DLIM Lecture 4: Detection Architectures

J. Chazalon

Session: Fall 2023

EPITA Research & Development Laboratory (LRDE)

Slides generated on October 25, 2023

Today's agenda

Computer Vision Tasks

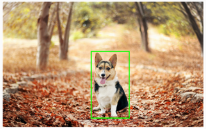
Classification

- Single label for the entire image.
- High-level understanding.
- No object locations.

$\rightarrow$ "DOG"

Localization

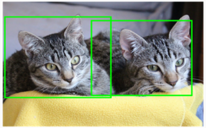
- Object position with bounding box.
- Single object.



$\rightarrow$ "DOG" + bbox

Object Detection

- Identify and locate multiple objects.
- Includes classification and localization.



$\rightarrow$ bboxes (class, coords)

Semantic Segmentation

- Classify each pixel.
- Image divided by classes.

$\rightarrow$ classification map

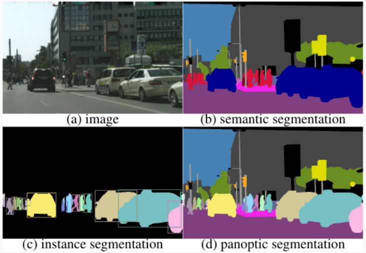
Instance Segmentation

- Unique label for each object instance.
- Pixel-level object masks.

$\rightarrow$ id map

Panoptic Segmentation

- Unifies semantic segmentation and instance segmentation.
- Assigns unique labels to all object instances and stuff classes.
- Provides pixel-level information for both objects and stuff.



$\rightarrow$ classification + id maps

Object Detection Techniques

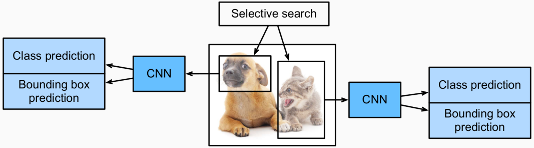
Region-Proposal-Based vs. Single-Stage Detection Networks

- **Region-Proposal-Based Detection Networks:**
  - Two-step process: Region proposal and classification.
  - Greater accuracy, especially in complex scenarios.
  - Flexibility in handling object sizes and shapes.
  - Examples: R-CNN, Fast R-CNN, Faster R-CNN.
- **Single-Stage Detection Networks:**
  - One-step process for object detection.
  - Simplicity and speed.
  - Suited for real-time applications.
  - Examples: YOLO, SSD.

Evolution of Region-Proposal-Based Detection Networks

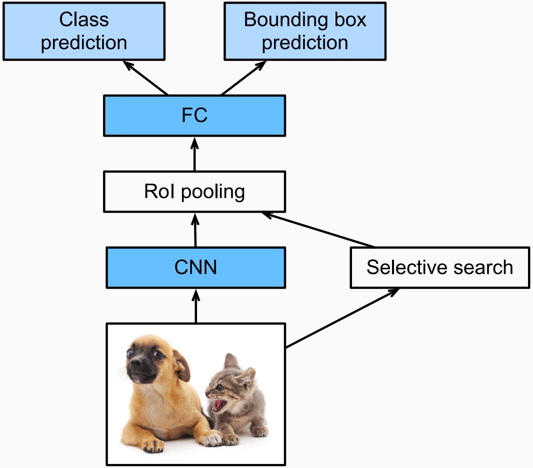
R-CNN (Region-based Convolutional Neural Network)

- **Key Contribution:**
  - Introduced region proposals for object detection.
- **Incremental Improvements:**
  - Used selective search to propose regions.
  - Each region was processed through a pre-trained CNN.
  - Computationally expensive.



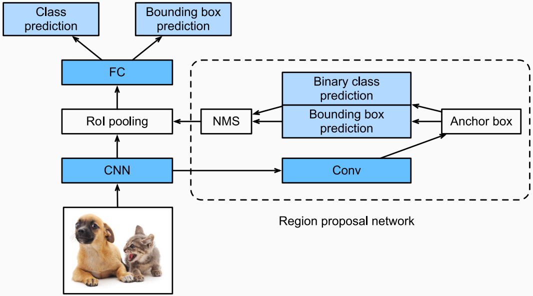
Fast R-CNN

- **Key Contribution:**
  - Unified the region proposal and feature extraction steps.
- **Incremental Improvements:**
  - Introduced RoI (Region of Interest) pooling layer.
  - Combined region proposal and feature extraction.
  - Faster and more efficient.



Faster R-CNN

- **Key Contribution:**
  - Integrated region proposal generation into the neural network.
- **Incremental Improvements:**
  - Introduced the Region Proposal Network (RPN).
  - End-to-end trainable.
  - Achieved faster and more accurate detection.



Views with autograd enabled!

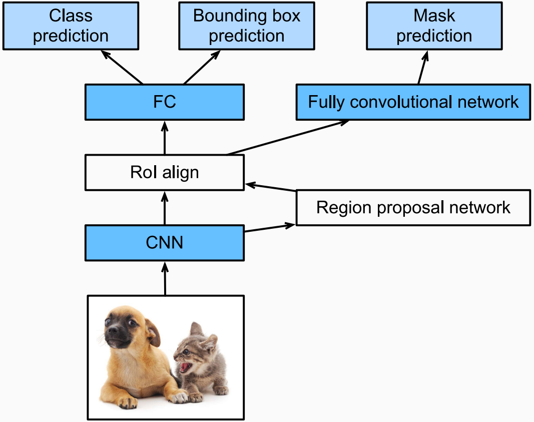
```
import torch
import torchvision

X = torch.arange(16.).reshape(1, 1, 4, 4)
rois = torch.Tensor([[0, 0, 0, 20, 20], [0, 0, 10, 30, 30]])
torchvision.ops.roi_pool(X, rois, output_size=(2, 2), spatial_scale=0.1)

>>> tensor([[[[ 5.,  6.],
              [ 9., 10.]]],
            [[[ 9., 11.],
              [13., 15.]]]])
```

Mask R-CNN

- **Key Contribution:**
  - Extended Faster R-CNN to include pixel-level instance segmentation.
- **Incremental Improvements:**
  - Added a mask prediction branch.
  - Enabled pixel-wise segmentation.
  - Simultaneous detection, localization, and segmentation.



Single-Stage Detection Networks

YOLO (You Only Look Once)

- **You Only Look Once (YOLO)** is a real-time object detection system that can detect multiple objects in an image in a single pass.
- **Principle:** divides an image into a grid and predicts bounding boxes, class probabilities, and objectness scores for each grid cell.
- **Known for:** speed and efficiency, suitable for real-time applications.
- **Note:** many versions

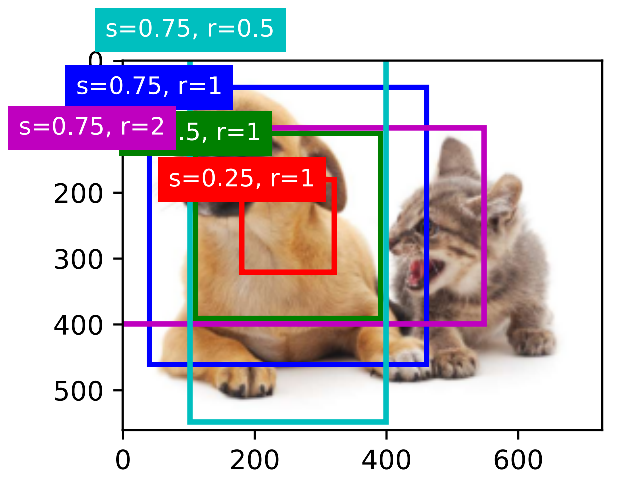
SSD (Single Shot MultiBox Detector)

- **SSD (Single Shot MultiBox Detector)** is another popular single-stage object detection system.
- **Principle:** combines the benefits of multi-scale feature maps and default boxes (priors) to efficiently predict object locations and categories.
- **Known for:** capable of detecting objects of various sizes and aspect ratios in a single forward pass, providing a good trade-off between speed and accuracy.
- **Note:** widely used in real-time and embedded systems for object detection.

Anchor-Based Detection

What Are Anchors?

- Anchors are predefined bounding boxes of various sizes and aspect ratios.
- Used in anchor-based detection for predicting object locations and attributes.



Why Are Anchors Useful?

1. **Dense, Accurate Object Coverage:**
   - Ensures detection at various scales and positions in an image.
   - Accurate prediction of object locations.
2. **Efficient Computation:**
   - Reduces computational complexity by predicting anchor adjustments.
3. **Training Stability:**
   - Stable training with a consistent reference point.

Going Further

Training Improvement

- Explore better loss functions for specific tasks, like Focal Loss.
- Consider techniques like Hard Online Example Mining (HOEM) to focus on challenging samples during training.

Transformer-Based Architectures

- **Segment Anything**: Segment Anything leverages vision transformers for versatile dense instance segmentation on a wide range of objects and scenarios.
- **DETR (Data-efficient Image Transformer)**: DETR revolutionizes object detection, predicting both class and bounding box simultaneously for better data efficiency.
- *and much much more...*

Lab Session: SSD Reimplementation

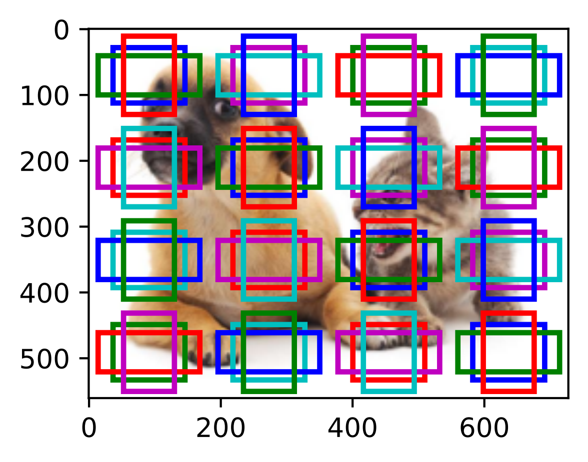
Architecture

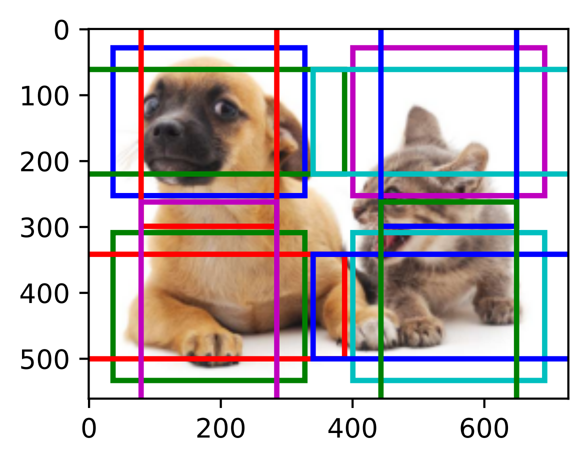
au tableau

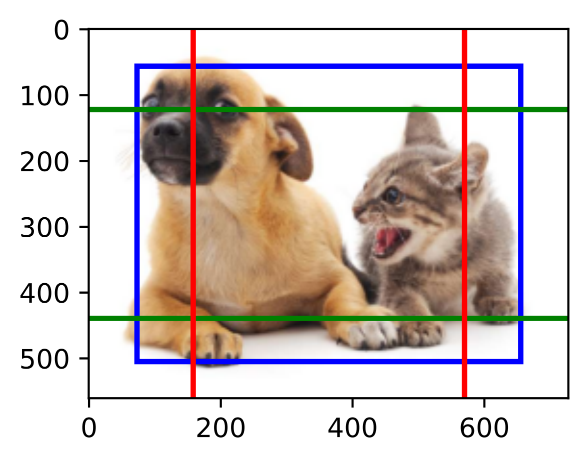
Anchors Generation

- `multibox_prior(data, sizes, ratios)` : box priors, $size * ratios - 1$

Multi-scale Features Maps



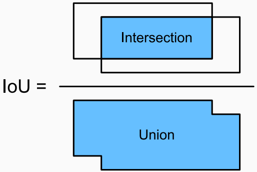

0
100
200
300
400
500
0
200
400
600


0
100
200
300
400
500
0
200
400
600

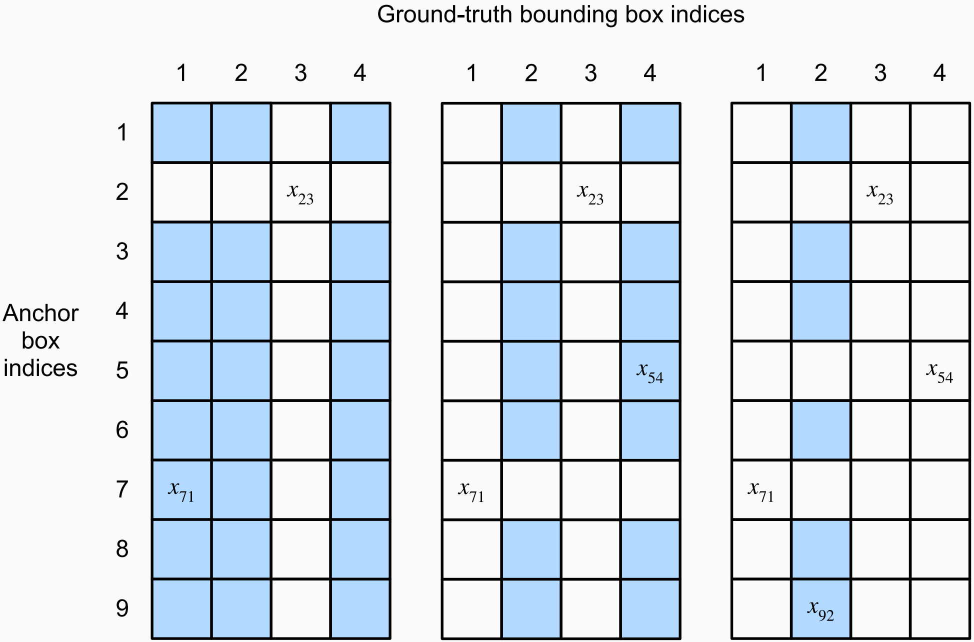
Anchors Matching with Ground-truth

- IoU computation: `box_iou(boxes1, boxes2)`

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$



- matching algorithm: `assign_anchor_to_bbox(ground_truth, anchors, device, iou_threshold=0.5)`



Generate Actual Targets

- `offset_boxes(anchors, assigned_bb, eps=1e-6)`: $(\delta_x, \delta_y, \log(\delta_w), \log(\delta_h))$
- `multibox_target(anchors, labels) -> bbox_offset, bbox_mask, class_labels`: final targets with object/background assignment

Predicting Bounding Boxes with Non-Maximum Suppression

- `nms(boxes, scores, iou_threshold) -> keep`: filter overlapping boxes
- `multibox_detection(cls_probs, offset_preds, anchors, nms_threshold=0.5) -> boxes`: finale detection

Losses

au tableau

Remaining Work and Grading

Remaining Work and Grading

TODOS:

- lab session
  - contribute the collective dataset ← **submission on Moodle, graded**
  - understand SSD in depth
  - rewrite (naively) parts of the code
  - train on the new dataset (add: validation measure + early stopper + model saver + seeding)
  - process the test set
  - submit your predictions on the test set ← **submission on Moodle, graded**
- quiz 4 ← **submission on Moodle, graded**
- Fill feedback forms for session 4 and course overall

Deadlines:

- Tomorrow everning for the dataset (25 images p. person, with annotations, 1 logo p. img)
- Thursday, Nov. 9th evening for the rest (will grade everything on Friday 11th)