Motivation
Why Perception?

Observation → Perception Stack → High-Level Planning

Low-Level Planning → Action

Sensors

Environment

Control
Why Perception?

Observation → Perception Stack → High-Level Planning → Low-Level Planning → Action

Sensors

Environment

Deep RL

Control
Sensors

- Vision-based
  - RGB
  - Thermal Infrared
  - Depth
- Laser scanner
- RADAR
- GPS/GNSS
- Ultrasonic Sensor
- Microphones
- Inertial Measurement Unit (IMU)
- Bumper

https://www.nuscenes.org/nuscenes
Sensors - FreiCAR
Sensors - FreiCAR

- VIVE VR Tracker
- Intel Depth Camera
- ZED Stereo RGB Camera
- SICK Laserscanner
Perception for Self-driving

- Lane Detection/Regression
- Object Detection (2D)
- Object Detection (3D)
- Semantic Segmentation
- Instance Segmentation
- Panoptic Segmentation
- Depth estimation (Mono/Stereo)
- Optical Flow estimation
- Scene Flow estimation
- Visual Odometry
- ...
2D Object Detection
2D Object Detection

Semantic Segmentation

Classification + Localization

Object Detection

Instance Segmentation

GRASS, CAT, TREE, SKY
No objects, just pixels

CAT
Single Object

DOG, DOG, CAT
Multiple Object
Image Classification

Feature Extractor / Encoder

Vector

Fully Connected

Class-wise softmax probabilities

0.7 Cat
0.2 Dog
0.1 Bird
Classification + Localization

1 Bounding Box: Regression + Classification

Feature Extractor / Encoder

Fully Connected

Vector

Fully Connected

Class Softmax

0.7 Cat
0.2 Dog
0.1 Bird

Classification loss

Bounding Box regression loss

x: 21
y: 50
w: 40
h: 30

Bounding Box coordinates
2D Object Detection

What if we have varying numbers of objects?
Region Proposals:
- Find regions likely to contain objects (region proposal method such as Selective Search)
- Select top 2000 regions and classify them as background / class X
R-CNN (2014)

1 - **Extract possible objects** using a region proposal method (the most popular one being Selective Search) to some fixed size
2 - **Extract features** from each region using a CNN
3 - **Classify each region** with SVMs (using hinge loss)
4 - **Predict offset loss** to correct the prediction values of location produced in region proposal stage (using least square l2 loss)


Figure copyright Ross Girshick, 2015. [source](source). Reproduced with permission.
R-CNN (2014)

**Issues:**

- Slow training
- Proposal method is not learned (can always produce bad ROIs)
- Super slow inference (~47s on 2014 modern hardware)
Non-Maximum Suppression

**Problem:** Multiple predicted anchor boxes for one object

**Solution:**
- Sort all the bounding boxes based on their scores (confidence).
- Select the box with the highest confidence. This box will be definitely kept.
- Calculate it’s IoU with all the other boxes.
- Remove the boxes which have an IoU of over threshold (e.g. 0.3 with the selected box. This will remove all the overlapping boxes that include the same object or a part of it.
- From the boxes left, select the next box (2nd highest score) and repeat the procedure.
Faster R-CNN (2015)

**Improvements** compared to R-CNN:
- Training is single-stage, using a multi-task loss
- SVM is replaced by Softmax classifier
- Use CNN-based ROI proposal generation instead of Selective Search method
- Extracted feature map is sent to two different networks: RPN and Fast R-CNN

**Result:**
- Higher detection quality (mAP)
  At runtime, the detection network requires 200ms per image
Region Proposal Network (RPN)

- First, the picture goes through conv layers and feature maps are extracted.
- Then a sliding window is used in RPN for each location over the feature map.
- For each location, \( k \) (\( k=9 \)) anchor boxes (3 scales of 128, 256 and 512, and 3 aspect ratios) are used for generating region proposals.
- A classification layer outputs 2k scores whether there is an object or not for k boxes.
- A regression layer outputs 4k for the coordinates (box center coordinates, width and height) of k boxes.
ROI Pooling

Hi-res input image: 3 x 640 x 480 with region proposal

Hi-res conv features: 512 x 20 x 15;
Projected region proposal is e.g. 512 x 18 x 8 (varies per proposal)

RoI conv features: 512 x 7 x 7 for region proposal

Fully-connected layers expect low-res conv features: 512 x 7 x 7

ROI Pooling

Hi-res input image: 3 x 640 x 480 with region proposal

Hi-res conv features: 512 x 20 x 15;

Projected region proposal is e.g. 512 x 18 x 8 (varies per proposal)

Roi conv features: 512 x 7 x 7 for region proposal

Fully-connected layers expect low-res conv features: 512 x 7 x 7

2D Object Detection - losses

- Bounding box: **Multi-Class Classification**

\[ \mathcal{L}_{\text{class}} = - \sum_{i} C t_i \log(s_i) \]

- Bounding box: **Regression**

\[ L_{\text{loc}}(t^u, v) = \sum_{i \in x, y, w, h} \text{smooth}_{L_1}(t^u_i - v_i) \]

\[
\text{smooth}_{L_1}(x) = \begin{cases} 
0.5x^2 & \text{if } |x| < 1 \\
|x| - 0.5 & \text{otherwise}
\end{cases}
\]
Intersection over union

\[
\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}
\]
Intersection over union
Mean average precision (mAP)

Confusion matrix

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Positive</td>
</tr>
<tr>
<td>Positive</td>
<td>True Positive</td>
</tr>
<tr>
<td>Negative</td>
<td>False Negative</td>
</tr>
</tbody>
</table>

True positive: IoU > threshold (0.5, 0.8, 0.95, ..)

Recall = \( \frac{tp}{tp + fn} \)

Precision = \( \frac{tp}{tp + fp} \)
Mean average precision (mAP)
Mean average precision (mAP)

Varying the model prediction score from 0.0 to 1.0 → P/R curve

mAP: mean of AP for all classes (person, vehicle, lamp, ...)
→ Depends on IoU threshold for counting tp/tn/fp/fn in confusion matrix

mAP@0.5: mAP with IoU threshold of 0.5

In practice: use library E.g.: https://github.com/Cartucho/mAP
Two-Stage Detectors
- R-CNN (2014)
- Faster RCNN (2015)

- High mAP
- Flexible to integrate other task (e.g. object mask prediction)
- Slow
- More complex architecture

One-Stage Detectors
- YOLO family (2016-2018)
- SSD (2015)
- EfficientDet (2019)

- Faster than 2-stage detectors (real-time possible)
- Simple architecture
- Not as flexible as 2-stage detectors
- worse performance
EfficientDet (2019)
EfficientDet (2019)
Block 1 - Task 1: Object Detection
Block 1 - Task 1: Object Detection
Further reading

- Stanford course cs231n lecture 11: Detection and Segmentation
Thank you!