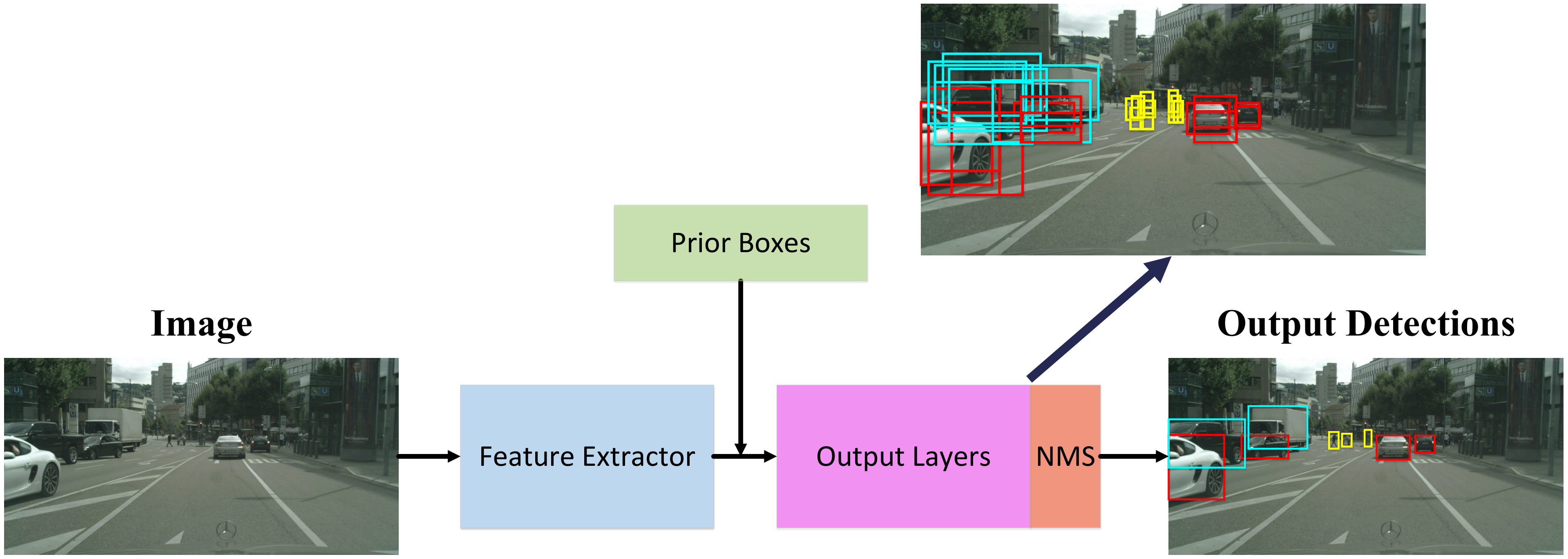
Training Vs Inference

Course 3, Module 4, Lesson 3

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

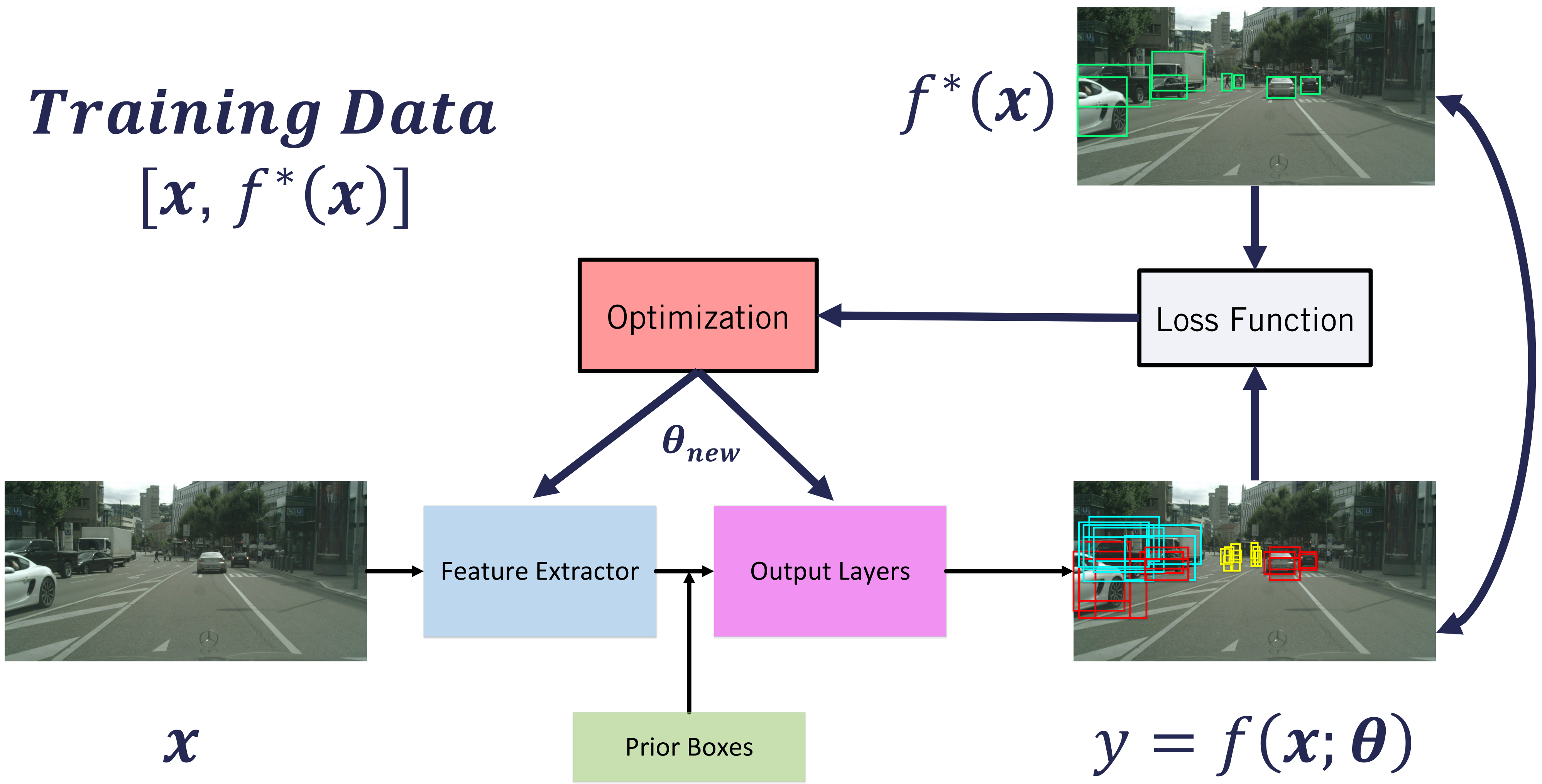
ConvNets For 2D Object Detection



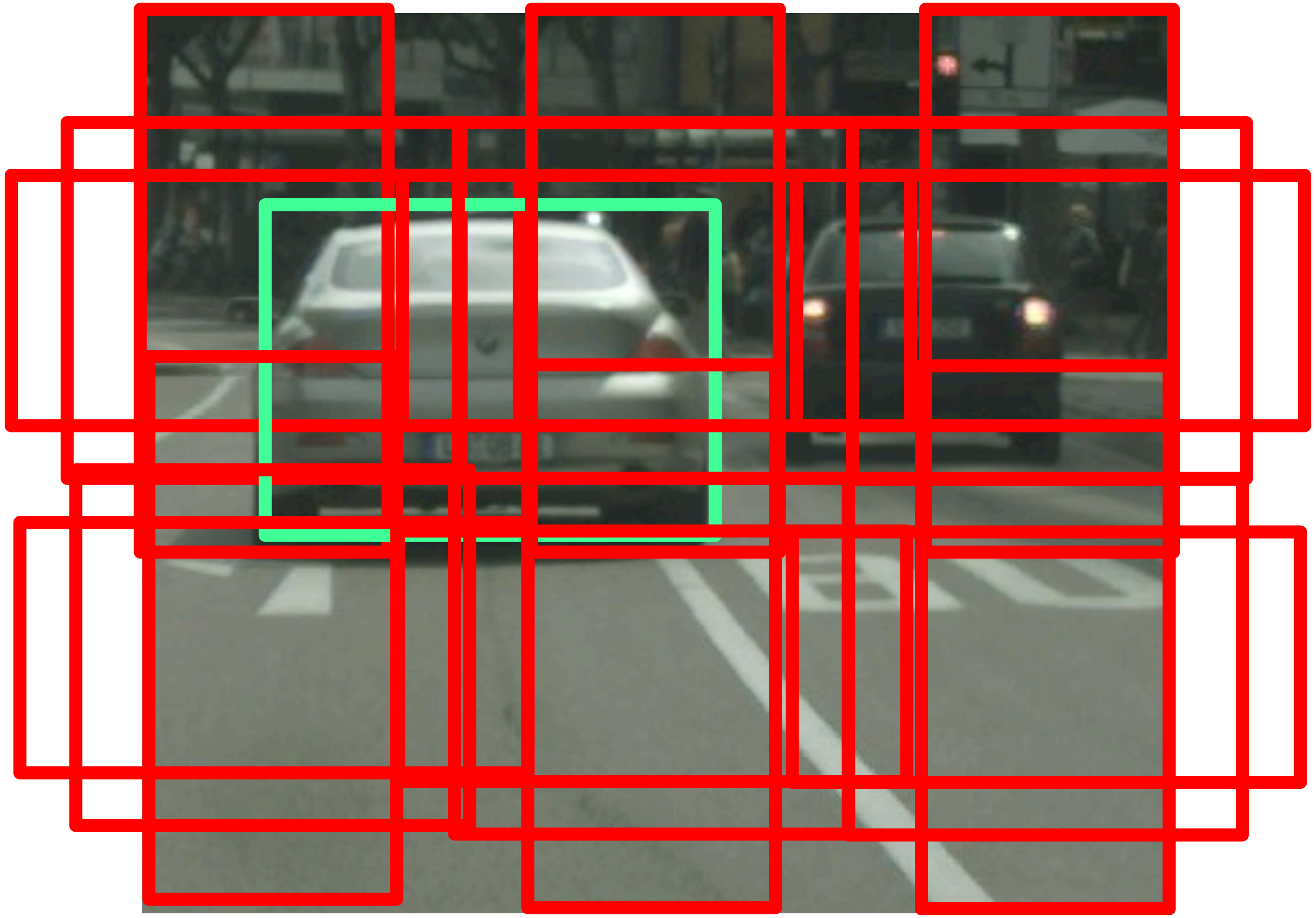
Learning objectives

- Learn how to handle multiple detections per object during training through **minibatch selection**
- Learn how to handle multiple detections per object during inference, through **non-maximum suppression** (NMS)

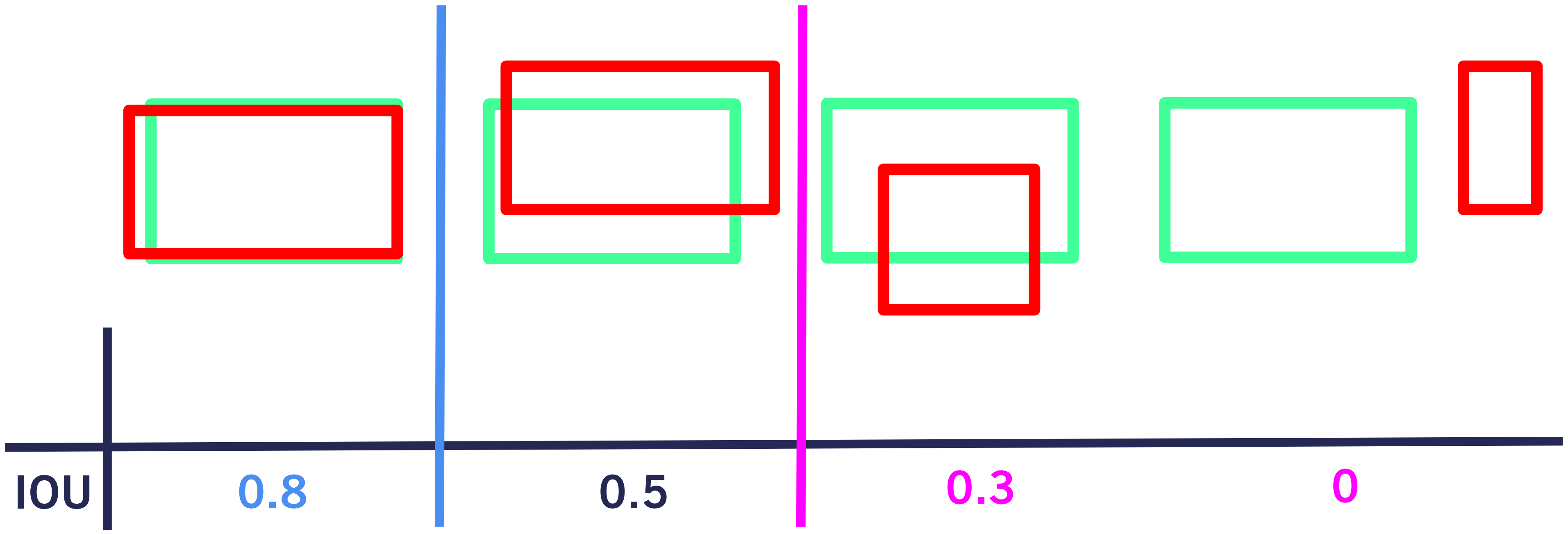
2D Object Detector Training



MiniBatch Selection



MiniBatch Selection



- Negative Member Threshold: < 0.4
- Positive Member Threshold: > 0.6

Anchor in-between discarded

Minibatch Selection

- Negative anchors target:
  - **Classification:** Background
  - **Regression:** None
- Positive anchors target:
  - **Classification:** Category of the ground truth bounding box
  - **Regression:** Align box parameters with highest IOU ground truth bounding box

Minibatch Selection

- **Problem:** Majority of anchors are negatives results in neural network will label all detections as background
- **Solution:** Sample a chosen **minibatch size**, with 3:1 ratio of negative to positive anchors to eliminate bias towards the negative class
- Choose negatives with **highest classification loss** (online hard negative mining) to be included in the minibatch

- Example: minibatch size is 64→ 48 hardest negatives and 16 positives

biased towards negative class

Classification Loss

$$L_{cls} = \frac{1}{N_{total}} \sum_{i} CrossEntropy(s_i^*, s_i)$$

- N_{total} is the size of our minibatch
- s_i is the output of the neural network
- s_i^* is the anchor classification target:
  - **Background** if anchor is negative
  - **Ground truth box class** if anchor is positive

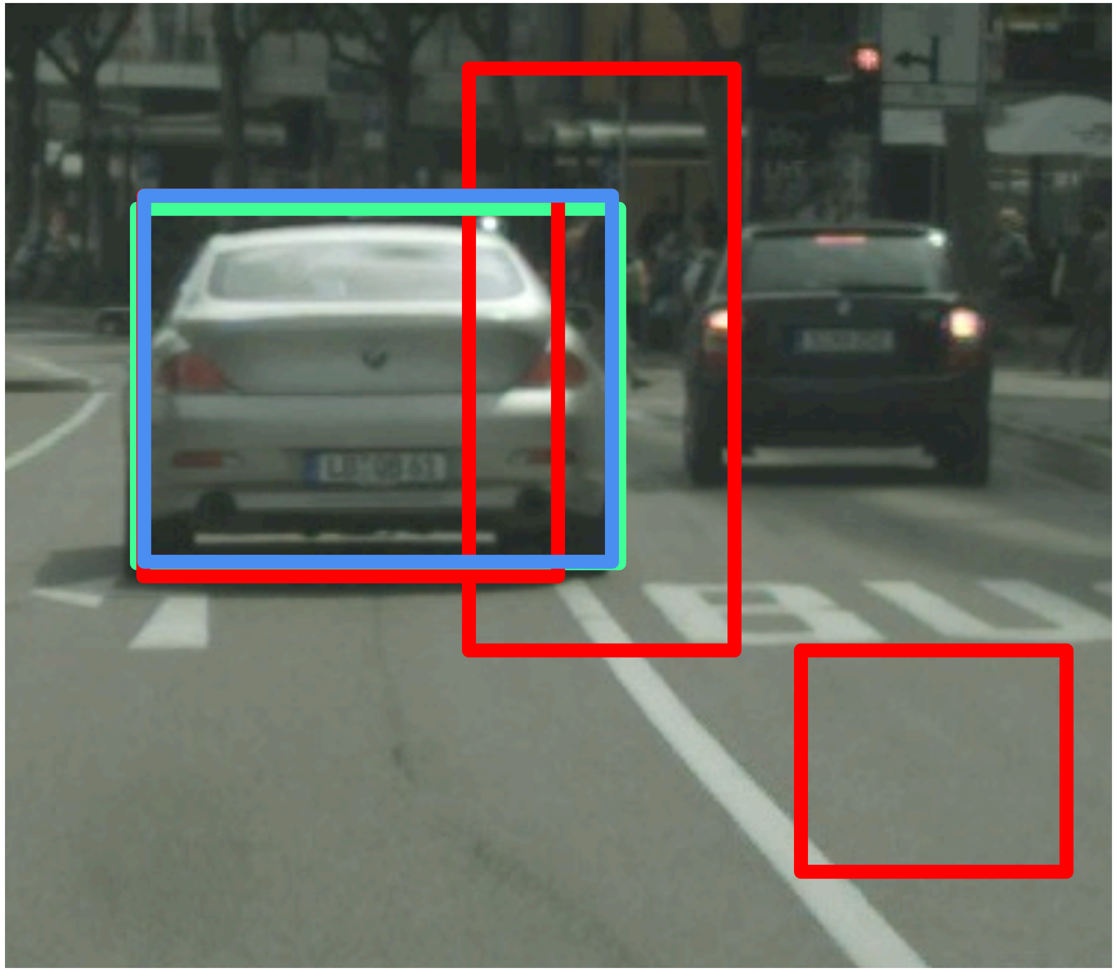
Regression Loss

$$L_{reg} = \frac{1}{N_p} \sum_i p_i L_2(b_i^*, b_i)$$

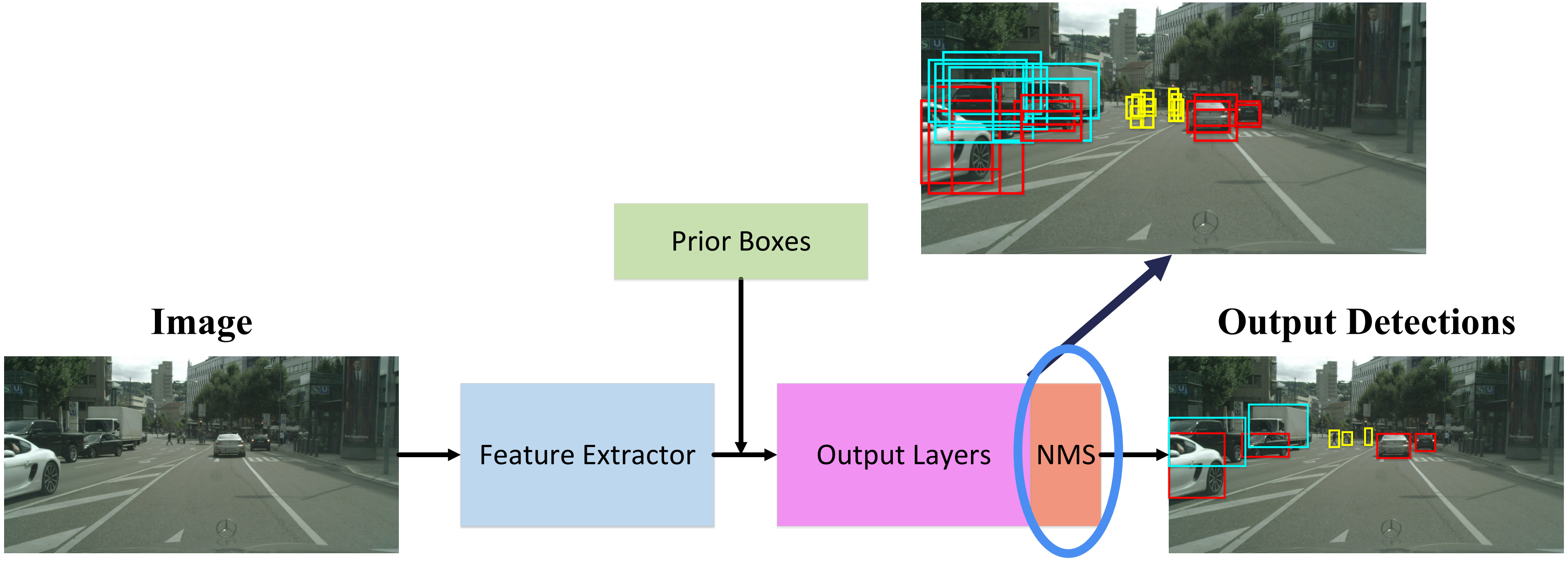
- p_i is 0 if anchor is negative and 1 if anchor is positive
- N_p is the number of positive anchors in the minibatch
- b_i^* is the ground truth bounding box
- b_i is the estimated bounding box, applying the regressed residuals to the anchor box parameters

modify the box parameters by an additive residual or a multiplicative scale.

Visual Representation Of Training



Inference Time



Non-Maximum Suppression

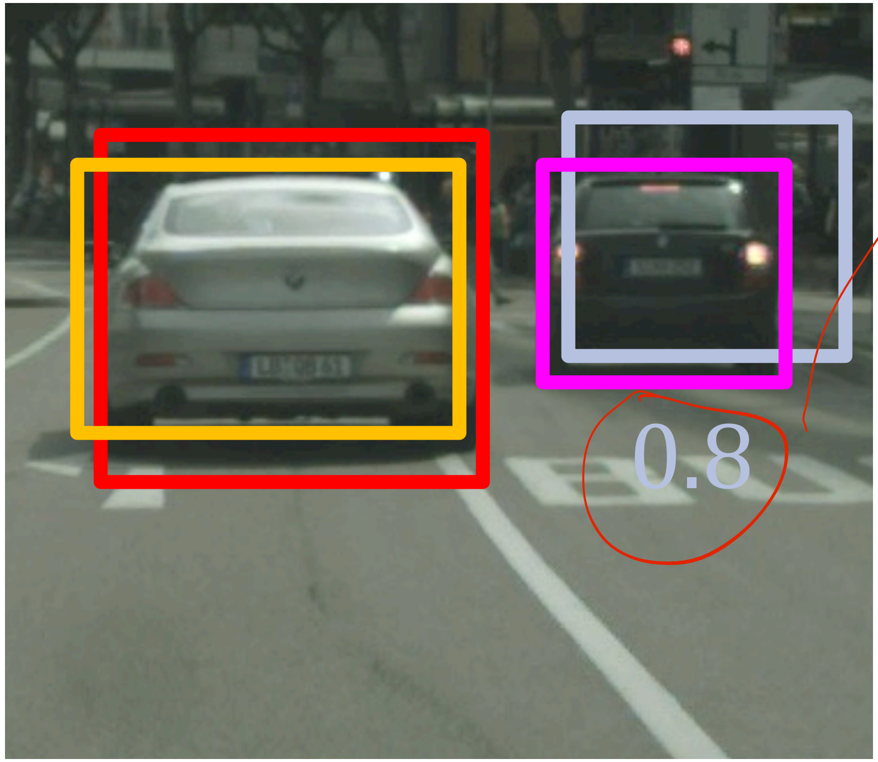
list of bounding box

Input:
$B = \{B_1 ... B_n\} | B_i = (x_i, y_i, w_i, h_i, s_i) \forall i \in [1, n]$
IOU threshold η
begin
  $\bar{B} = Sort(B, s, \downarrow)$
  $D = \emptyset$ [ ]
  for $b \in \bar{B}$ *and* $\bar{B}$ *not* $\emptyset$ **do**
    $b_{max} = b$
    $\bar{B} \leftarrow \bar{B} \backslash b_{max}$ → remove bmax from $\bar{B}$
    $D \leftarrow D \cup b_{max}$ → add bmax to D
    for $b_i \in \bar{B} \backslash b_{max}$ **do**
      if $IOU(b_{max}, b_i) \geq \eta$ **then**
        $\bar{B} \leftarrow \bar{B} \backslash b_i$
      end
    end
  end
  Output: D
end

Non-Maximum Suppression

$\geq \quad \eta = 0.7$

remove B_3 from $\bar{B}$



$$\boldsymbol{B} = \{B_1, B_2, B_3, B_4\}$$

$$\bar{\boldsymbol{B}} = \{B_1, B_2, B_3, B_4\}$$

$$\boldsymbol{S} = \{0.98, 0.94, 0.6, 0.45\}$$

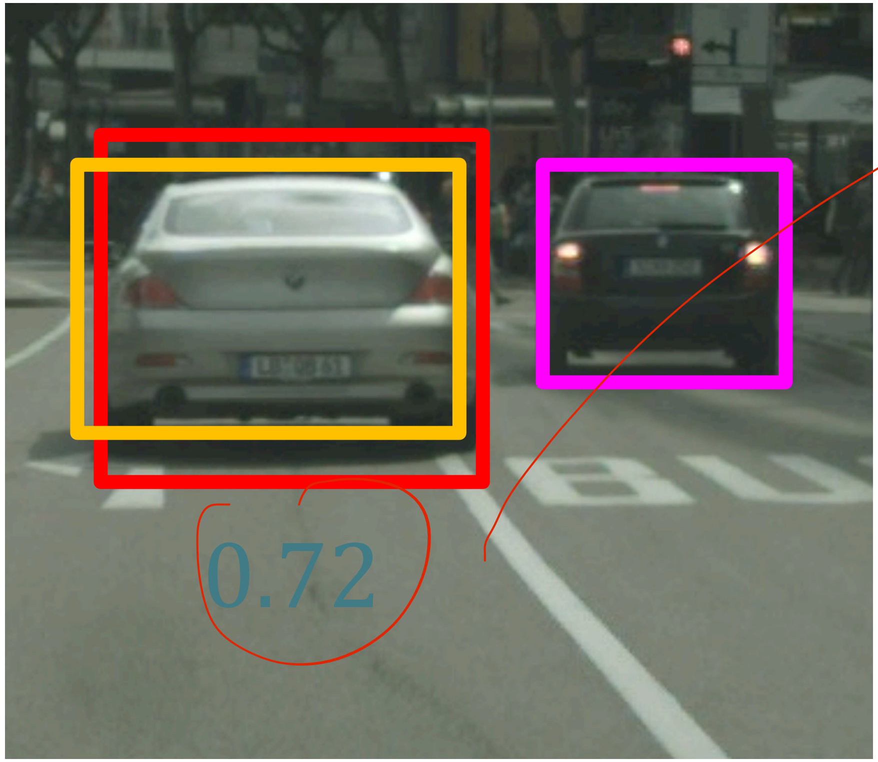
$$\boldsymbol{D} = \{\ \}$$

$$b_{max} = \{B_1\}$$

Non-Maximum Suppression

$$\geq \quad \eta = 0.7$$

remove B_4 from $\bar{B}$



$$\boldsymbol{B} = \{B_1, B_2, B_3, B_4\}$$

$$\bar{\boldsymbol{B}} = \{B_2, B_4\}$$

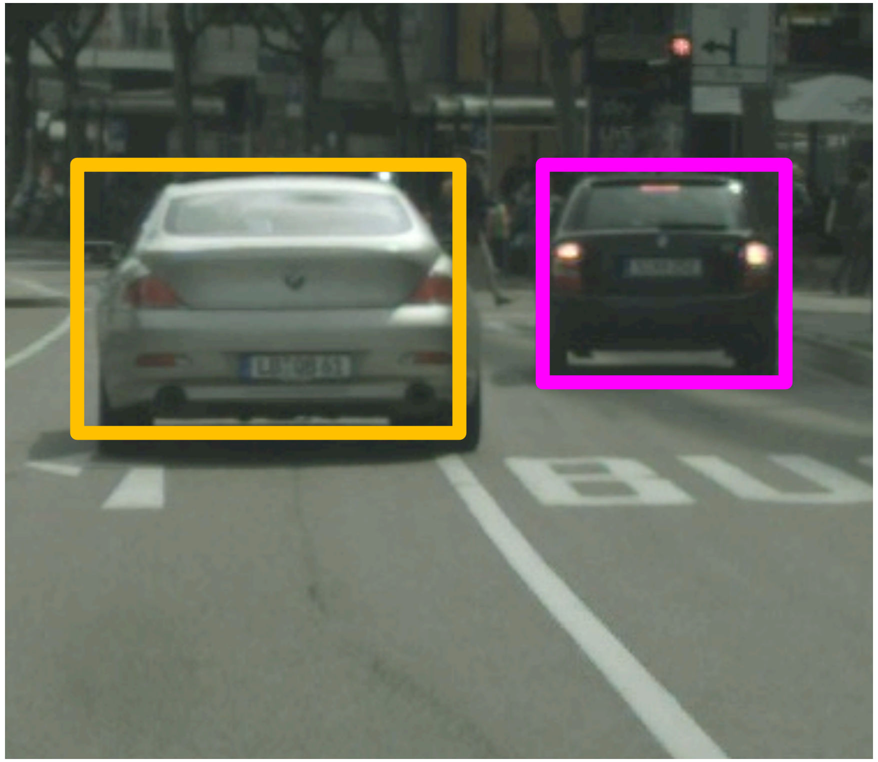
$$\boldsymbol{S} = \{0.94, 0.45\}$$

$$\boldsymbol{D} = \{B_1\}$$

$$b_{max} = \{B_2\}$$

Non-Maximum Suppression

$\eta = 0.7$



$\boldsymbol{B} = \{B_1, B_2, B_3, B_4\}$

$\bar{\boldsymbol{B}} = \{\}$

$\boldsymbol{S} = \{\}$

$\boldsymbol{D} = \{B_1, B_2\}$ return

$b_{max} = \{B_2\}$

Summary

- To train a neural network for 2D object detection, use minibatch selection on anchors
- For inference, use Non-Maximum Suppression to get a single output bounding box per object
- **Next: Using 2D object detectors for autonomous driving**