

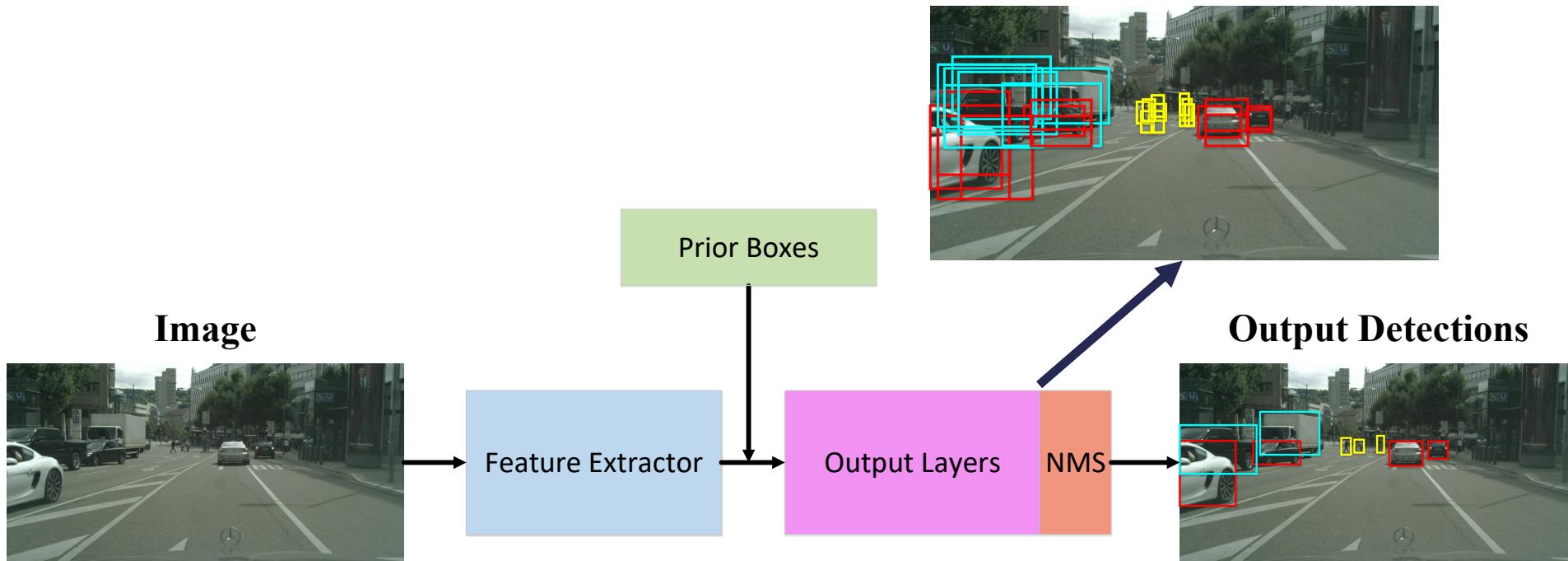
Training Vs Inference

Course 3, Module 4, Lesson 3



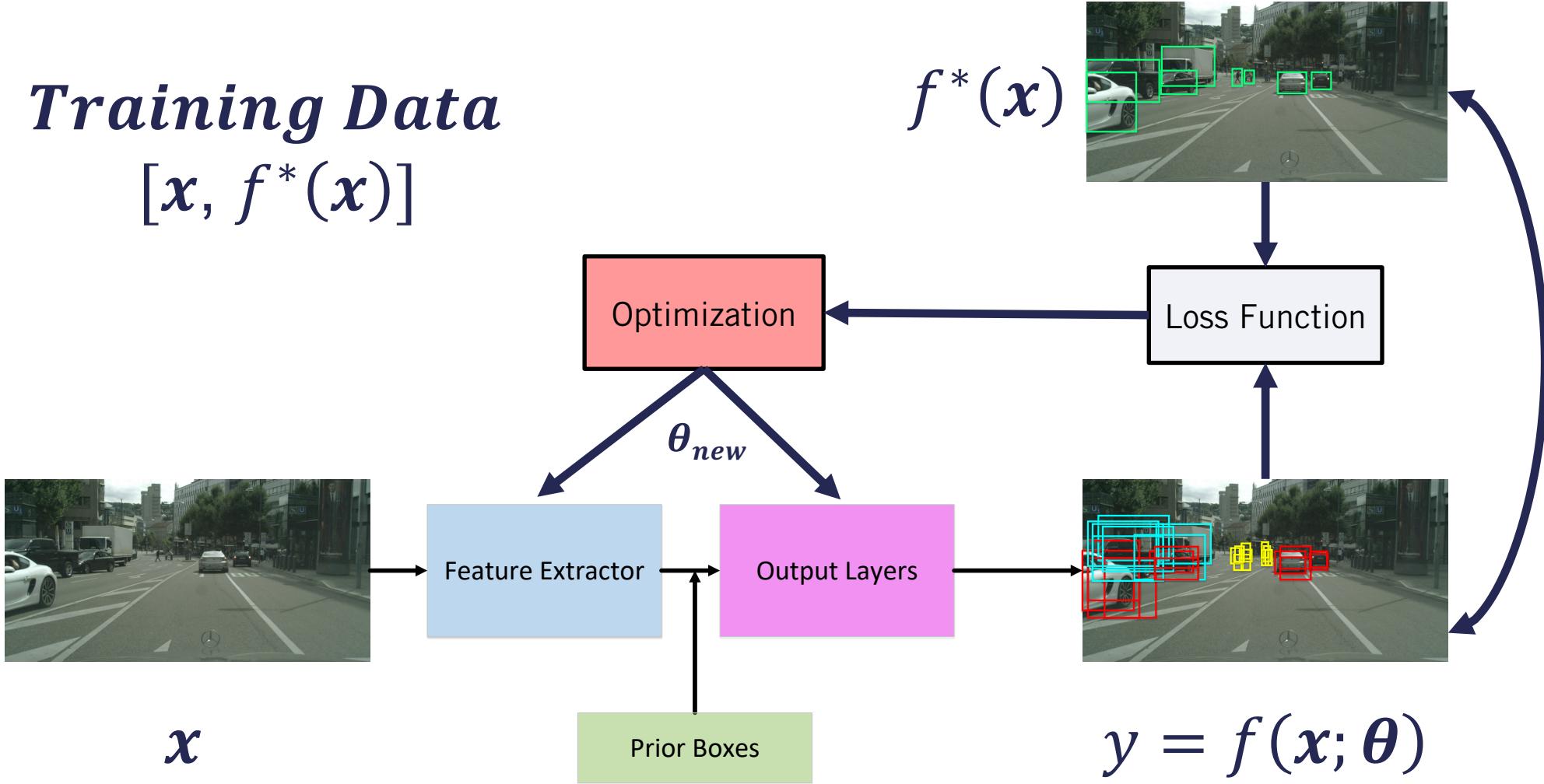
UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

ConvNets For 2D Object Detection

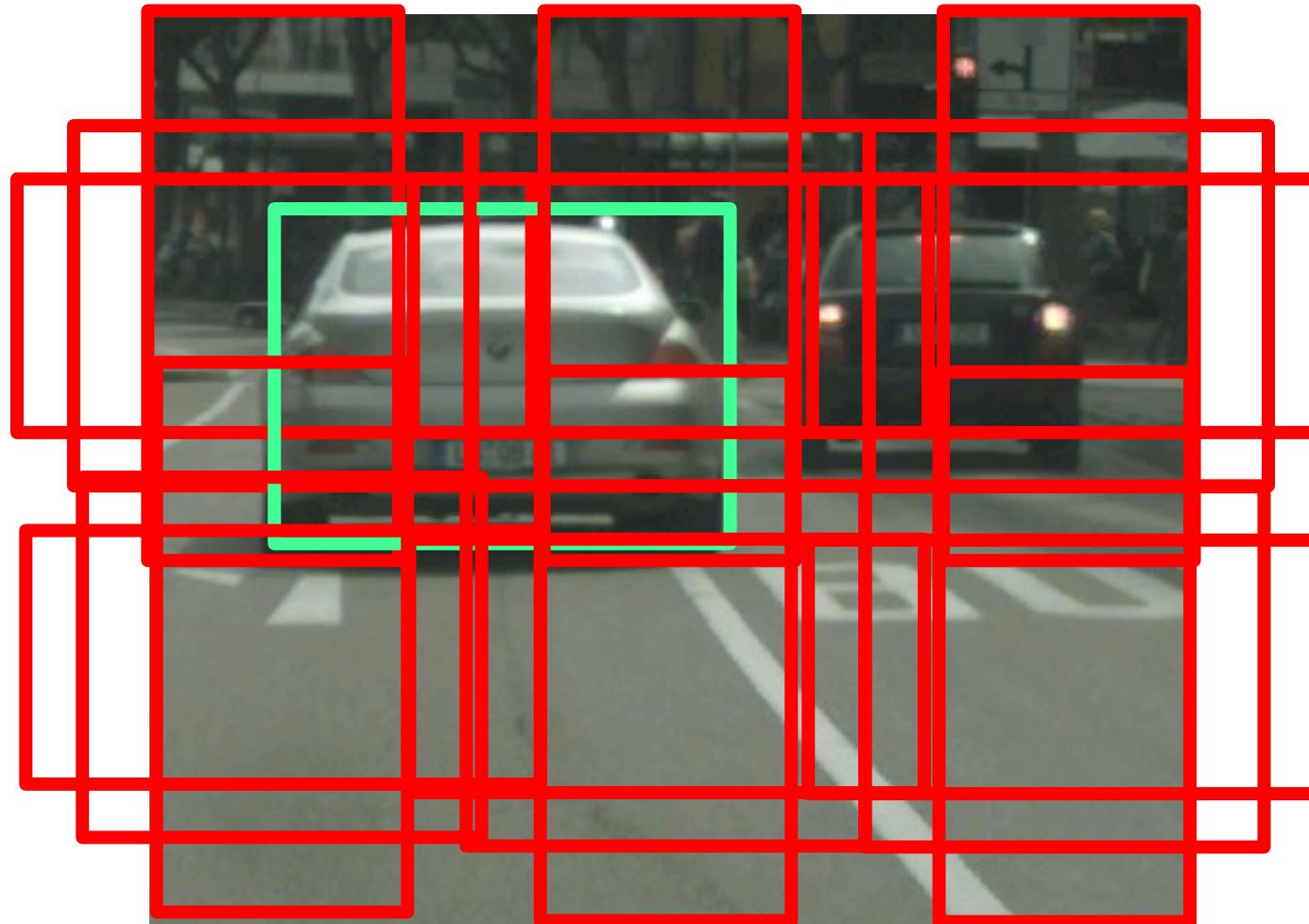


2D Object Detector Training

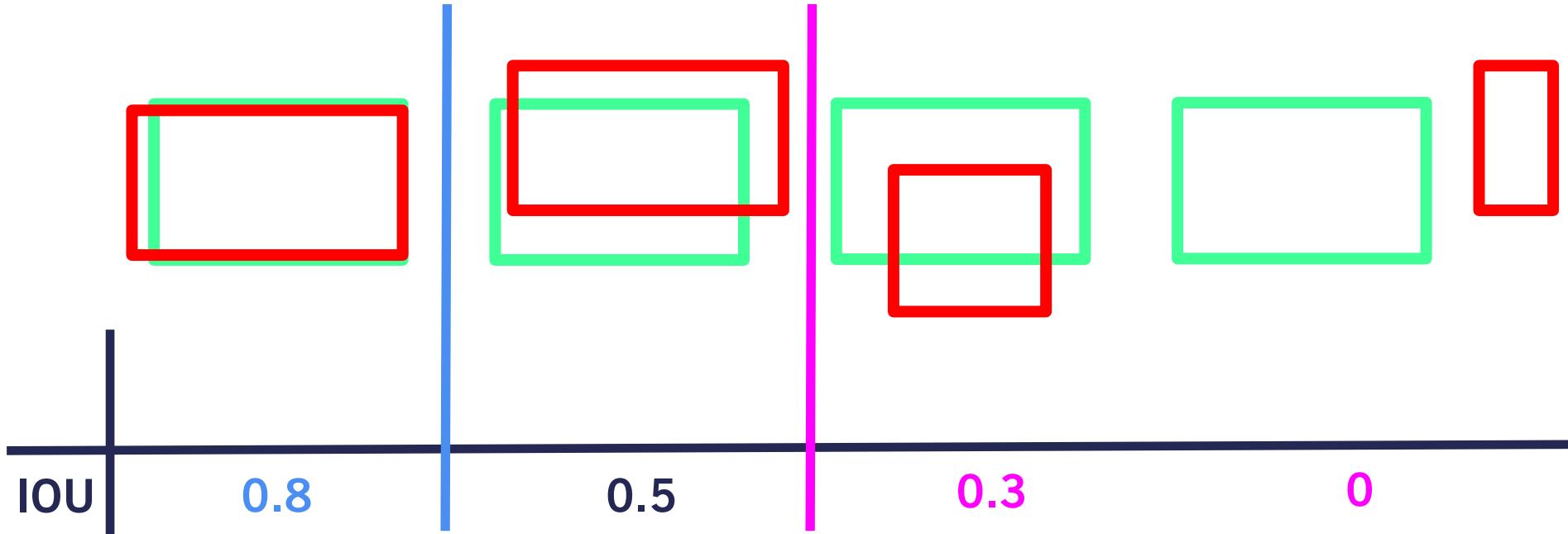
Training Data
 $[x, f^*(x)]$



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 $64 \rightarrow 48$ hardest negatives and 16 positives

biased
towards
negative class

Classification Loss

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

- 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

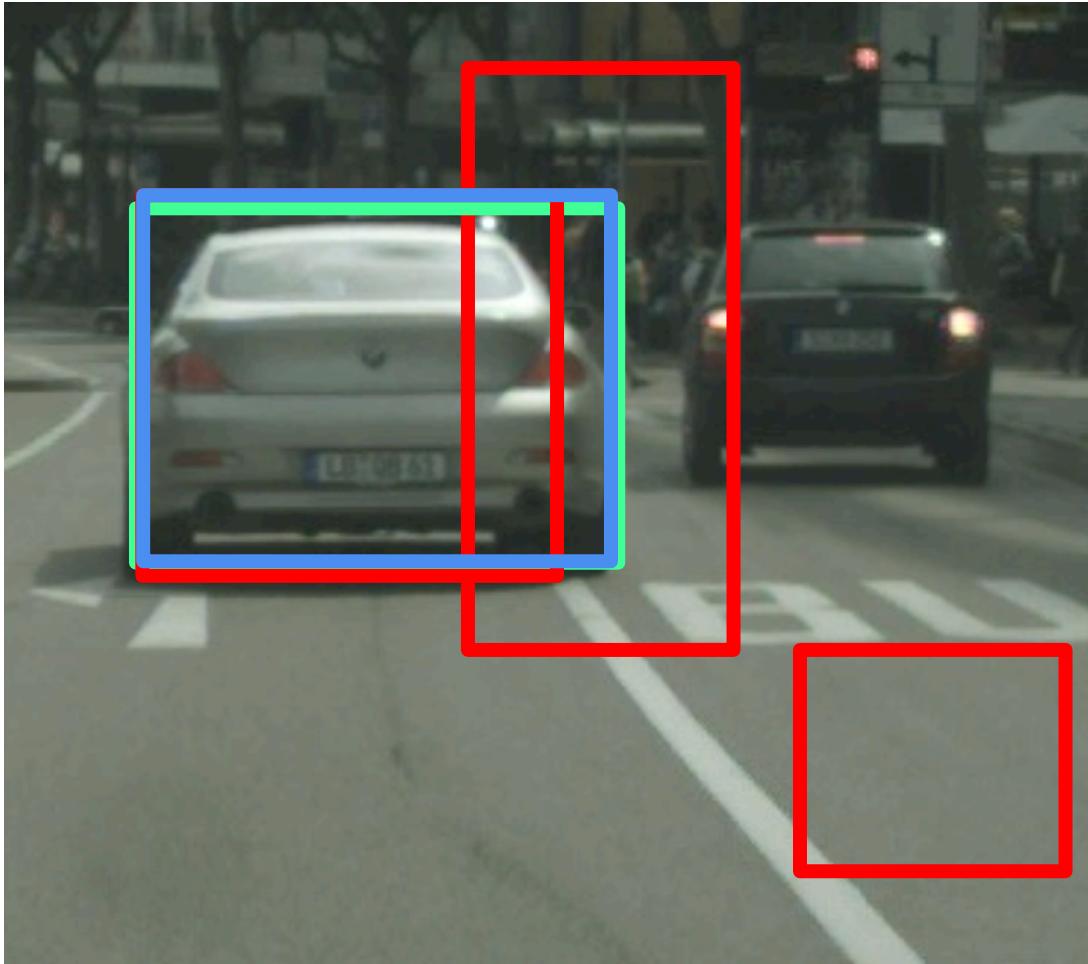
Regression Loss

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

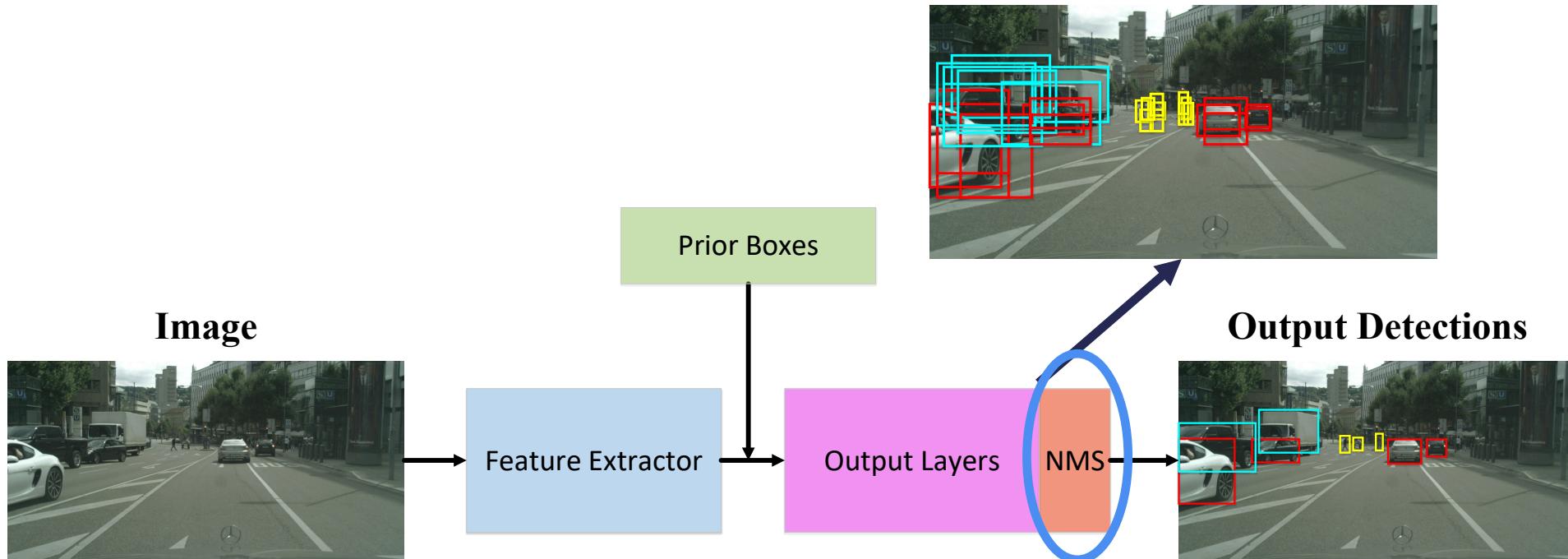
- p_i is 0 if anchor is negative and 1 if anchor is positive
- N_p is the number of positive anchors in the minibatch
- b_i^* is the ground truth bounding box
- b_i is the estimated bounding box, applying the regressed residuals to the anchor box parameters

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

Visual Representation Of Training



Inference Time



Non-Maximum Suppression

Input: *list of bounding box*

$B = \{B_1 \dots B_n\} | B_i = (x_i, y_i, w_i, h_i, s_i) \forall i \in [1, n]$

IOU threshold η

begin

$\bar{B} = Sort(B, s, \downarrow)$

$D = \emptyset$

 for $b \in \bar{B}$ and $\bar{B} \neq \emptyset$ do

$b_{max} = b$

$\bar{B} \leftarrow \bar{B} \setminus b_{max}$ *remove b_{max} from \bar{B}*

$D \leftarrow D \cup b_{max}$

 for $b_i \in \bar{B} \setminus b_{max}$ do *add b_{max} to D*

 if $IOU(b_{max}, b_i) \geq \eta$ then

$\bar{B} \leftarrow \bar{B} \setminus b_i$

 end

 end

 end

Output: D

end

Non-Maximum Suppression



$$\geq \eta = 0.7$$

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

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

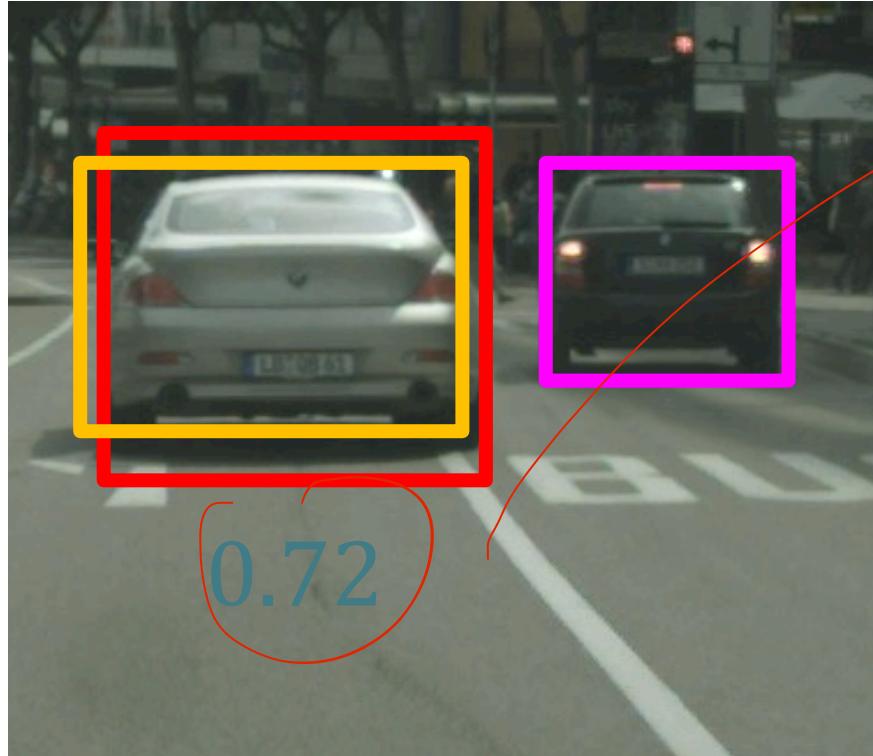
$$S = \{0.98, 0.94, 0.6, 0.45\}$$

$$D = \{ \}$$

remove B_3
from \bar{B}

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

Non-Maximum Suppression



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

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

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

$$D = \{B_1\}$$

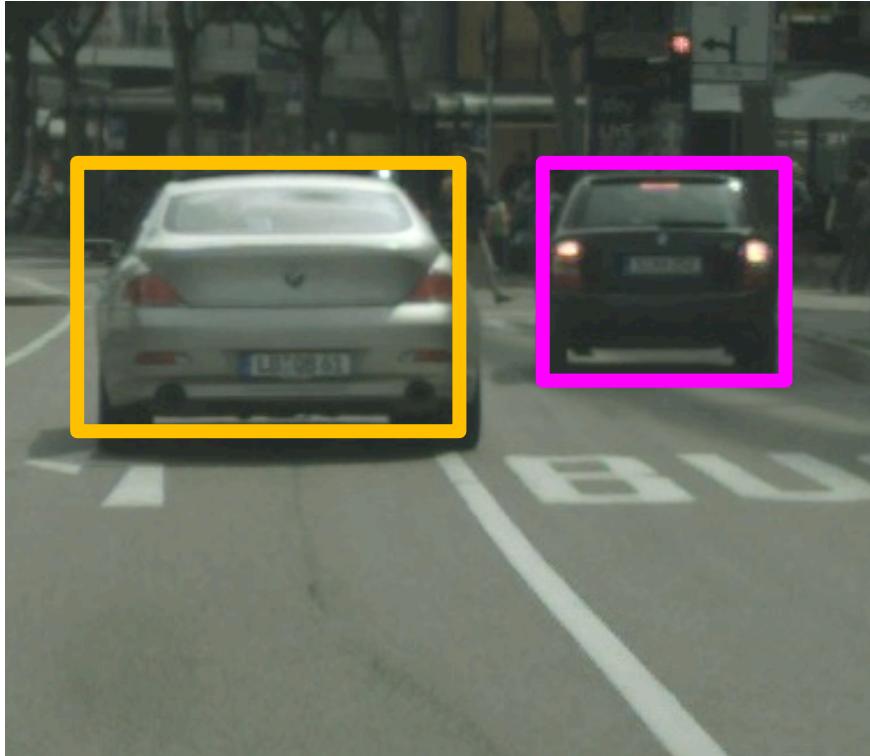
$$\geq \eta = 0.7$$

remove B_4
from \bar{B}

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

Non-Maximum Suppression

$$\eta = 0.7$$



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

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

$$S = \{\}$$

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

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**