CV / VLM

Unit 2: Introduction to Object Detection (OD)

TIL-AI
TODAY I LEARNED-AI

2.2.2

Region-Based Object Detection

Selection Search and Region Proposals

Why Not Sliding Window?

> Computational Efficiency Challenge:
Using the sliding window algorithm, the computational cost of convolution operations can be significant for large images.

Methods of improving efficiency of object detection:

1. Selective Search
2. Region Proposal Networks (RPNs)

Region-Based Object Detection
Selective Search

Divide, search and conquer

- Sliding window is slow because it does multiple OD passes over the same object
- Solution: identify objects first before passing them to the object detection network

Pros

Capture all scales, faster to compute, Diversification

Labeled Image

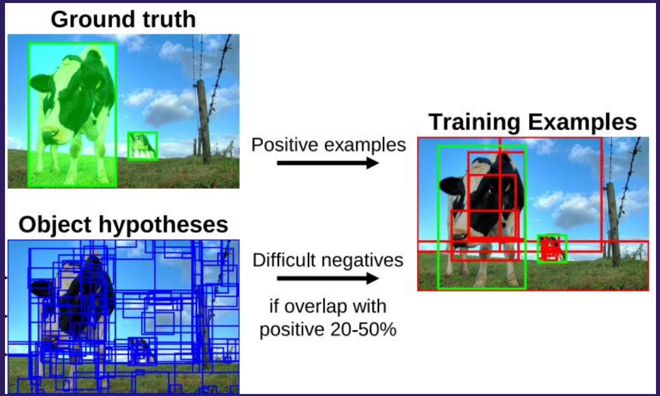


Image with hypotheses on where the object could potentially be
(algorithm driven)

Paper on [Selective Search]

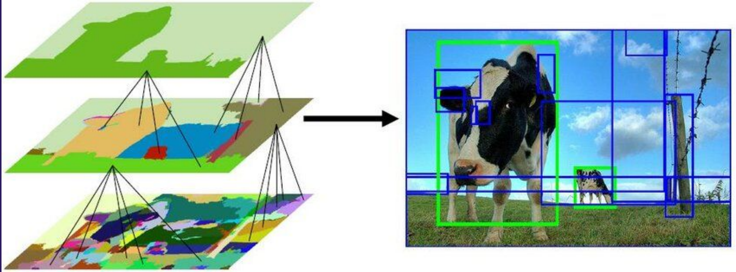
Region-Based Object Detection
Selective Search

Object Hypotheses
Computation Steps

1. Compute similarity between regions of the image (color, texture, size, shape, meta)

2. Combine similar image regions

3. Repeat with the combined regions



Regions of image being Iteratively combined.

Raw Image

Note that this iterative approach where we start with small image segments and iteratively merge them is also known as a **bottom up method.**

Paper on Selective Search
Mining Web Images for Concept Learning (researchgate.net)

Region-Based Object Detection
Region Proposal Network (RPN)

Specialized CNN to pick up Regions

CNN network that is trained to take a feature map as a input and output region proposals (object bounding boxes and detection scores)

One key optimization: anchor boxes

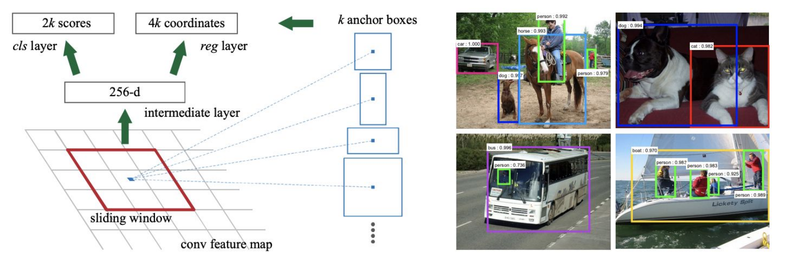
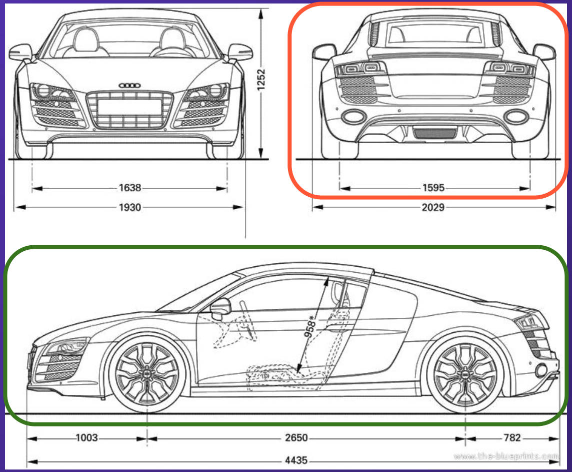


Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

Adopted from RPN Explained | Papers With Code

Region-Based Object Detection
Anchor Boxes

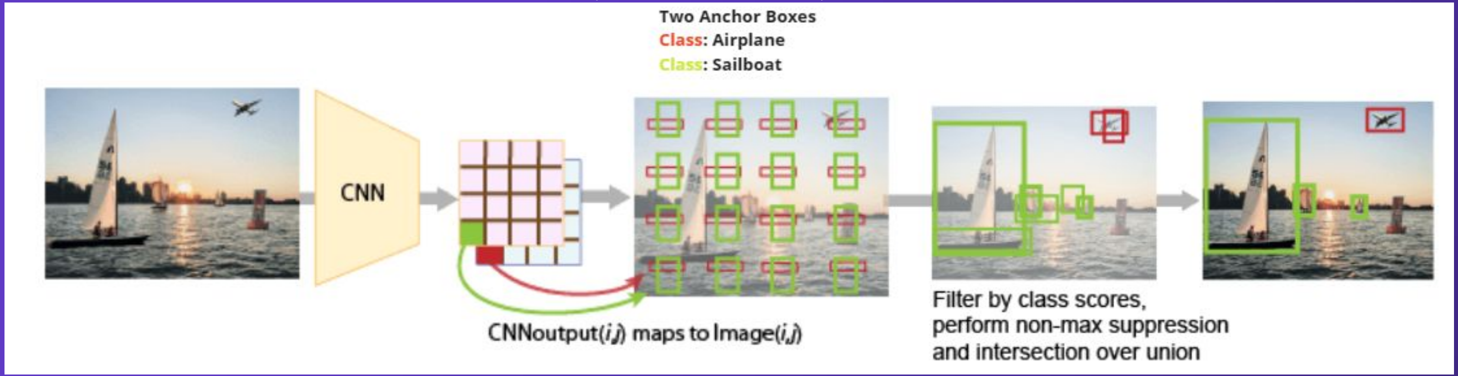
- Also known as prior boxes
- Used in Faster R-CNN, SSD, YOLO v3+
- Pre-defined bounding boxes of various sizes and aspect ratios used to help identify objects in an image
- Constructed from bounding box training data and selected through K-means clustering
- Serve as representations of the ideal location, shape, size of an object class



(Top) a vehicle would be 1:1 (square) when looking from the front or rear, but 2:1 (rectangular) when viewed from the side.

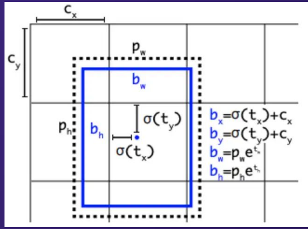
We would need both anchor boxes to be able to consistently detect vehicles

Region-Based Object Detection Anchor boxes (cont.)



(Top) **Process of Anchoring Bounding Box**

- **Anchor Box Generation**: Create thousands of candidate anchor boxes around every grid point.
- **Offset Prediction**: Predict offset from every box as a candidate bounding box (Left).
- **Loss Function Calculation**: Calculate loss function based on the ground truth example.
- **Overlap Probability Calculation**: Calculate the probability that a given offset box overlaps with a real object.
- **Thresholding and Loss Update**: If the overlap probability is greater than 0.5, factor the prediction into the loss function.
- **Model Optimization**: By rewarding and penalizing predicted boxes, slowly pull the model towards only localizing true objects.



Adapted from [Anchor Boxes for Object Detection - MATLAB & Simulink (mathworks.com)]