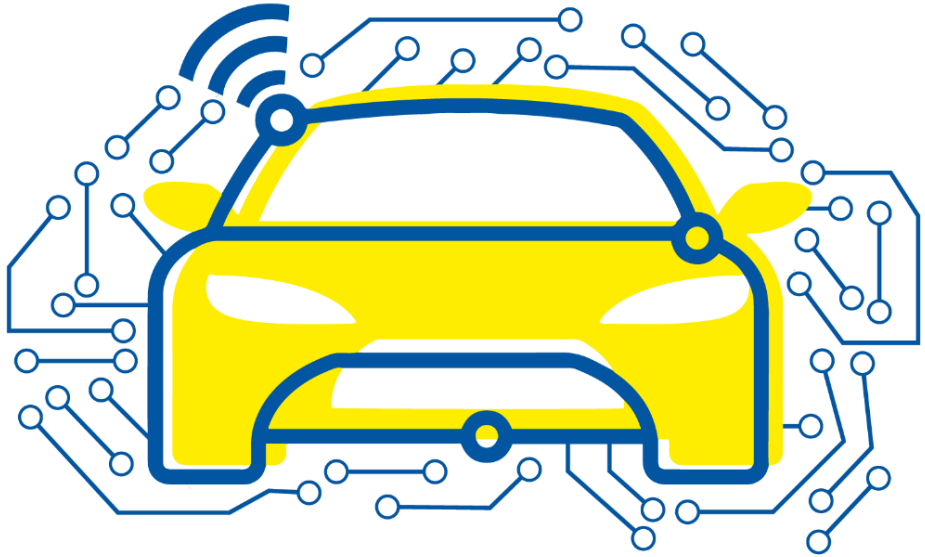
Automated and Connected Driving Challenges

Section 2 – Sensor Data Processing

Object Detection

Deep Learning



Bastian Lampe

Institute for Automotive Engineering

Object Detection - Deep Learning

Overview

- Improved performance of **Deep Learning** methods especially in computer vision due to efficient and fast computations

- Supervised learning
  - Large amount of labeled **reference** / **ground truth data**

- End-to-end learning
  - **Input**: sensor data
  - **Output**: detected objects with **bounding boxes** and **classifications**

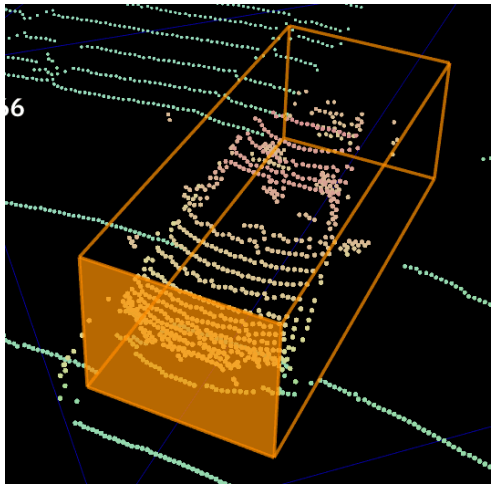
- 2D object detection in camera images

- **3D object detection in LiDAR point clouds**

→ Main concepts are comparable, network input is more complicated in 3D space

Object Detection - Deep Learning

Single Object Detection



- Simplified label example:
  - Small 3D point cloud
  - Maximal one object per point cloud or none
  - Three classes: 1 – Pedestrian, 2 – Car, 3 – Motorcycle

$$y = \begin{bmatrix} p_o \\ b_x \\ b_y \\ b_z \\ b_l \\ b_w \\ b_h \\ \phi \\ p_{c_1} \\ p_{c_2} \\ p_{c_3} \end{bmatrix}$$

confidence probability
$p_o = p(object)$

location and size (3D bounding box)

orientation

conditional class probability
$p_{c_n} = p(class_n|object)$

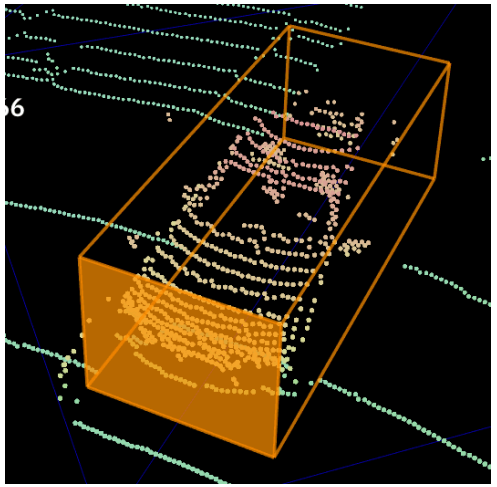
- B = 8 is number of bounding box characteristics
- N = 3 is number of classes

Images: scale

Object Detection - Deep Learning

Single Object Detection

- Simplified label example:
  - Small 3D point cloud
  - Maximal one object per point cloud or none
  - Three classes: 1 – Pedestrian, 2 – Car, 3 – Motorcycle



$$y = \begin{bmatrix} \mathbf{1} \\ 32.1 \\ 0.5 \\ 0.1 \\ 4.6 \\ 1.7 \\ 1.5 \\ 0.2 \\ 0 \\ \mathbf{1} \\ 0 \end{bmatrix}$$

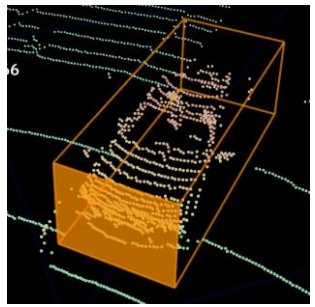
$$y = \begin{bmatrix} \mathrm{p}_o \\ \mathrm{b}_x \\ \mathrm{b}_y \\ \mathrm{b}_z \\ \mathrm{b}_l \\ \mathrm{b}_w \\ \mathrm{b}_h \\ \phi \\ p_{c_1} \\ p_{c_2} \\ p_{c_3} \end{bmatrix}$$

confidence probability
$\mathrm{p}_o = p(object)$

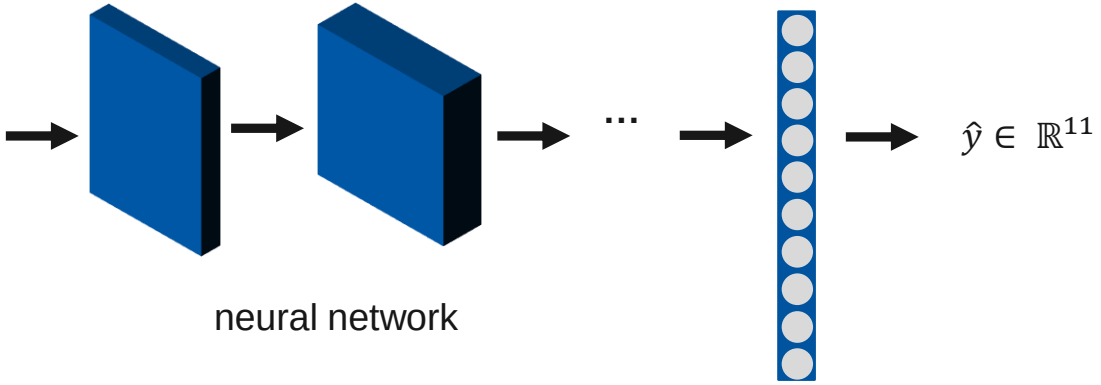
location and size (3D bounding box)

orientation

conditional class probability
$\mathrm{p}_{c_n} = p(class_n|object)$



output layer with
$B + N = 11$ neurons



neural network

- B = 8 is number of bounding box characteristics
- N = 3 is number of classes

Images: scale

Object Detection - Deep Learning

Multi Object Detection

- Idea: **divide** domain **into** a **grid** of size $H \times W$ (e.g. 7 x 8 cells)

- **Solve** previous task **for each grid cell**
  - Compute **target tensor** for object existence in **every** grid **cell**
  - Labels and outputs of network are tensors of size $H \times W \times (\mathrm{B} + \mathrm{N})$

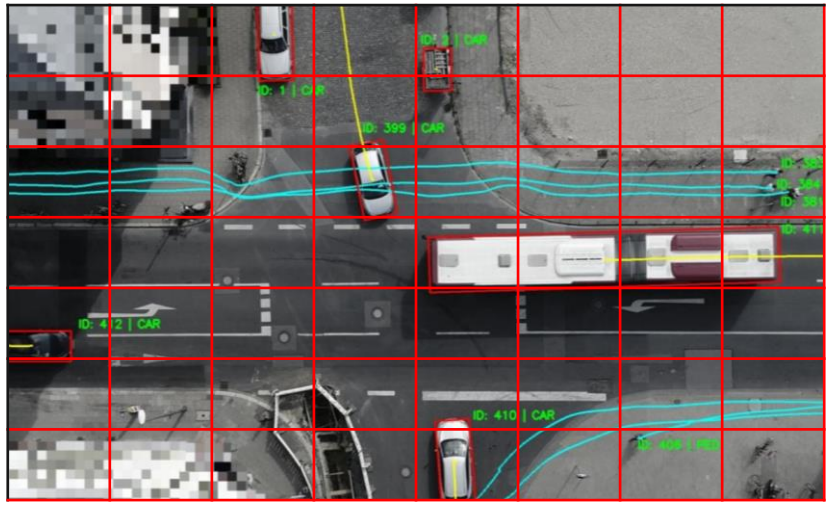


Image: ika

Object Detection - Deep Learning

Multi Object Detection

- Idea: **divide** domain **into** a **grid** of size $H \times W$ (e.g. 7 x 8 cells)

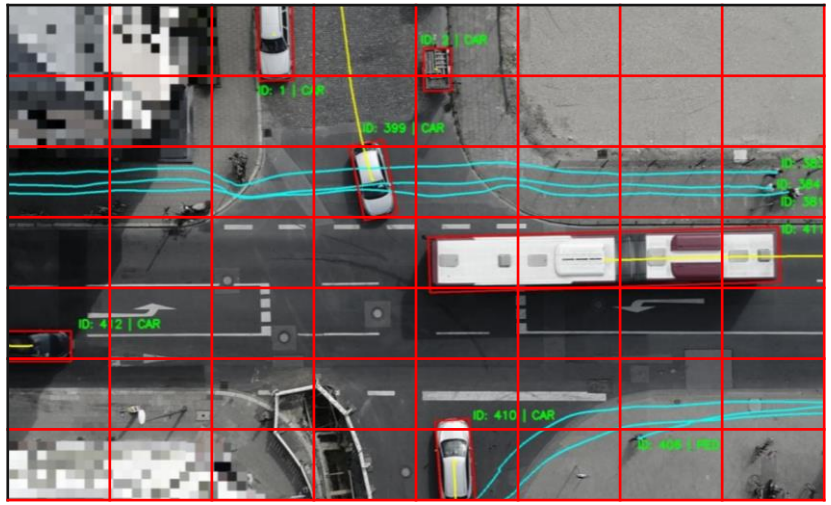


Image: ika

- **Solve** previous task **for each grid cell**
  - Compute **target tensor** for object existence in **every** grid **cell**
  - Labels and outputs of network are tensors of size $H \times W \times (\mathrm{B} + \mathrm{N})$

- Problems:

  Each grid cell is **limited to** detect only **one object (type)**

  ⇩ **multiple** object **hypotheses** per cell

  Anchor Boxes

  Same objects are **detected multiple times** from different anchors

  ⇩ **reduce** detections based on **probabilities**

  Non-Maximum Suppression

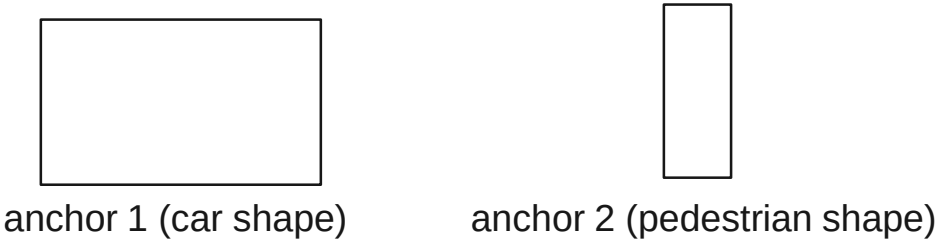
Object Detection - Deep Learning

Anchor Boxes

- Problem: each grid cell is **limited to** detect only **one object (type)**

- Sometimes **multiple overlapping objects** with center point **in same cell**
- Objects can **come** in **different sizes** and **rotations (different types)**

→ Define A different **shapes** / object **hypotheses** (anchor boxes)



anchor 1 (car shape) anchor 2 (pedestrian shape)

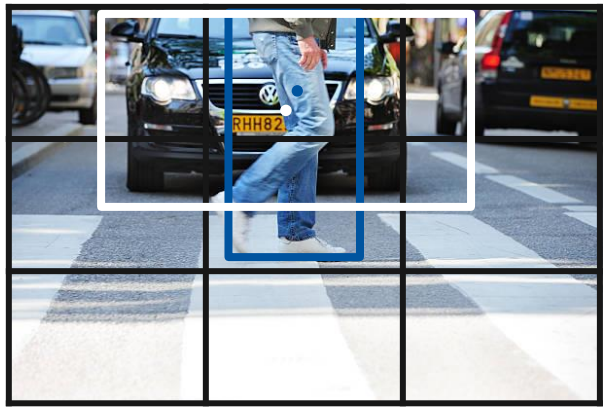


Image: ika, istock

Object Detection - Deep Learning

Anchor Boxes

- Problem: each grid cell is **limited to** detect only **one object (type)**

- Sometimes **multiple overlapping objects** with center point **in same cell**
- Objects can **come** in **different sizes** and **rotations (different types)**

→ Define A different **shapes** / object **hypotheses** (anchor boxes)

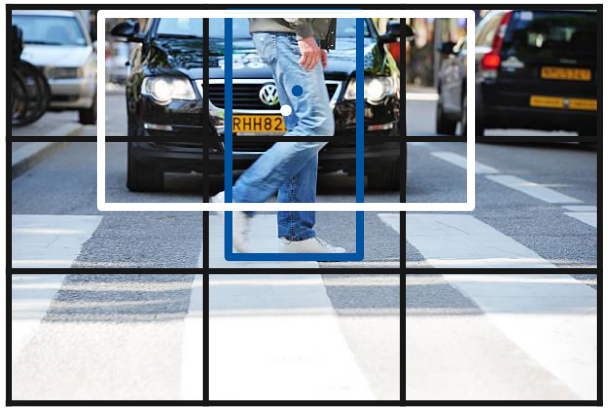
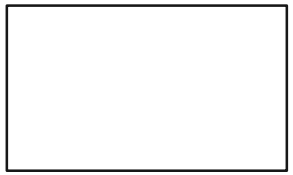
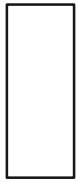


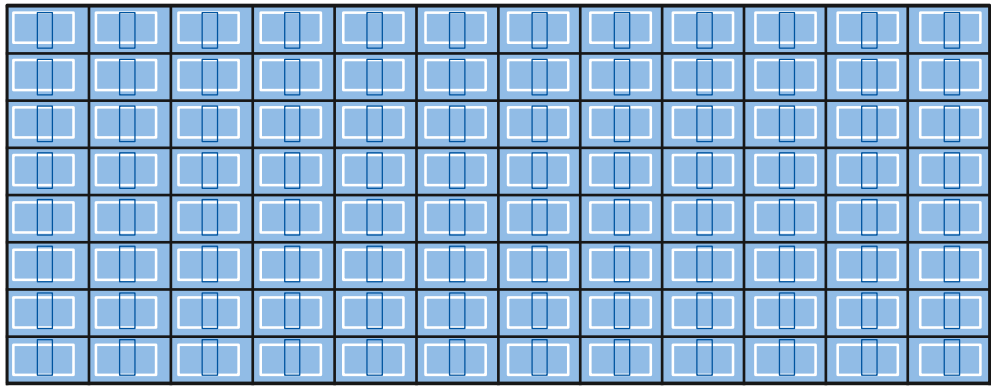
Image: ika, istock



anchor 1 (car shape)



anchor 2 (pedestrian shape)



- More **variance in possible object** means more different anchor types
- Anchor box can be **larger than grid cell**

Object Detection - Deep Learning

Anchor Boxes

- Target vector $\boldsymbol{y}_{ij}$ is calculated for each cell

$$\boldsymbol{y}_{ij} = [\underbrace{p_{o_1}, b_{x_1}, b_{y_1}, b_{h_1}, b_{w_1}}_{B}, \dots, \underbrace{p_{o_A}, b_{x_A}, b_{y_A}, b_{h_A}, b_{w_A}}_{B}, \underbrace{p_{c_1}, \dots, p_{c_N}}_{N}] \in \mathbb{R}^{A \cdot B + N}$$

$A \cdot B$

- **Total** network output of **size**: $H \times W \times (\mathrm{A} \cdot B + N)$

- A **confidence score** CS combines the confidence probability of there being some object p_{o_a} with the conditional class probability $p_{c_n}(class|object)$

- $$\boldsymbol{CS_a} = \begin{bmatrix} p_{c_1} \\ \dots \\ p_{c_N} \end{bmatrix} \cdot p_{o_i} = \begin{bmatrix} p_{c_1} \cdot p_{o\,i} \\ \dots \\ p_{c_N} \cdot p_{o\,i} \end{bmatrix} = \begin{bmatrix} CS_{c_1,i} \\ \dots \\ CS_{c_N,i} \end{bmatrix}$$

- We can therefore compute a confidence score $CS_{c,a}$ **for each class** for each anchor type for each grid cell.

Object Detection - Deep Learning

Thresholding and Non-Maximum Suppression

- Problem: same objects are **detected multiple times** from different cells

- **Confidence score thresholding**
  - Delete all detections below threshold ^{min}CS

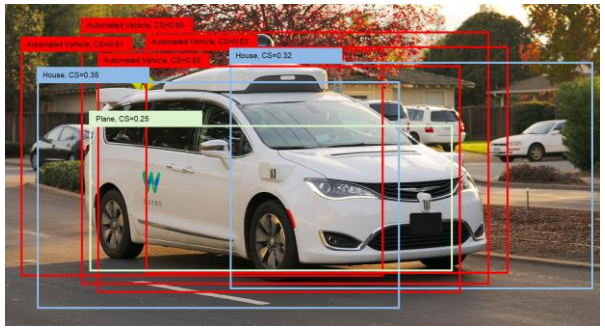
  → *High confidence hypotheses*

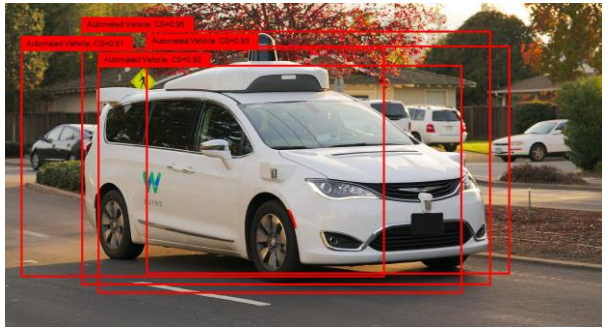
- **Non-maximum suppression**
  - **Loop** (while unmarked boxes remain)
    - Find unmarked bounding box with highest confidence score CS
    - Delete all unmarked bounding boxes with overlap larger ^{min}IoU

  → *Final hypotheses*

- Object with highest confidence score remains

- Apply non-maximum suppression **for each class** individually





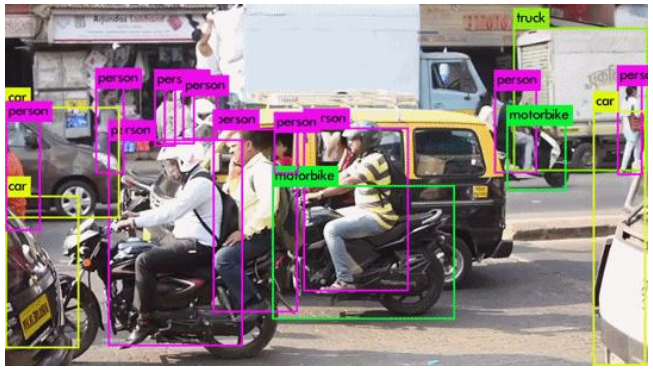
Images: wikipedia

Object Detection - Deep Learning

Common Object Detection Architectures

2D Examples

- YOLO (**Y**ou **O**nly **L**ook **O**nce)
  - **Divide** domain in **grid cells**
  - Predictions for each anchor and each cell
  - Non-maximum suppression
- Faster **R-CNN** (two stages)
  - Initial sub-segmentation
  - Extract larger regions of interest
  - Compute object detection on RoI's



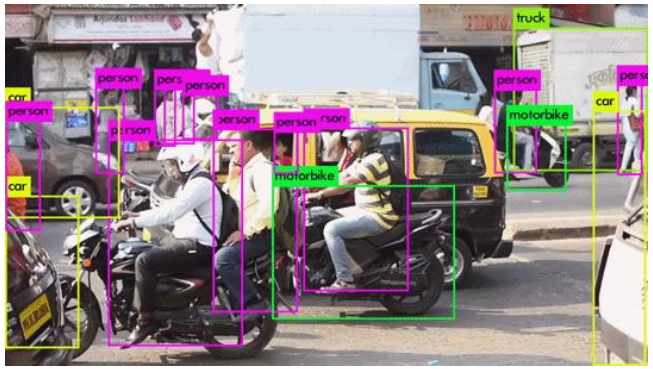
Video: pinterest

Object Detection - Deep Learning

Common Object Detection Architectures

2D Examples

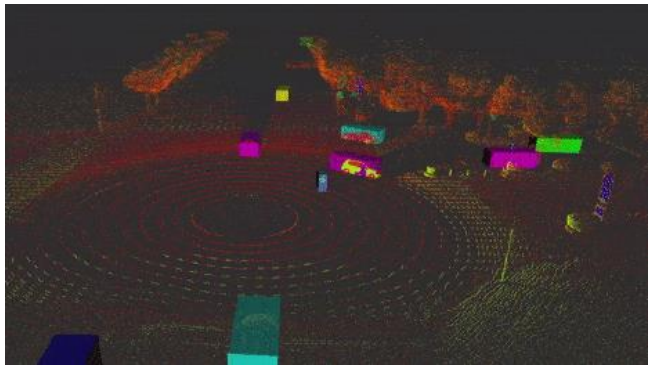
- YOLO (**Y**ou **O**nly **L**ook **O**nce)
  - **Divide** domain in **grid cells**
  - Predictions for each anchor and each cell
  - Non-maximum suppression
- Faster **R-CNN** (two stages)
  - Initial sub-segmentation
  - Extract larger regions of interest
  - Compute object detection on RoI's



Video: pinterest

3D Examples

- LaserNet
  - Map point cloud to 2D ***pseudo image***
  - 2D object detection
- VoxelNet
  - Divide point cloud into 3D voxels
  - **3D convolutions**
- **PointPillars**
  - **Divide** point cloud into **xy-grid** with **vertical pillars**
  - **Feature encoding** for each pillar
  - **2D CNN** backbone with stacked features in z-dimension

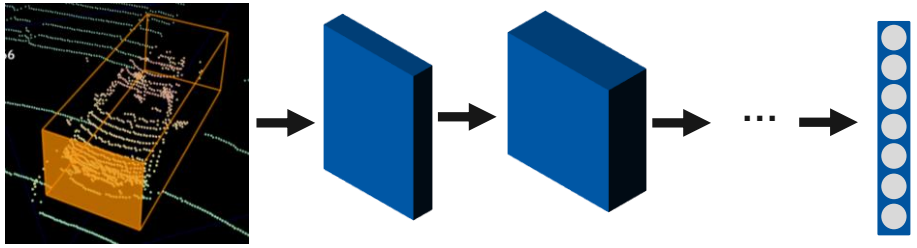


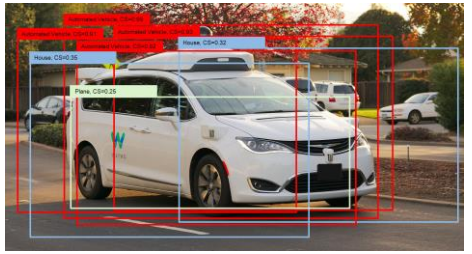
Video: shangzhouye

Object Detection - Deep Learning

Summary

- **Target vector** for 3D bounding boxes
- **Grid cell** approach for detection of **multiple objects**
- **Anchor box** principle
- **Non-maximum suppression**
- Many common deep learning **approaches**





Images: ika, scale, wikipedia