Object Detection

Through Machine Learning

Uros Stegic
uros.stegic@everseen.com

Classification experiment

Task Description

> ***Object detection*** *is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos.*
>
> *Wikipedia*

Visualizing the Task

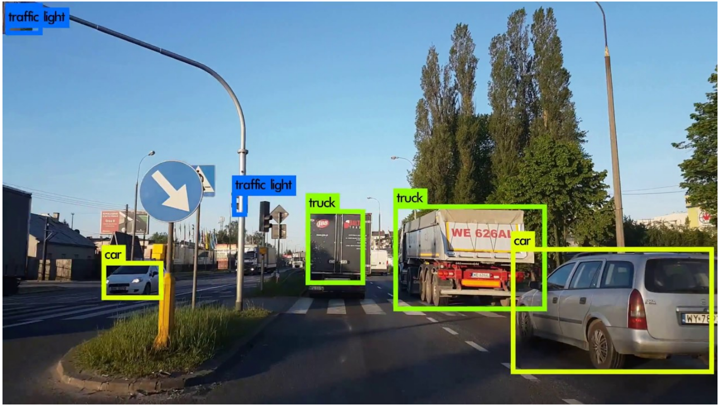


Figure: Object Detection Task

Building a metric for bounding box predictions

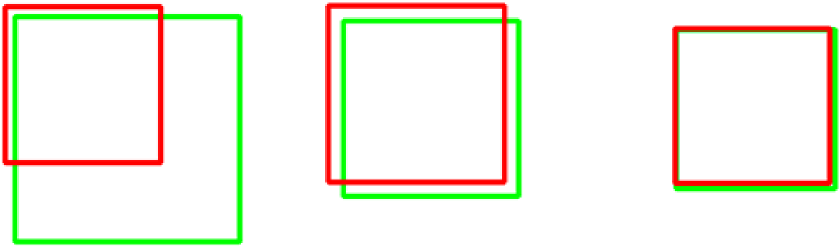


Figure: Bounding Box Missmatch

Defining the IoU

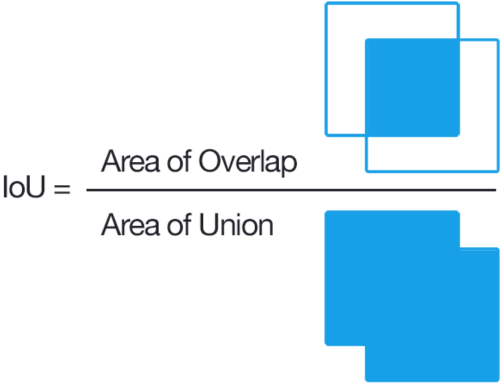


Figure: Intersection over Union

IoU - Sanity check

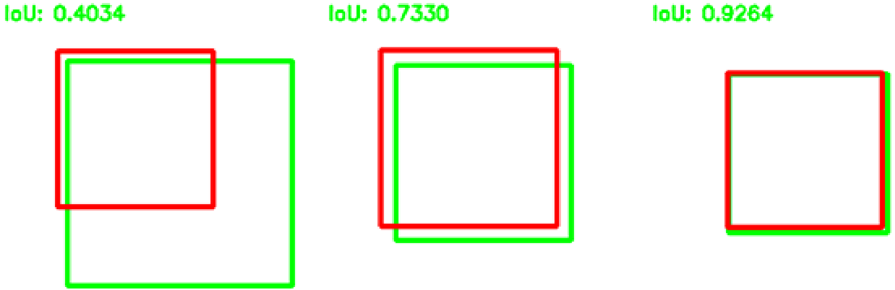


Figure: Intersection over Union - Example

Removing Redundancy

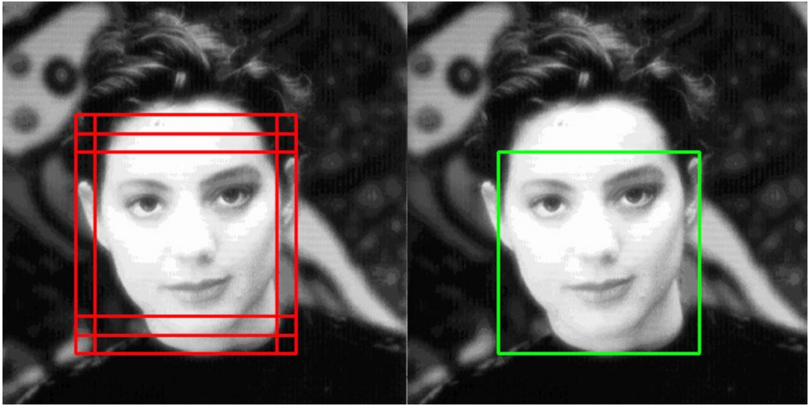


Figure: Elimination of Multiple Bounding Boxes

Non-Maximum Suppression

- Threshold every bounding box
- Sort bounding boxes by detection probability in decresing order
- For each bounding box b_i remove all bounding boxes $b_j (j \neq i)$ such that $IoU(b_i, b_j) \geq t$ for some fixed t

YOLO

You Only Look Once [RDGF15]

YOLO - Introduction

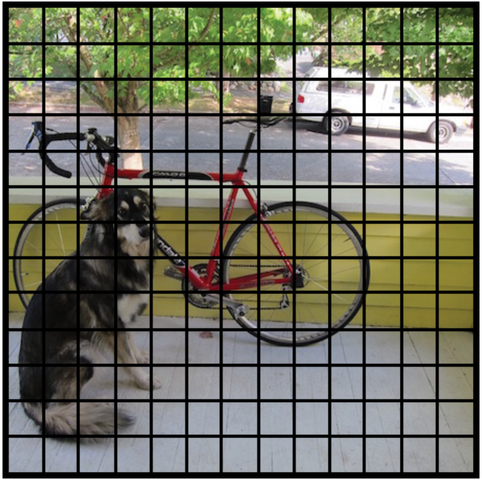


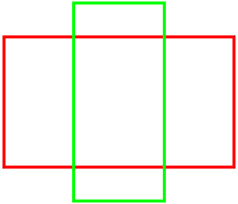
Figure: Grid for YOLO

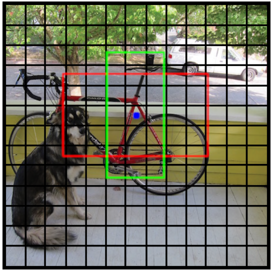
$$\hat{y} = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix}$$

Anchor Boxes

- Choose a number of anchors
- Modify the output to include this anchors
- ...
- Profit

Anchor Boxes - Example





$$\hat{y}_1 = \begin{bmatrix} p_{c_1} \\ b_{x_1} \\ b_{y_1} \\ b_{w_1} \\ b_{h_1} \end{bmatrix}, \quad \hat{y}_2 = \begin{bmatrix} p_{c_2} \\ b_{x_2} \\ b_{y_2} \\ b_{w_2} \\ b_{h_2} \end{bmatrix}, \quad \hat{c} = \begin{bmatrix} c_1 \\ \dots \\ c_n \end{bmatrix}, \quad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{c} \end{bmatrix}$$

YOLO - Summary

- Convolutions: 24
- Trainable parameters: $51m$
- Input shape: 448×448
- Output shape: $G \times G \times (5A + C)$
- Output shape (from paper): $7 \times 7 \times (5 * 3 + 20)$

YOLO - Loss Function

$$\mathcal{L}(y, \hat{y}) = \lambda_{coord} L_{loc}(y, \hat{y}) + \lambda_{coord} L_{dim}(y, \hat{y}) + L_{obj}(y, \hat{y}) + \lambda_{noobj} L_{noobj}(y, \hat{y}) + L_{class}(y, \hat{y})$$

$$\mathrm{L}_{loc}(y, \hat{y}) = \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$

$$\mathrm{L}_{dim}(y, \hat{y}) = \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

$$\mathrm{L}_{obj}(y, \hat{y}) = \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_ij)^2$$

$$\mathrm{L}_{noobj}(y, \hat{y}) = \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_ij)^2$$

$$\mathrm{L}_{class}(y, \hat{y}) = \sum_{i=0}^{s^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

Region Based Approach

Two-Stage Detectors

General Idea

- Propose Regions of Interest (RoI)
- Classify each RoI
- Refine Bounding Box Coordinates around each RoI

R-CNN Family

- Regions with CNN (R-CNN) [GDDM13]
- Fast R-CNN [Gir15]
- Faster R-CNN [RHGS15]
- Mask R-CNN [HGDG17]

Region Proposals - Selective Search

Figure: Selective Search Algorithm Visualized

R-CNN

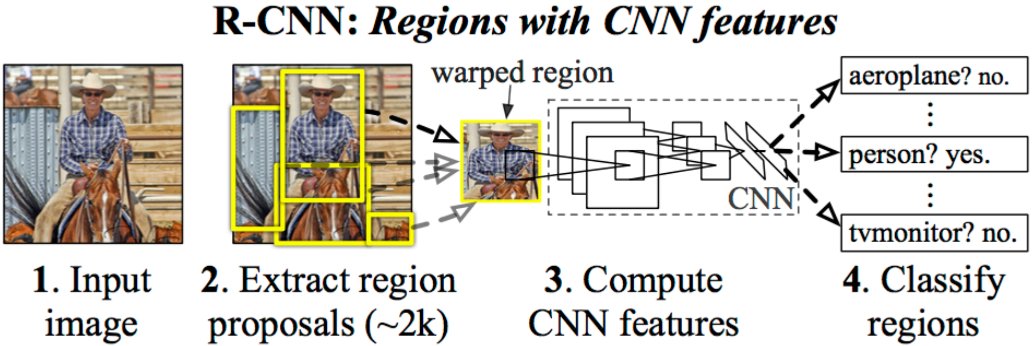


Figure: R-CNN Pipeline

Fast R-CNN

- Convolution Based Sliding Window
- RoI Pooling
- Softmax Classification

Sliding Window - CNN Way

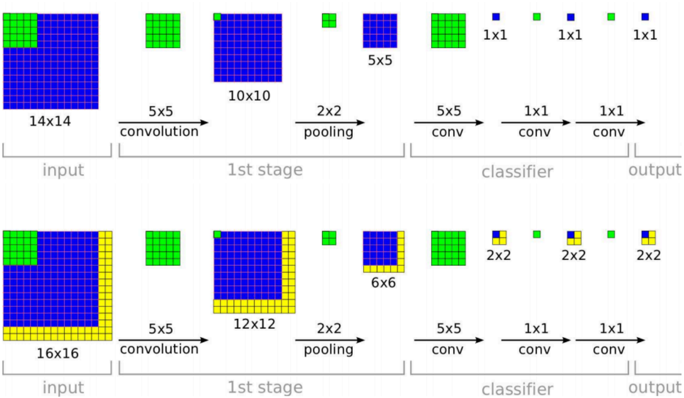


Figure: Sliding Window - CNN Implementation

Fast R-CNN - Visualized

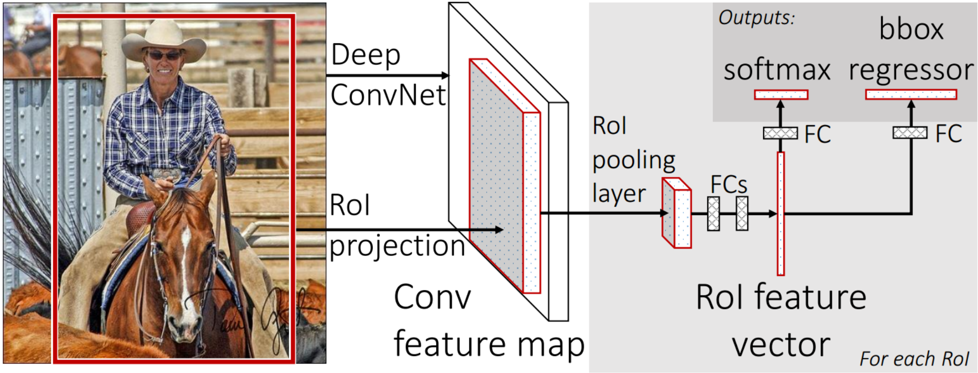


Figure: Fast R-CNN Pipeline

Fast R-CNN - Loss

$$\mathcal{L}(p, u, t^u, v) = L_{class}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

$$\mathrm{L}_{class}(p, u) = -\log p_u$$

$$\mathrm{L}_{loc}(t^u, v) = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i^u - v_i)$$

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } x \leq 1 \\ x - 0.5, & \text{otherwise} \end{cases}$$

Faster R-CNN

- Bottleneck: Region Proposals by Selective Search (2s)
- Solution: Region Proposals by CNN (0.01s)

Region Proposal Network

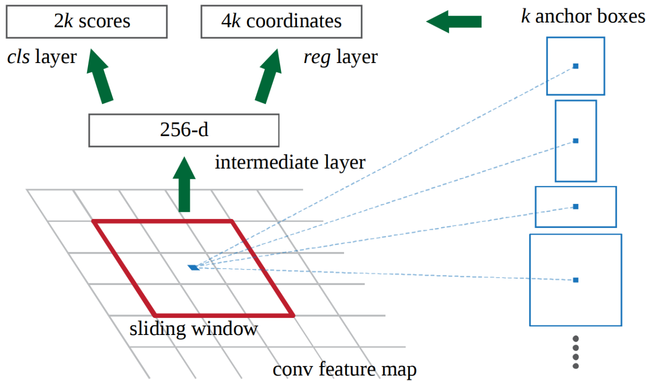


Figure: Region Proposal Network for Faster R-CNN

RPN - Loss

$$\mathcal{L}(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

Faster R-CNN - Architecture

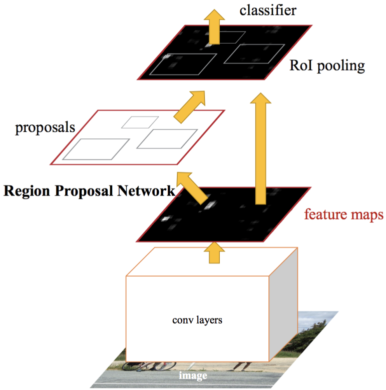


Figure: Model Scheme of Faster R-CNN

Mask R-CNN

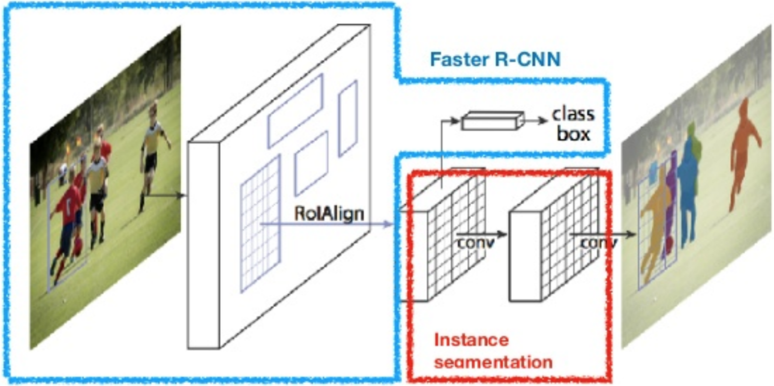


Figure: Model Scheme of Faster R-CNN

Other Influential Models

- RetinaNet (Focal Loss) [LGG+17]
- Single Shot Detector [LAE+15]
- R-FCN [DLHS16]

Practicalities

Speed vs. Precision

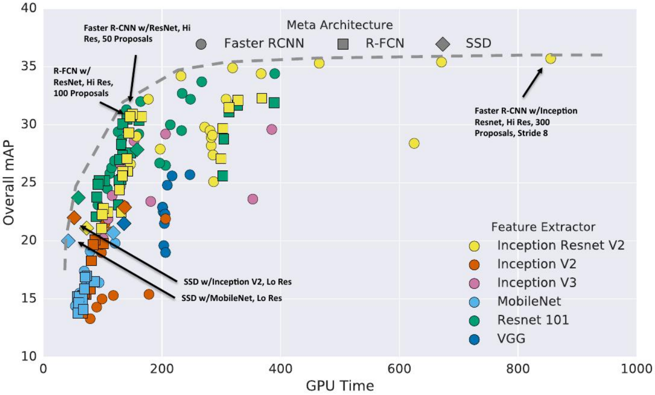


Figure: GPU Time vs. Precision [HRS+16]

Framework

- Tensorflow Object Detection API
- Pytorch Detectron 2

Lecture Pronouncement

CONVERGENCE

References I

- Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, R-FCN: object detection via region-based fully convolutional networks, CoRR **abs/1605.06409** (2016).
- Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CoRR **abs/1311.2524** (2013).
- Ross B. Girshick, Fast R-CNN, CoRR **abs/1504.08083** (2015).
- Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick, Mask R-CNN, CoRR **abs/1703.06870** (2017).
- Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy, Speed/accuracy trade-offs for modern convolutional object detectors, CoRR **abs/1611.10012** (2016).

References II

- Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg, SSD: single shot multibox detector, CoRR **abs/1512.02325** (2015).
- Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár, Focal loss for dense object detection, CoRR **abs/1708.02002** (2017).
- Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi, You only look once: Unified, real-time object detection, CoRR **abs/1506.02640** (2015).
- Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun, Faster R-CNN: towards real-time object detection with region proposal networks, CoRR **abs/1506.01497** (2015).