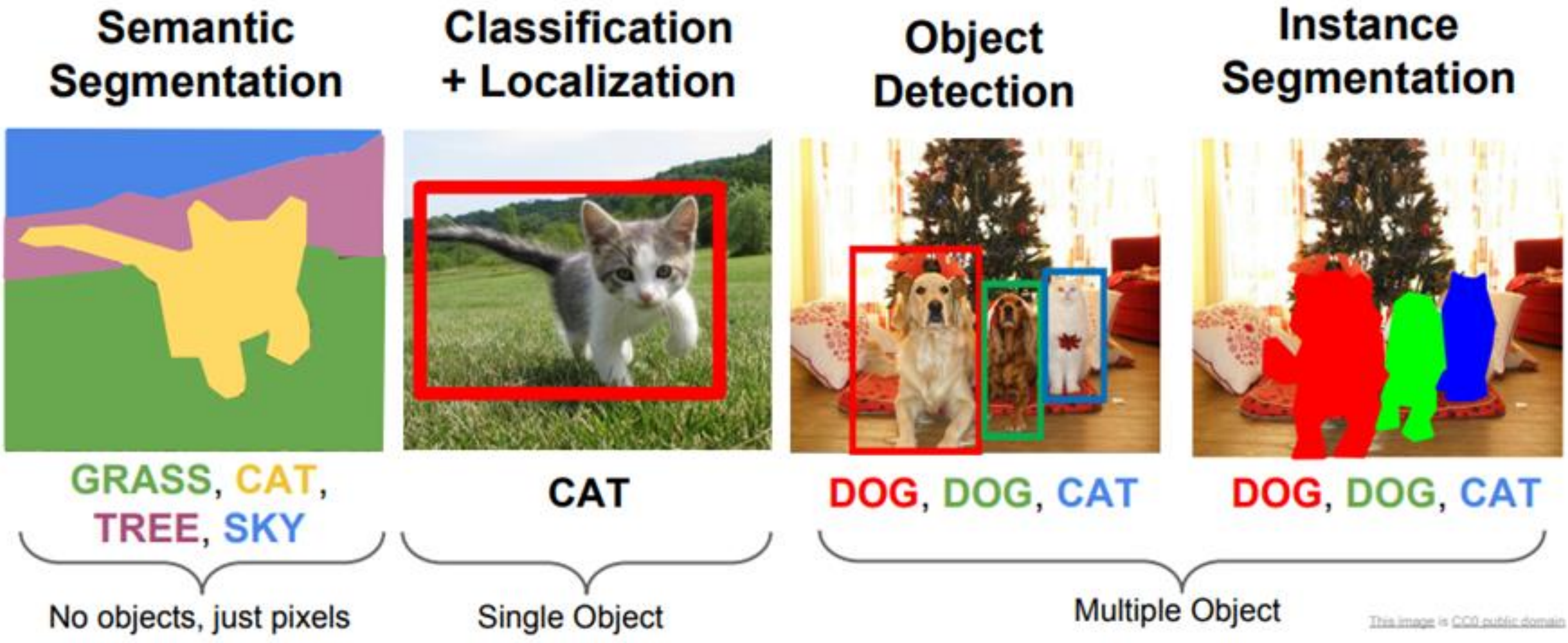
joaof.mari@ufv.br

Roteiro

Universidade Federal de Viçosa

- Localização, detecção e segmentação
- Família R-CNN
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
- Família YOLO

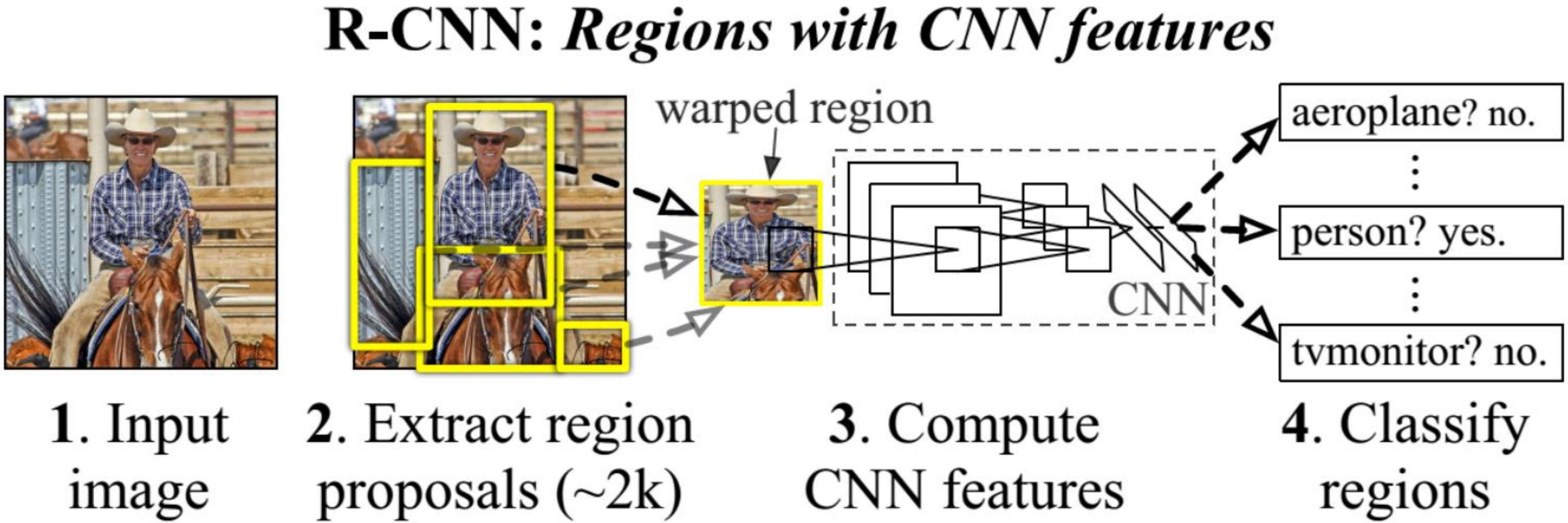
Localização, detecção e segmentação



Stanford cs231n (2022): http://cs231n.stanford.edu/slides/2022/lecture_9_jiajun.pdf

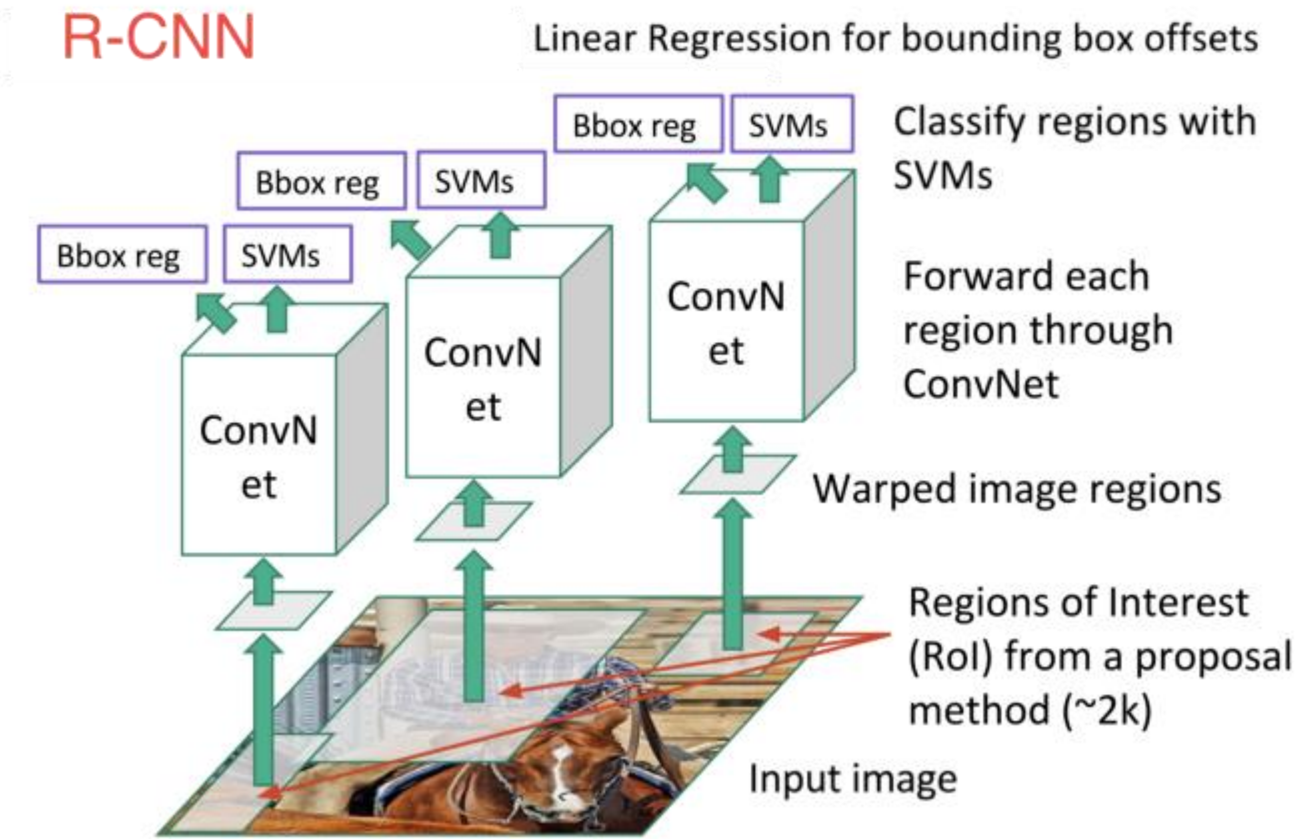
R-CNN

R-CNN



Ross Girshick, et al. “Rich feature hierarchies for accurate object detection and semantic segmentation.”, 2014.

R-CNN



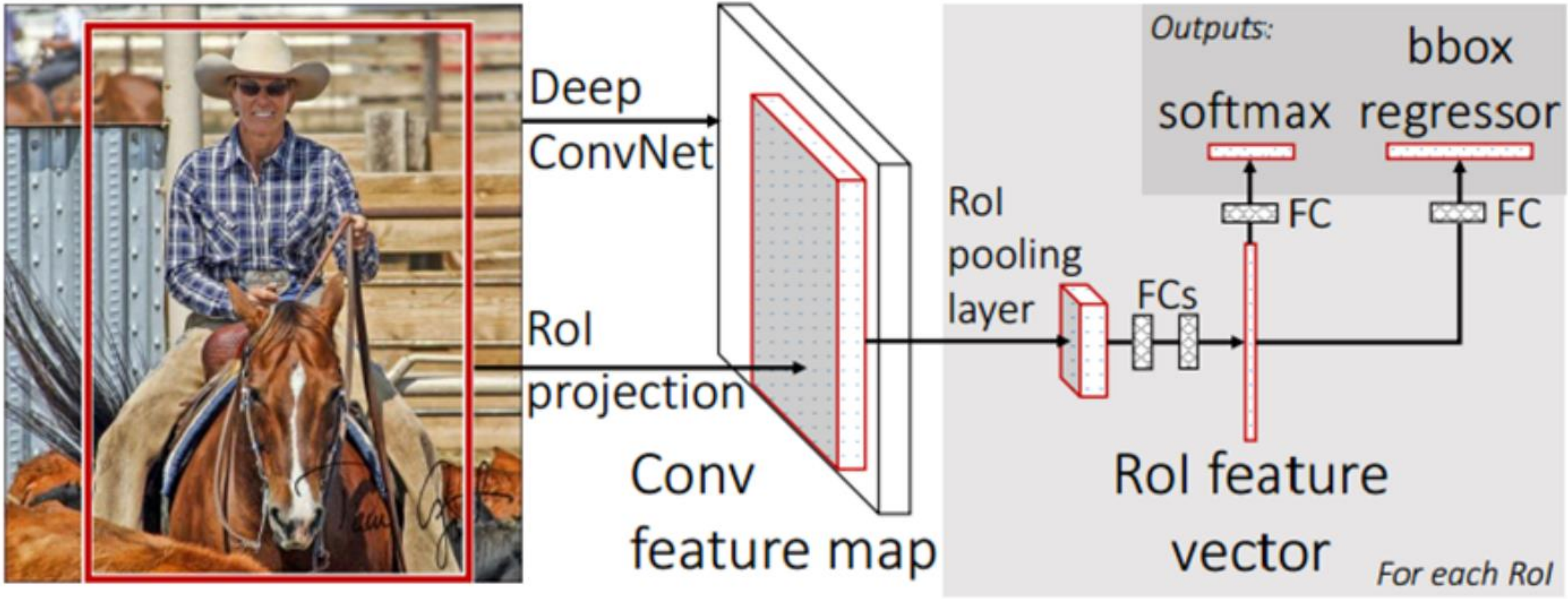
Ross Girshick. Fast R-CNN. Facebook AI Research (FAIR).

- **R-CNN – Region-based Convolutional Neural Network**

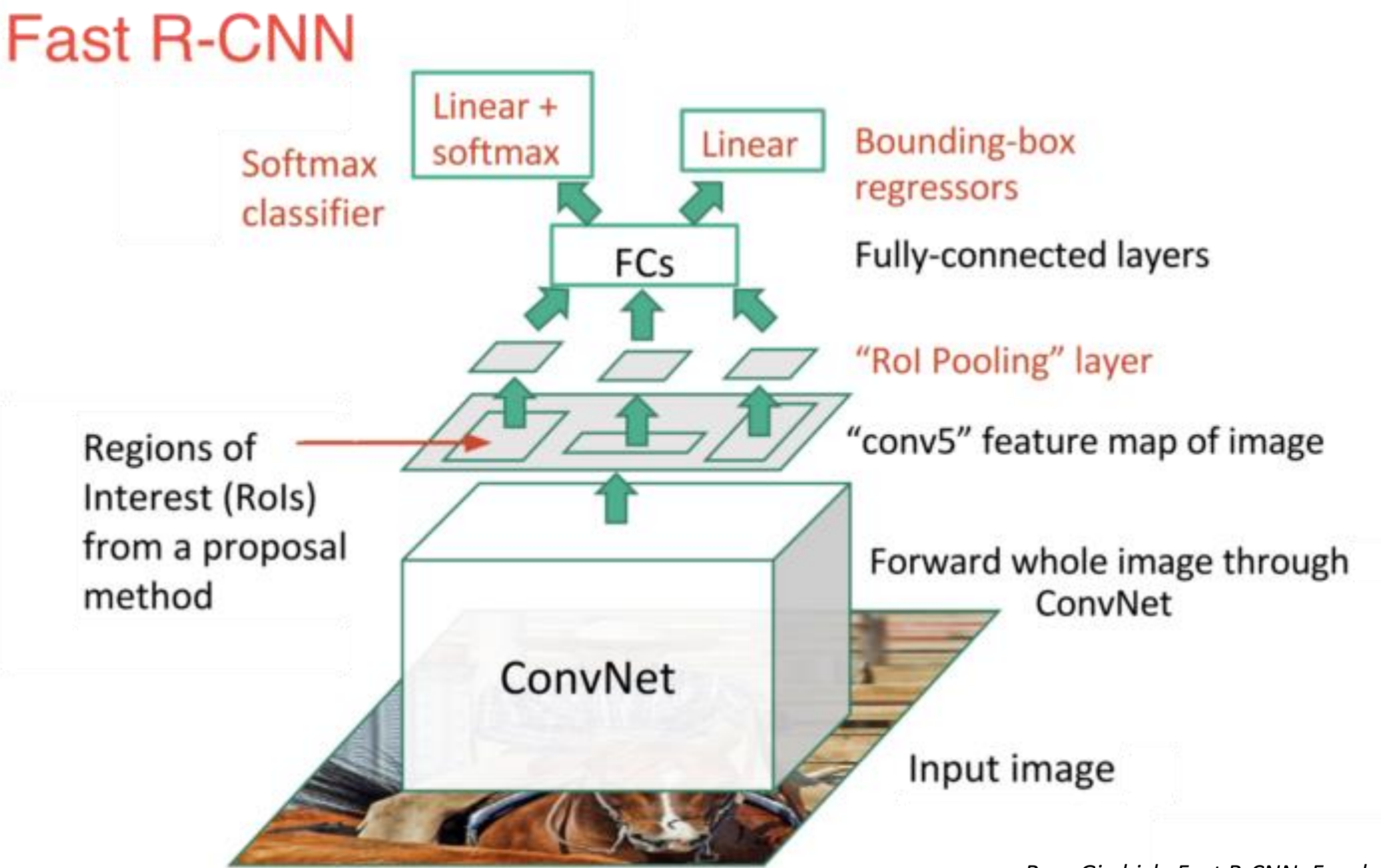
- Varre a imagem buscando possíveis objetos
  - Selective Search
  - Gera ~2000 propostas de regiões
- Passa as propostas de regiões por uma CNN
  - Antes, redimensiona as regiões para o tamanho da camada de entrada da CNN.
- Baseado na saída da CNN:
  - Classifica a região com um classificador SVM
  - Aplica regressão linear para localizar o *bounding-box* do objeto.

FAST R-CNN

Fast R-CNN



Ross Girshick. "Fast R-CNN.", 2015.



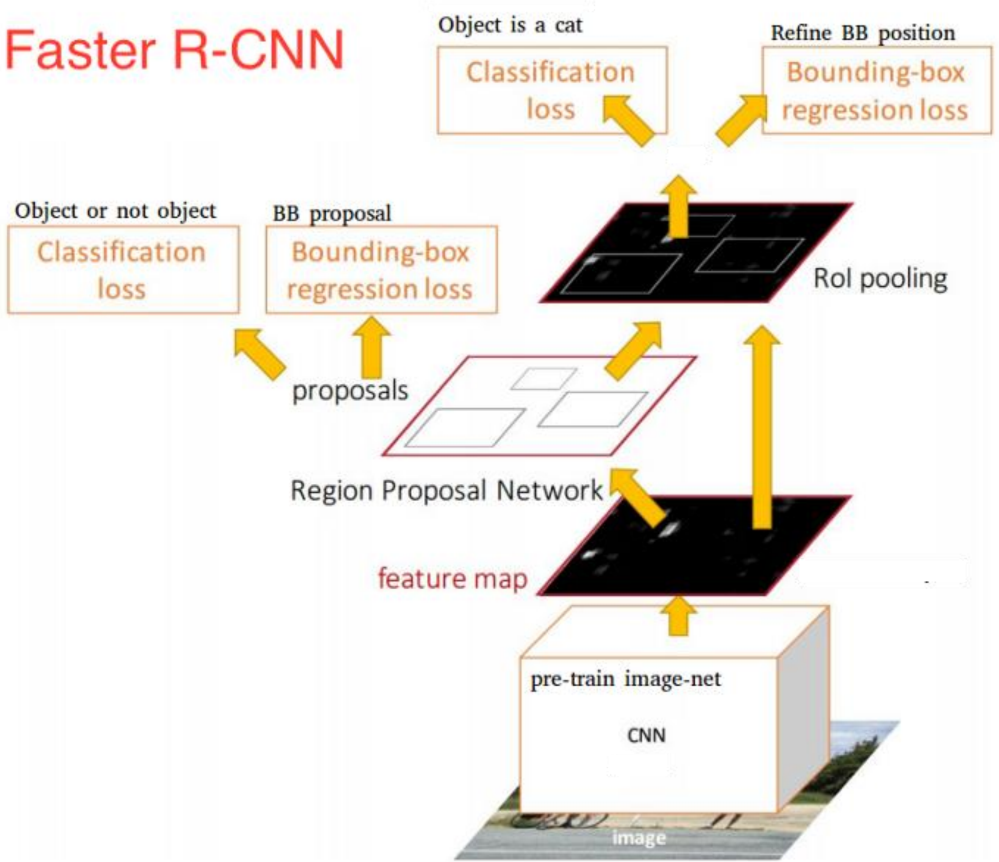
Ross Girshick. Fast R-CNN. Facebook AI Research (FAIR).

Fast R-CNN

- A imagem de entrada passa por uma CNN
  - Um modelo pré-treinado da VGG-16 é usado para extração de características.
- No final da CNN é incluída uma camada chamada de Region of Interest Pooling Layer (RoI Pooling)
  - RoIs obtidas a partir da imagem de entrada
- A saída da CNN é interpretada por uma camada completamente conectada.
- Na sequencia, o modelo bifurca em duas saídas:
  - Uma para a predição de classe via camada softmax
  - Outra com uma saída linear para o bounding-box.
- O processo é repetido múltiplas vezes para cada região de interesse em uma imagem.

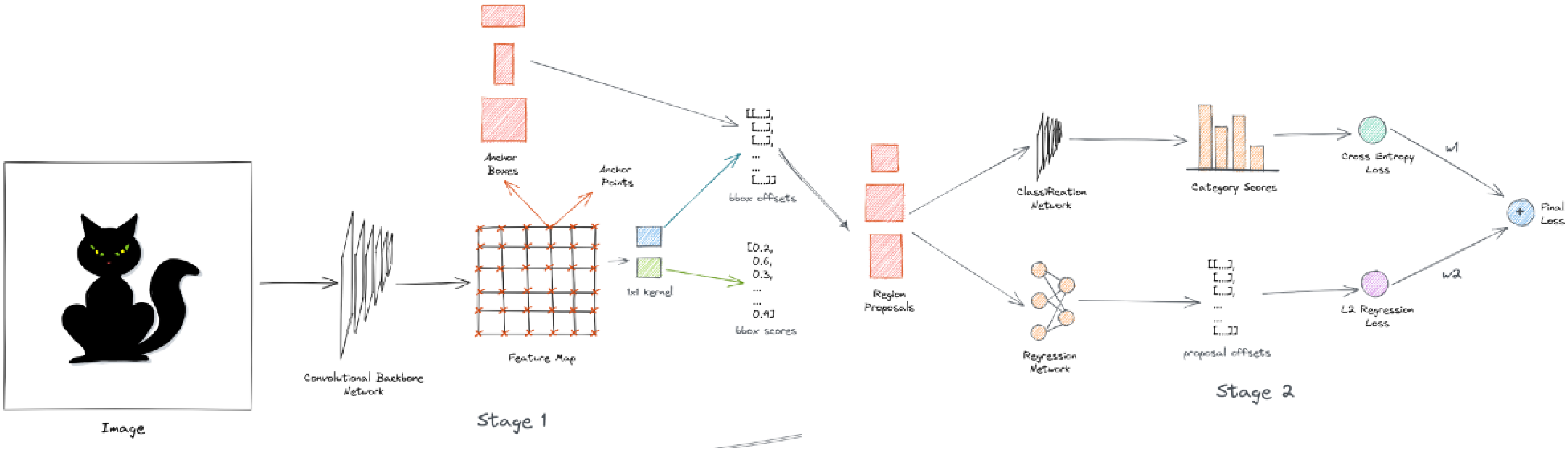
FASTER R-CNN

Faster R-CNN



Ren, et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.", 2016.

Faster R-CNN



https://towardsdatascience.com/understanding-and-implementing-faster-r-cnn-a-step-by-step-guide-11acfff216b0

Faster R-CNN

- **Etapa 1: Rede de propostas de regiões (*region proposal network*)**
  - A imagem toda passa por uma CNN pré-treinada (ResNet, VGG-16, ...)
    - A saída é um mapa de características
    - Cada ponto do mapa de características é tratado como uma âncora.
      - Para cada âncora, são geradas múltiplas caixas com tamanhos diferentes
    - Uma CNN com kernels 1x1 é usada para predizer a categoria e offset das caixas
      - Comparando as caixas com o *ground-truth* (durante o treinamento)
        - » Discriminadas caixas positivas (objetos) e negativas (fundo)
        - » Entropia cruzada binária com função de perda
      - Uma CNN com kernel 1x1 aprende os *bounding-boxes* dos objetos
        - » Regressão com função de perda L2 (distância Euclidiana)
    - As caixas são transformadas de acordo com os *bounding-boxes*
      - Propostas de região.

Faster R-CNN

- **Etapa 2:**
  - Uma CNN simples classifica as propostas de região em múltiplas classes
    - Entropia cruzada
    - *RoI pooling* é usado para redimensionar as propostas de região para o tamanho da entrada de rede.
  - Outra rede usa regressão para alinhar os *bounding-boxes*
    - Função de perda L2
  - A função de perda final combina entropia cruzada com L2

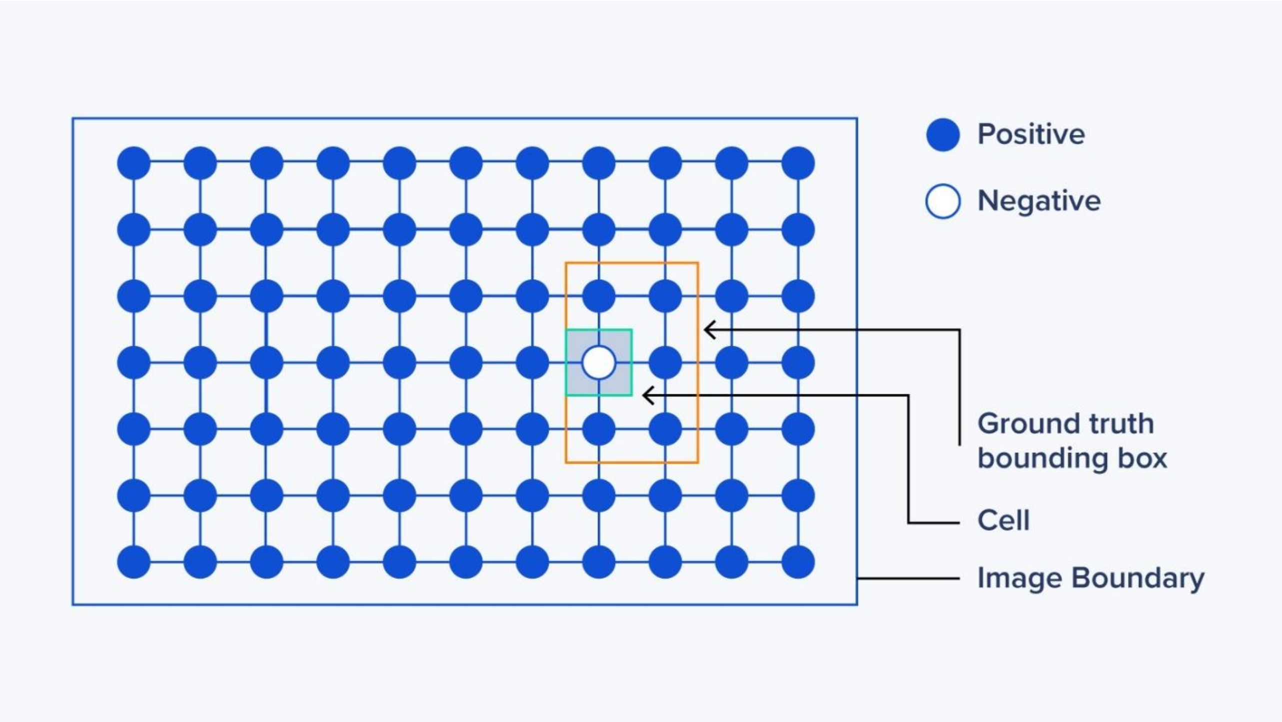
YOLO

YOLO family

- **YOLO - You Only Look Once**

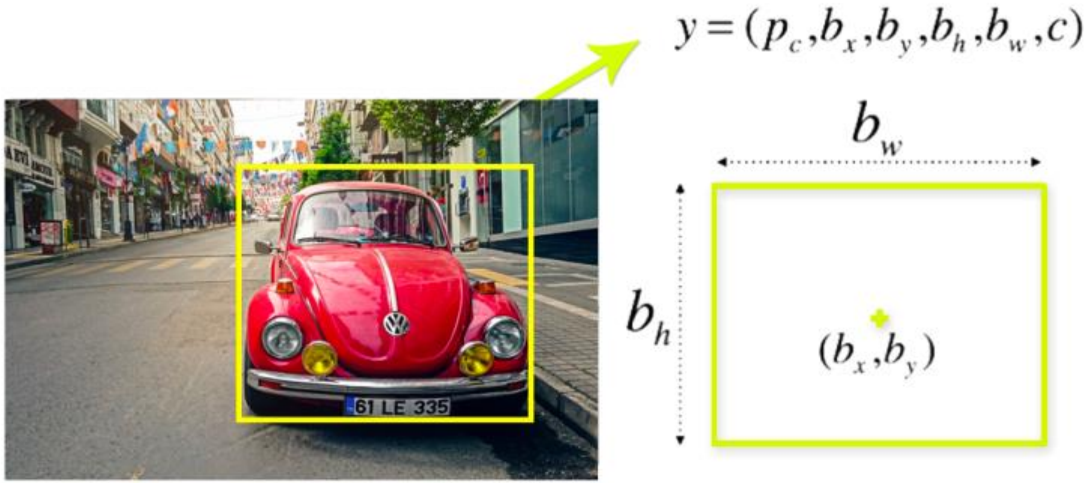
- YOLO (2015)
- YOLOv2 (2017)
- YOLOv3 (2018)
- YOLOv4 (2020)
- YOLOv5 (2020)
- YOLOv6 (2022)
- YOLOv7 (2022)
- ...

- Etapa 1: Blocos residuais
  - Dividir a imagem em pequenas caixas organizadas em grade



https://www.exxactcorp.com/blog/Deep-Learning/YOLOv5-PyTorch-Tutorial

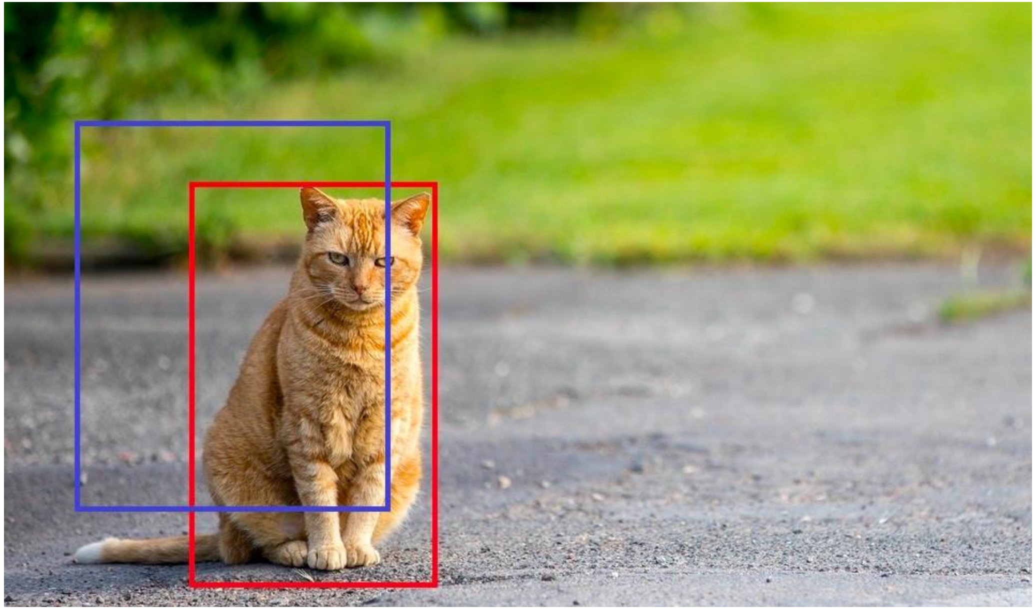
- Etapa 2: Regressão do bounding-box
  - Identifica os objetos dentro de um bounding-box
  - Após detectar um objeto, desenha um bounding-box ao seu redor
    - center point, height, width, and class



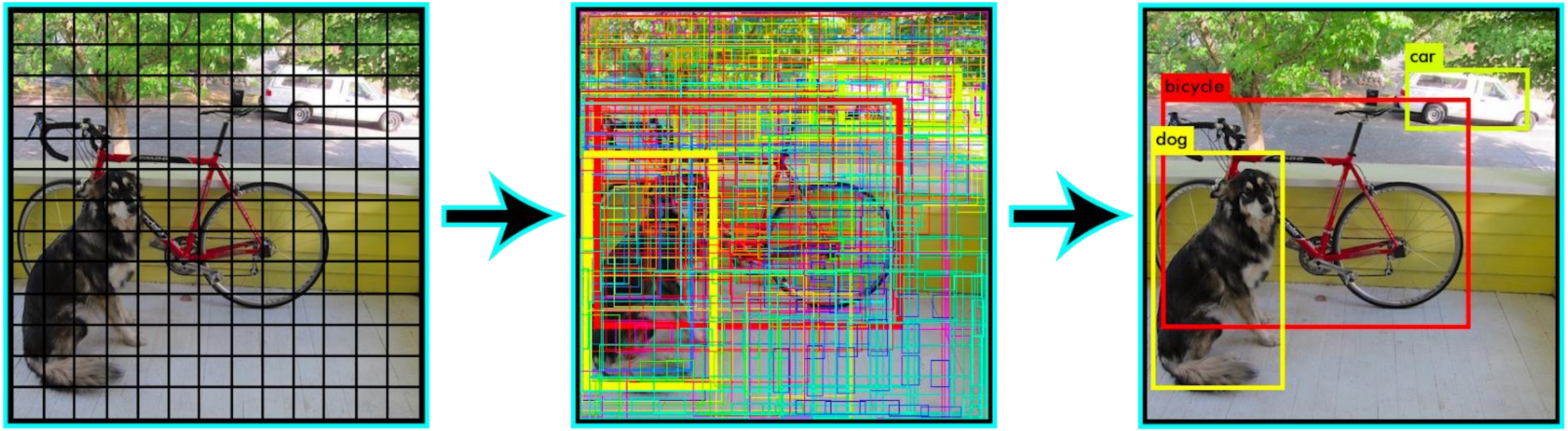
https://www.exxactcorp.com/blog/Deep-Learning/YOLOv5-PyTorch-Tutorial

YOLO

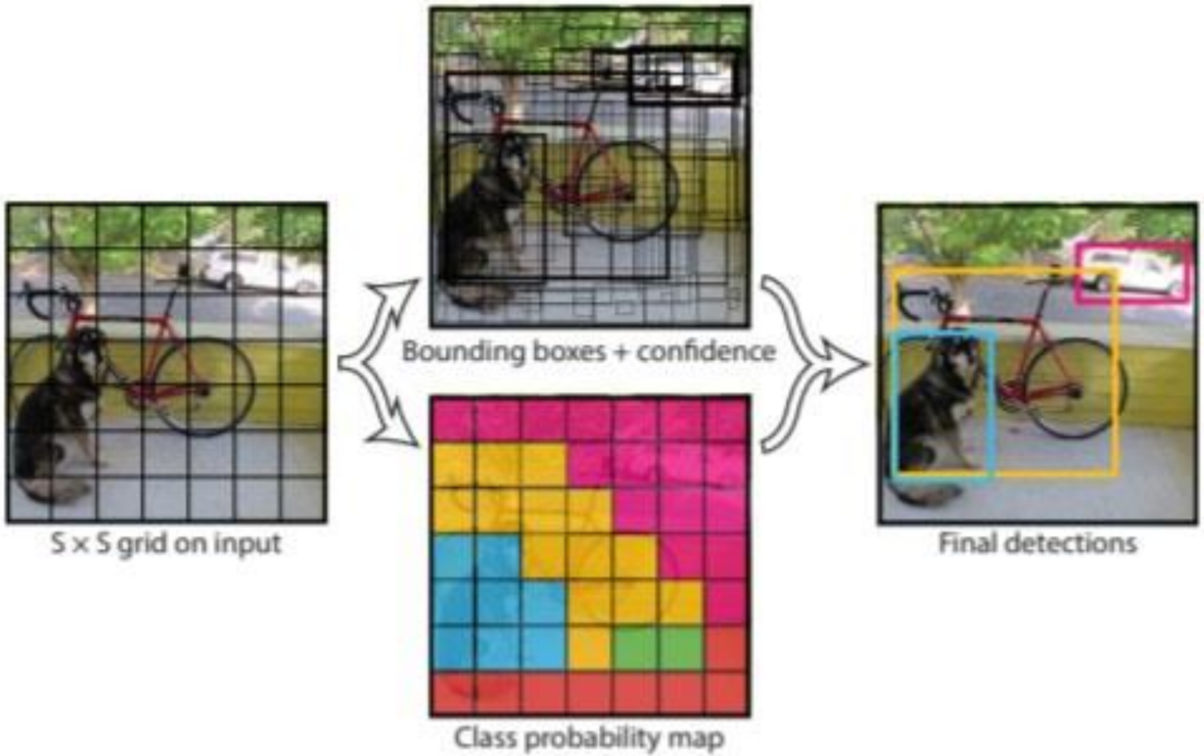
- Etapa 3: Intersecção sobre União (IoU – *Intersection over Union*)
  - Usado para calcular a acurácia do modelo



https://www.exxactcorp.com/blog/Deep-Learning/YOLOv5-PyTorch-Tutorial



https://www.exxactcorp.com/blog/Deep-Learning/YOLOv5-PyTorch-Tutorial



https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/

UFV
Universidade Federal de Viçosa

DATASETS

Datasets

- COCO DATASET
  - https://cocodataset.org/
  - Detecção de objetos
  - Segmentação
  - 330 K imagens
  - >200K imagens rotuladas
  - 1.5 milhões de instâncias de objetos
  - 80 classes de objetos
  - ...

Datasets

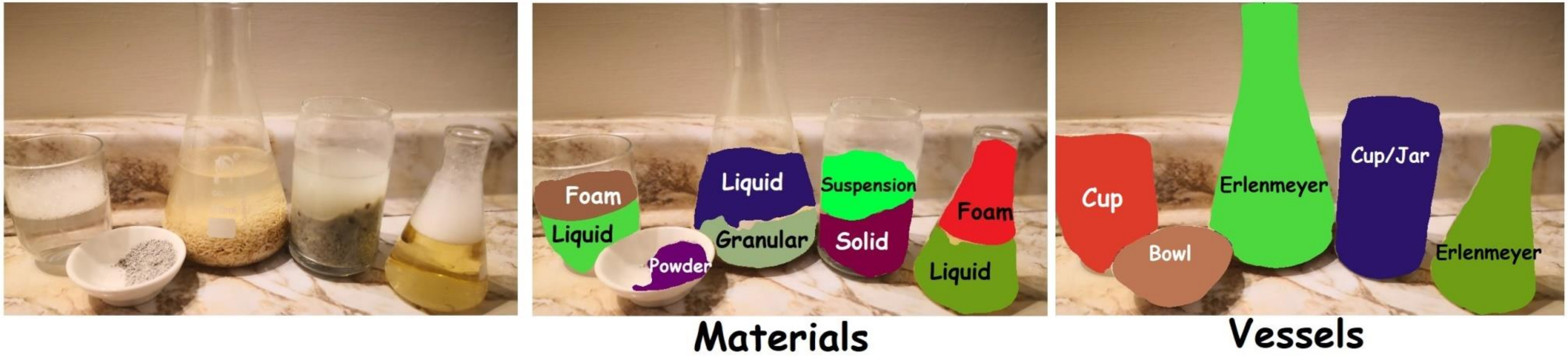
- PASCAL VOC 2012 DATASET
  - https://www.kaggle.com/datasets/gopalbhattrai/pascal-voc-2012-dataset
  - http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

20 classes

Datasets

UFV
Universidade Federal de Viçosa

- The Vector-LabPics dataset focus
  - https://www.cs.toronto.edu/chemselfies/



Bibliografia

- Ross Girshick, et al. “Rich feature hierarchies for accurate object detection and semantic segmentation.”, 2014.
  - https://arxiv.org/abs/1311.2524
- Ross Girshick. “Fast R-CNN.”, 2015.
  - https://arxiv.org/abs/1504.08083
- Shaoqing Ren, et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.”, 2016.
  - https://arxiv.org/abs/1506.01497
- Ross Girshick. Fast R-CNN. Facebook AI Research (FAIR). Work done at Microsoft Research.
  - https://dl.dropboxusercontent.com/s/vlyrkgd8nz8gy5l/fast-rcnn.pdf?dl=0
- Joseph Redmon, et al. “You Only Look Once: Unified, Real-Time Object Detection.” 2015.
  - https://arxiv.org/abs/1506.02640
- CS231n: Deep Learning for Computer Vision. Stanford - Spring 2022
  - http://cs231n.stanford.edu/

Bibliografia

- https://debuggercafe.com/a-simple-pipeline-to-train-pytorch-faster-rcnn-object-detection-model/
- https://pyimagesearch.com/2021/08/02/pytorch-object-detection-with-pre-trained-networks/
- https://machinelearningmastery.com/object-recognition-with-deep-learning/
- https://www.jeremyjordan.me/evaluating-image-segmentation-models/
- https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
- https://www.upgrad.com/blog/ultimate-guide-to-object-detection-using-deep-learning/
- https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9
- https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html
- https://pyimagesearch.com/2021/08/02/pytorch-object-detection-with-pre-trained-networks/
- https://github.com/chenyuntc/simple-faster-rcnn-pytorch
- https://blog.paperspace.com/faster-r-cnn-explained-object-detection/
- https://deepsense.ai/region-of-interest-pooling-explained/
- https://medium.com/codex/implementing-r-cnn-object-detection-on-voc2012-with-pytorch-b05d3c623afe
- https://towardsdatascience.com/understanding-and-implementing-faster-r-cnn-a-step-by-step-guide-11acfff216b0

Bibliografia

- https://www.exxactcorp.com/blog/Deep-Learning/YOLOv5-PyTorch-Tutorial
- https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection
- https://curiousily.com/posts/object-detection-on-custom-dataset-with-yolo-v5-using-pytorch-and-python/
- https://pyimagesearch.com/2022/04/04/introduction-to-the-yolo-family/
- https://towardsdatascience.com/yolov7-a-deep-dive-into-the-current-state-of-the-art-for-object-detection-ce3ffedeeaeb
- https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/
- https://www.section.io/engineering-education/object-detection-with-yolov5-and-pytorch/
- https://pytorch.org/hub/ultralytics_yolov5/

FIM