2D Object Detection With Convolutional Neural Networks

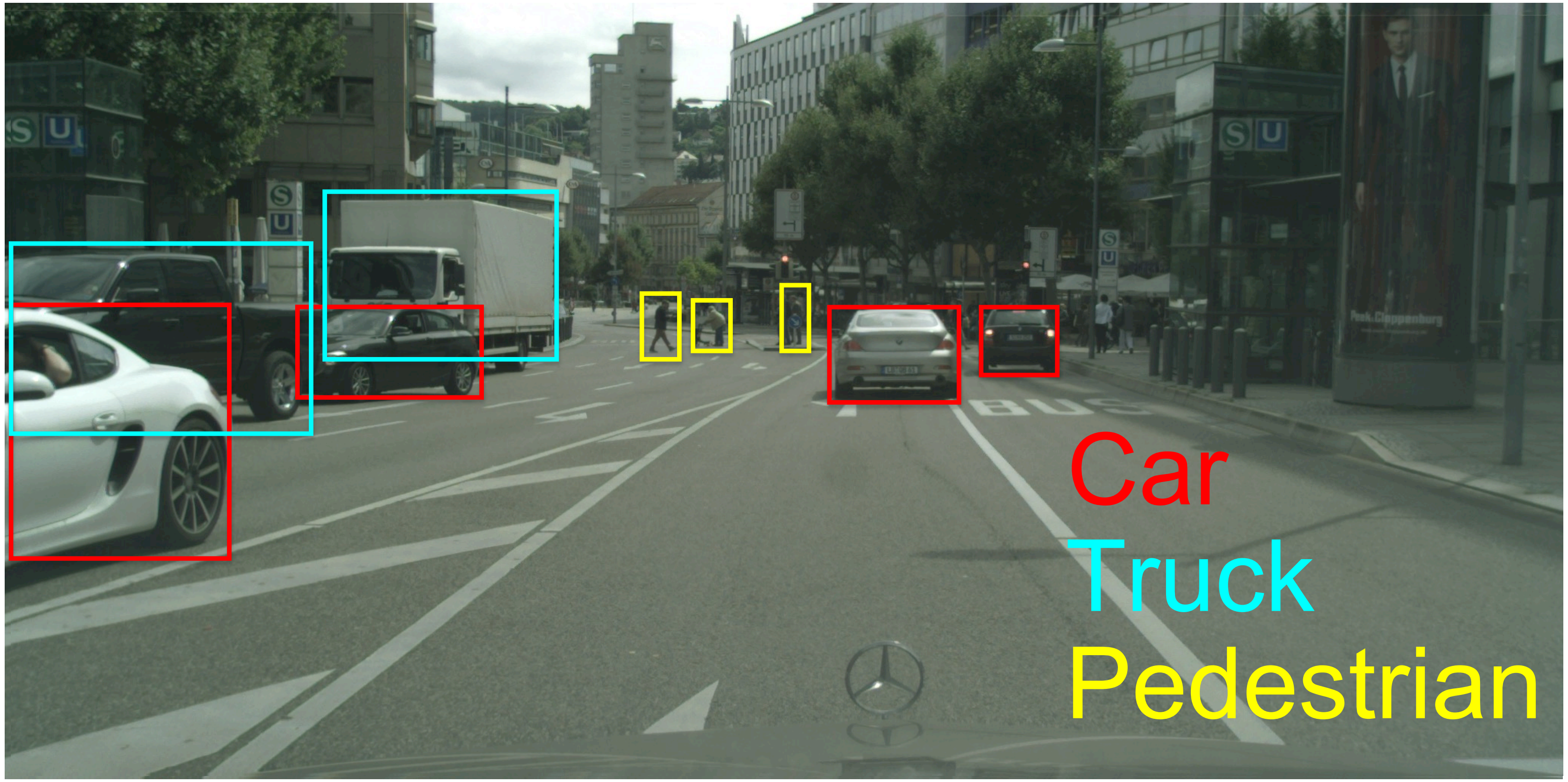
Course 3, Module 4, Lesson 2

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

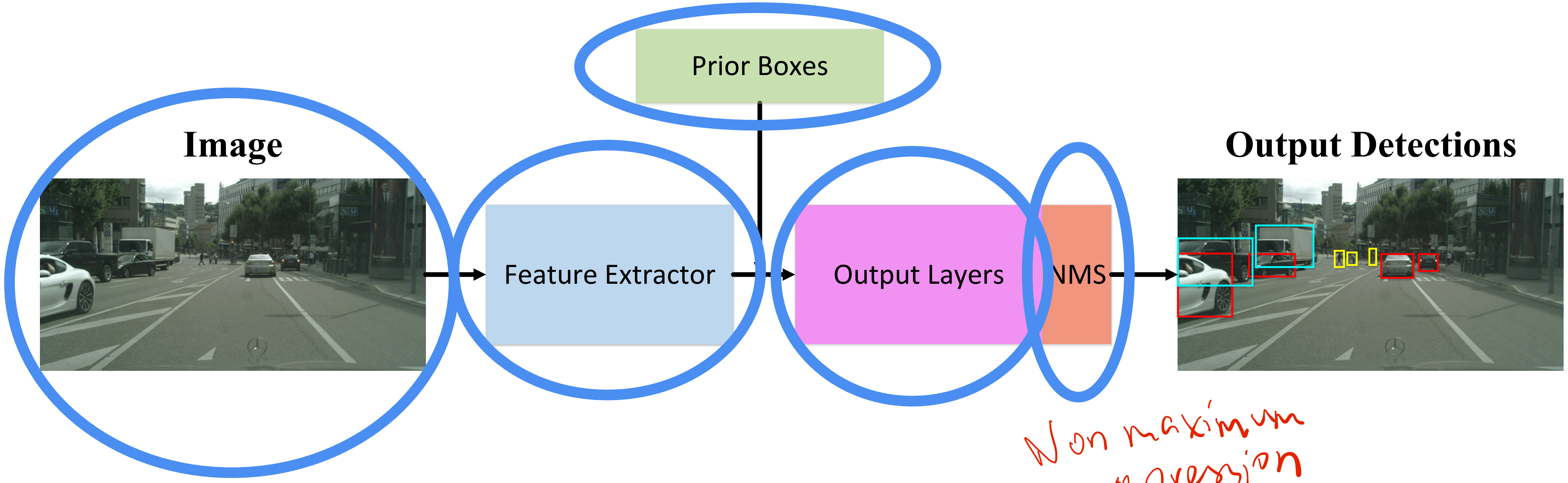
Learning Objectives

- Learn to build standard single stage architecture for 2D object detection
- Learn common neural network design choices for performing 2D object detection using the proposed architecture

The Object Detection Problem

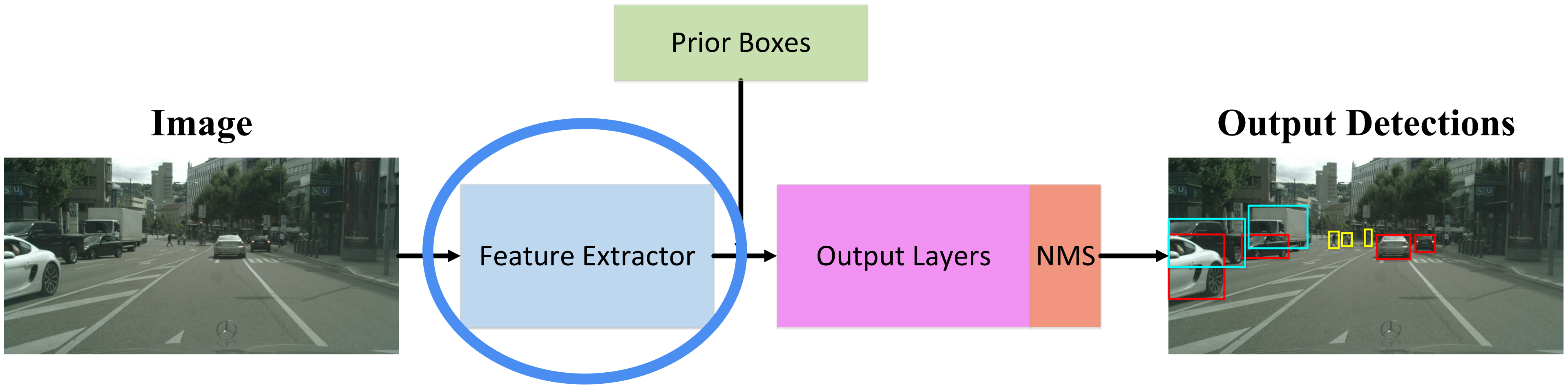


ConvNets For 2D Object Detection



Non maximum suppression

The Feature Extractor



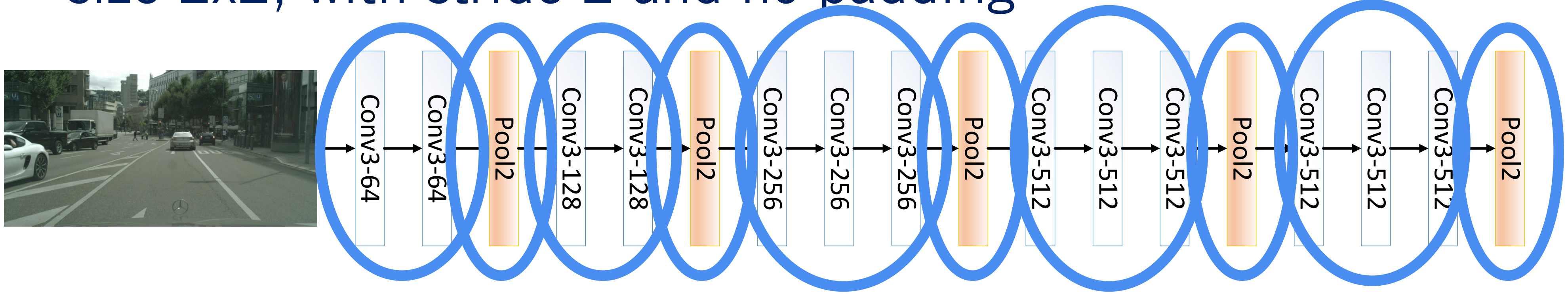
The Feature Extractor

- Feature extractors are the most computationally expensive component of the 2D object detector
- The output of feature extractors usually has much **lower width and height** than those of the input image, but much **greater depth**
- Very active area of research, with new extractors proposed on regular basis
- **Most common extractors are**: VGG, ResNet, and Inception

90% computation

VGG Feature Extractor

- Alternating convolutional and pooling layers
- All convolutional layers are of size 3x3xK, with stride 1 and 1 zero-padding
- All pooling layers use the **max** function, and are of size 2x2, with stride 2 and no padding



VGG Feature Extractor

- Conv output shape:
  - $W_{out} = \frac{W_{in} - m + 2 \times P}{S} + 1$
  - $H_{out} = \frac{H_{in} - m + 2 \times P}{S} + 1$
  - $D_{out} = K$
- All convolution layers are of size 3x3xK, with stride 1 and 1 zero-padding
  - $\boldsymbol{W_{out}} = \frac{W_{in} - m + 2 \times P}{S} + 1 = \frac{H_{in} - 3 + 2 \times 1}{1} + 1 = \boldsymbol{W_{in}}$
  - $\boldsymbol{H_{out}} = \frac{H_{in} - m + 2 \times P}{S} + 1 = \frac{H_{in} - 3 + 2 \times 1}{1} + 1 = \boldsymbol{H_{in}}$
  - $\boldsymbol{D_{out}} = K$
- All pooling layers use the max function, and are of size 2x2, with stride 2 and no padding.
  - $\boldsymbol{W_{out}} = \frac{W_{in} - m}{S} + 1 = \frac{W_{in} - 2}{2} + 1 = \frac{\boldsymbol{W_{in}}}{\boldsymbol{2}}$
  - $\boldsymbol{H_{out}} = \frac{H_{in} - m}{S} + 1 = \frac{H_{in} - 2}{2} + 1 = \frac{\boldsymbol{H_{in}}}{\boldsymbol{2}}$
  - $\boldsymbol{D_{out}} = \boldsymbol{D_{in}}$

The Feature Extractor

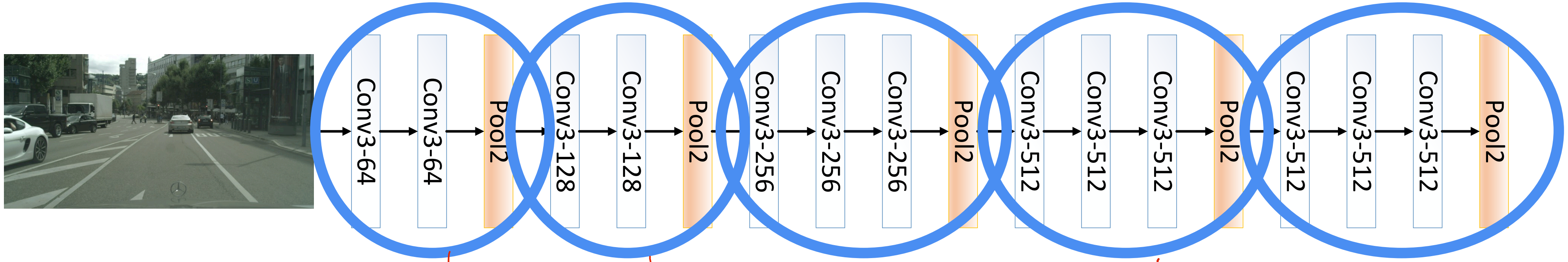
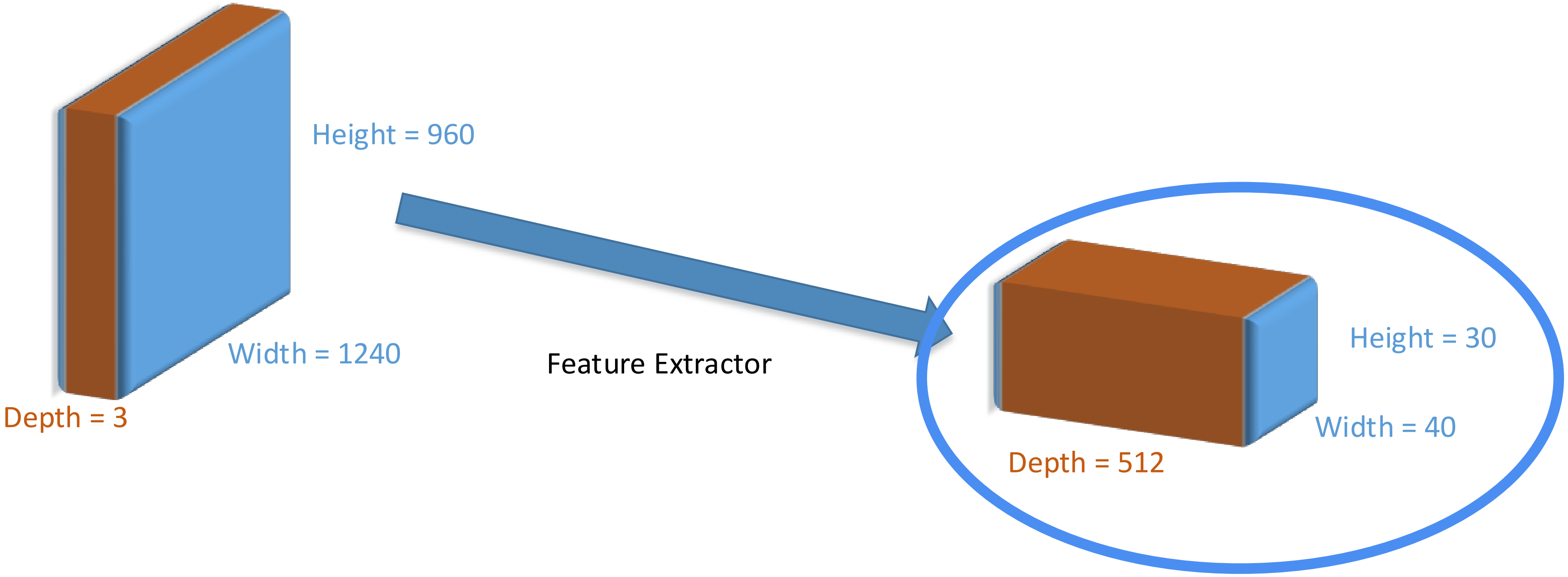


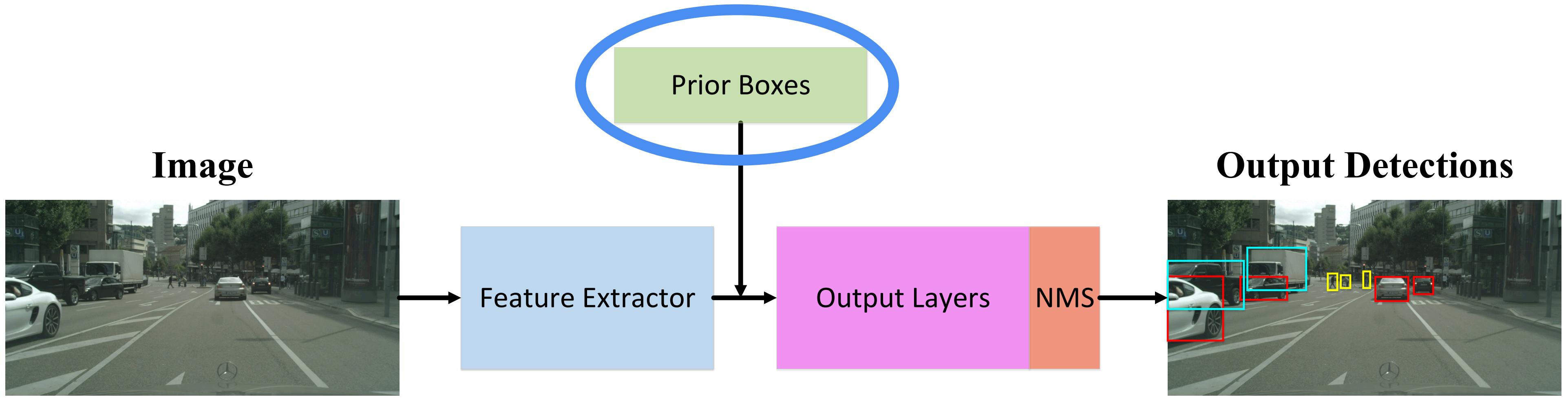
	Image	Conv1	Conv2	Conv3	Conv4	Conv5
Width	M	M/2	M/4	M/8	M/16	M/32
Height	N	N/2	N/4	N/8	N/16	N/32
Depth	3	64	128	256	512	512

$M \times N \times 3$

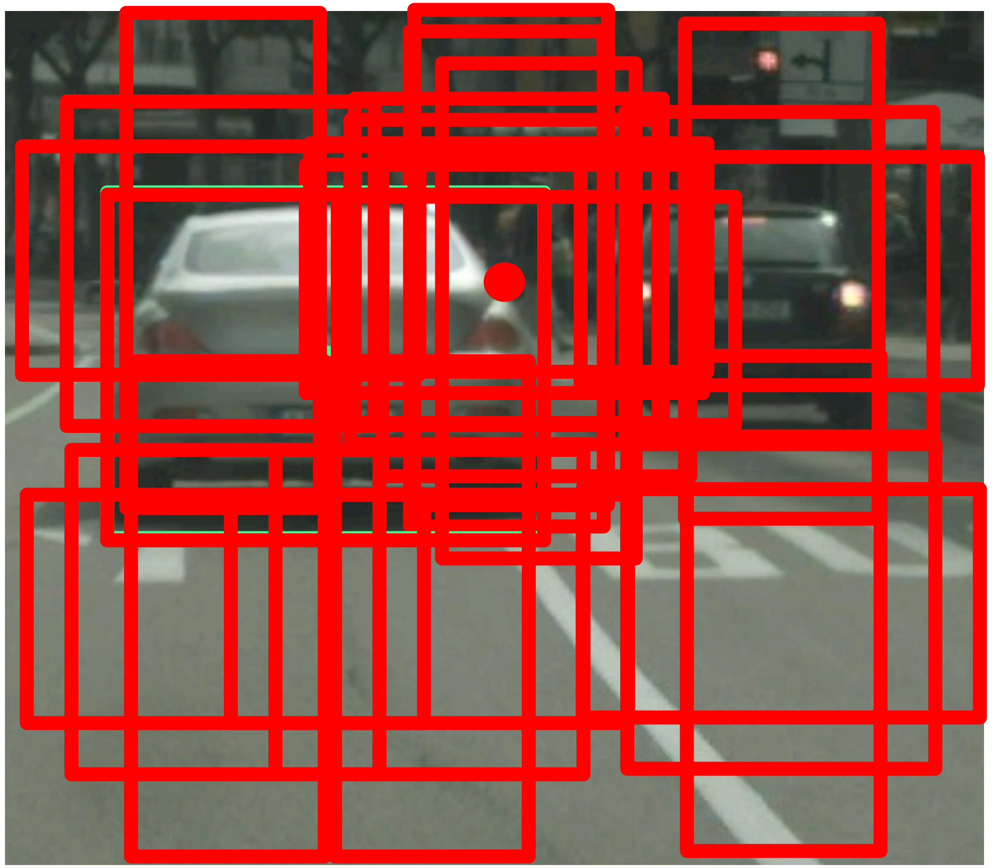
Output Volume Shape



Prior/Anchor Bounding Boxes

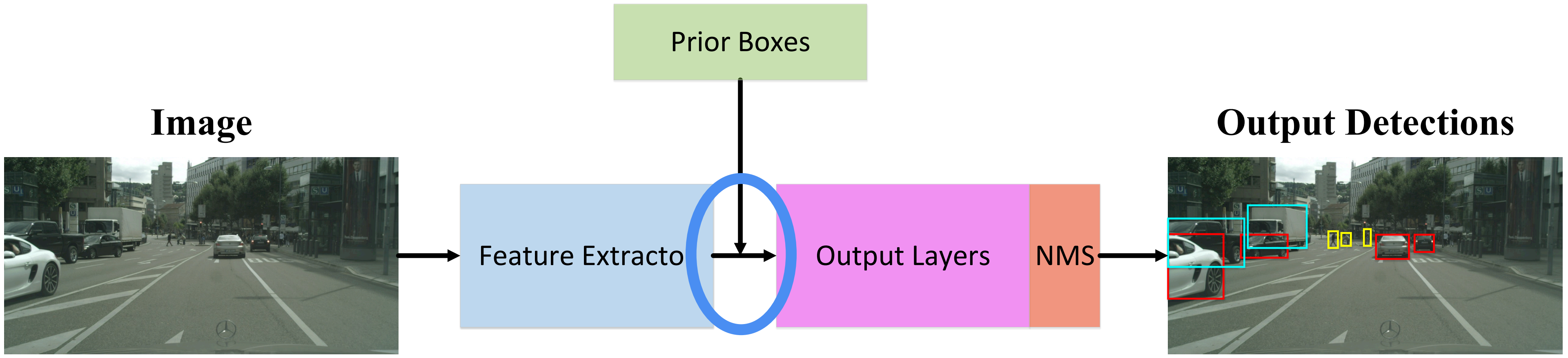


Prior/Anchor Bounding Boxes



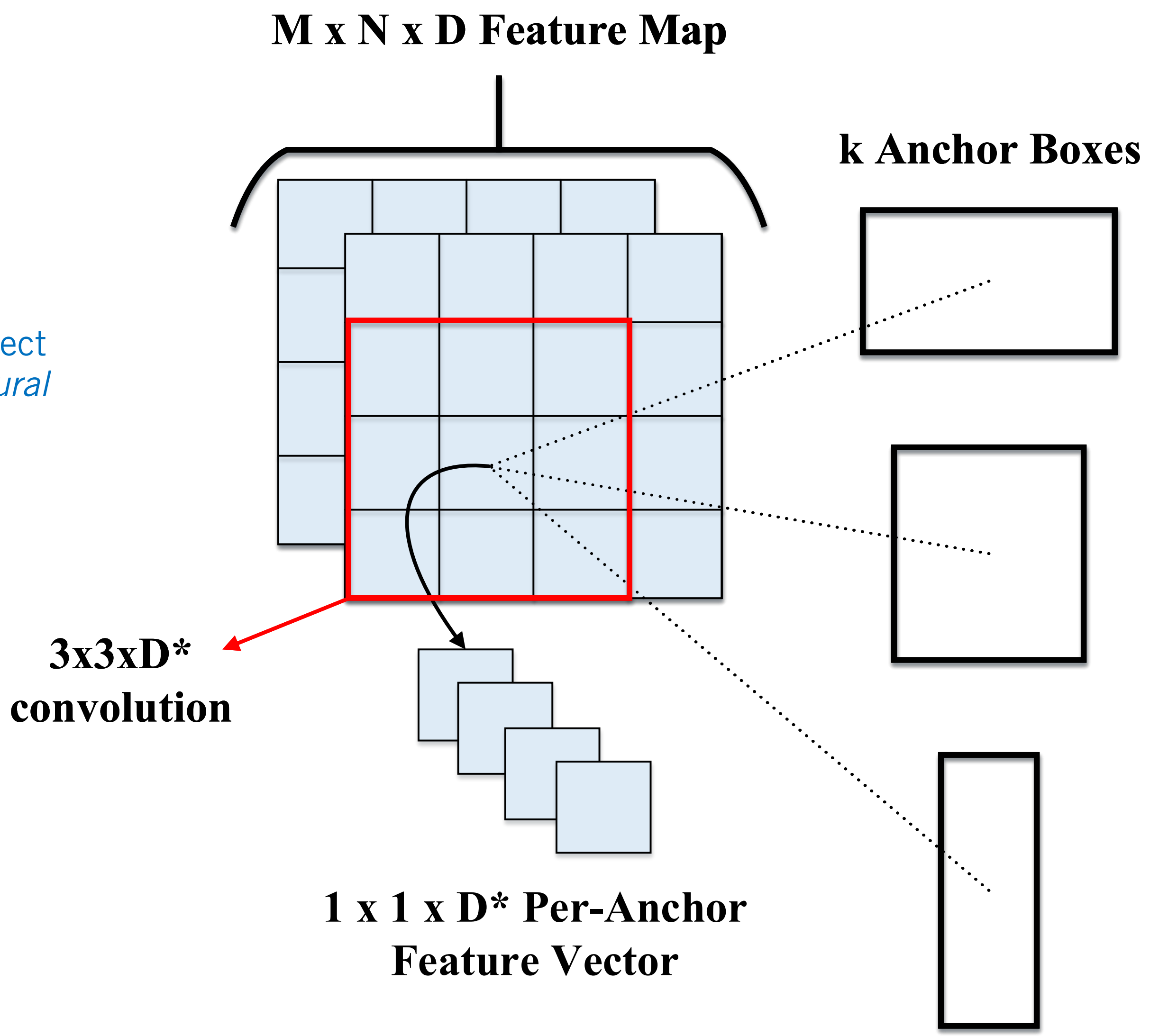
residual learning

Prior/Anchor Bounding Boxes

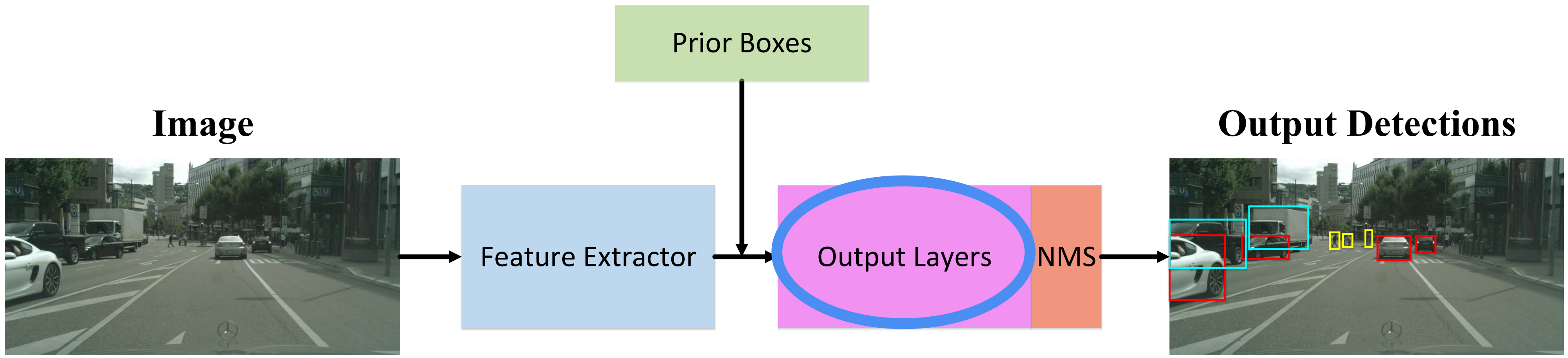


Using Anchor Boxes

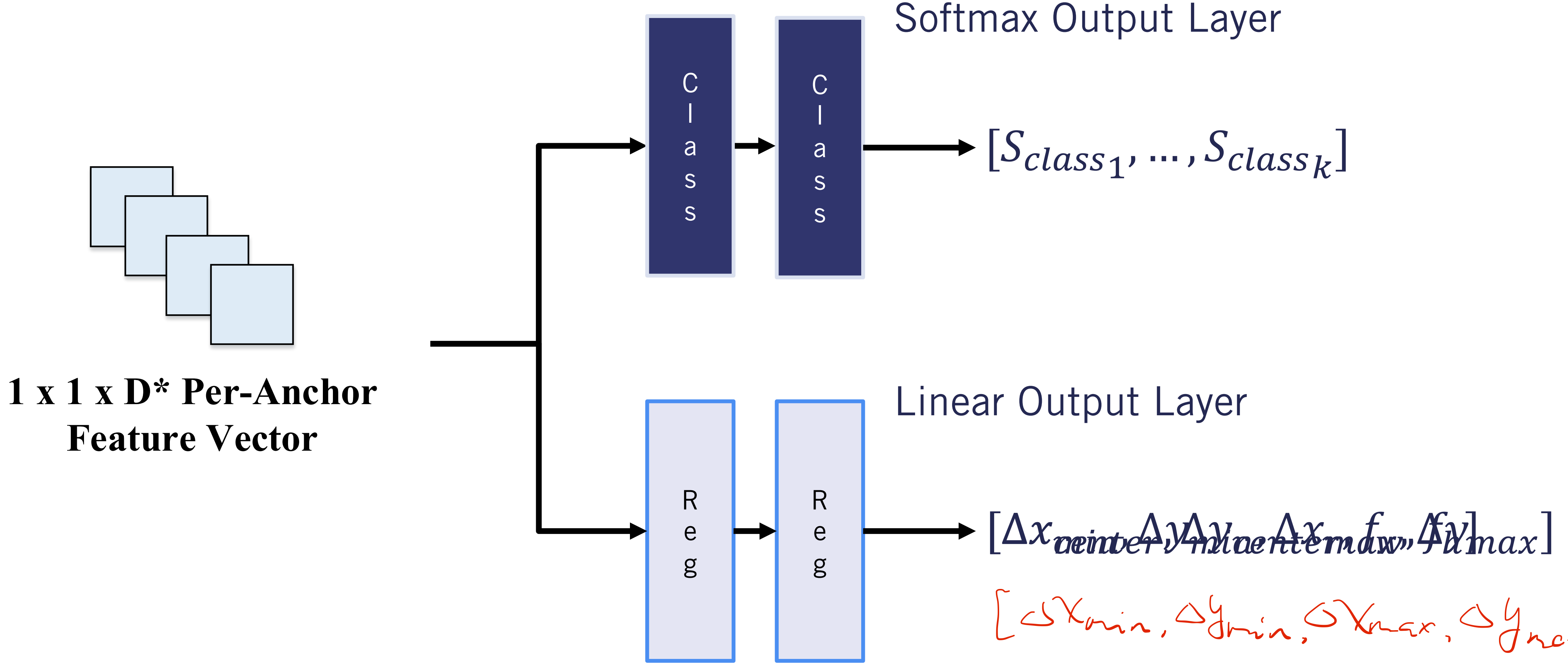
Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015



Output Layers



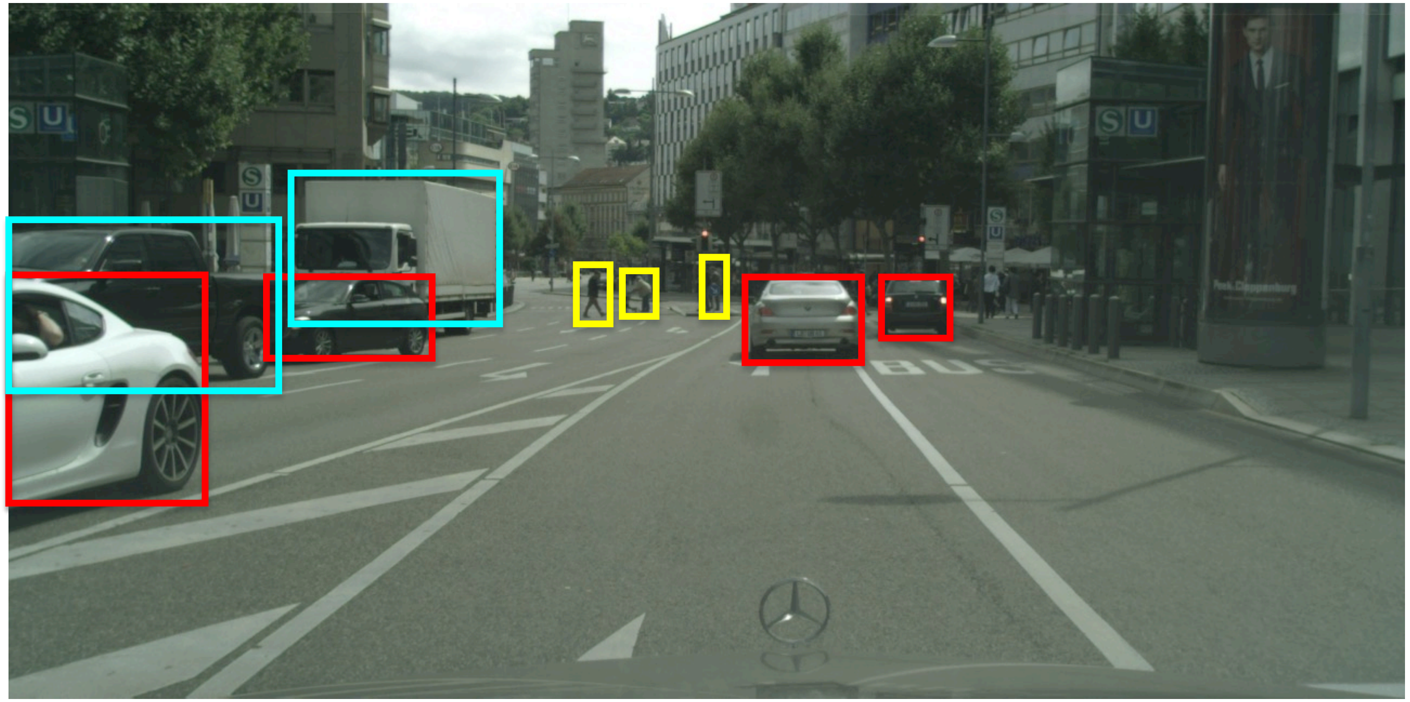
Classification VS Regression Heads



$[\Delta x_{min}, \Delta y_{min}, \Delta x_{max}, \Delta y_{max}]$

Or. $[\Delta x_{center}, \Delta y_{center}, f_w, f_h]$

Output handling



Summary

- 2D object detectors can be performed using convolutional neural networks
- Usually, anchor boxes are used as priors for the neural network to shift around to achieve object classification and localization

- **Next: Training vs Inference**