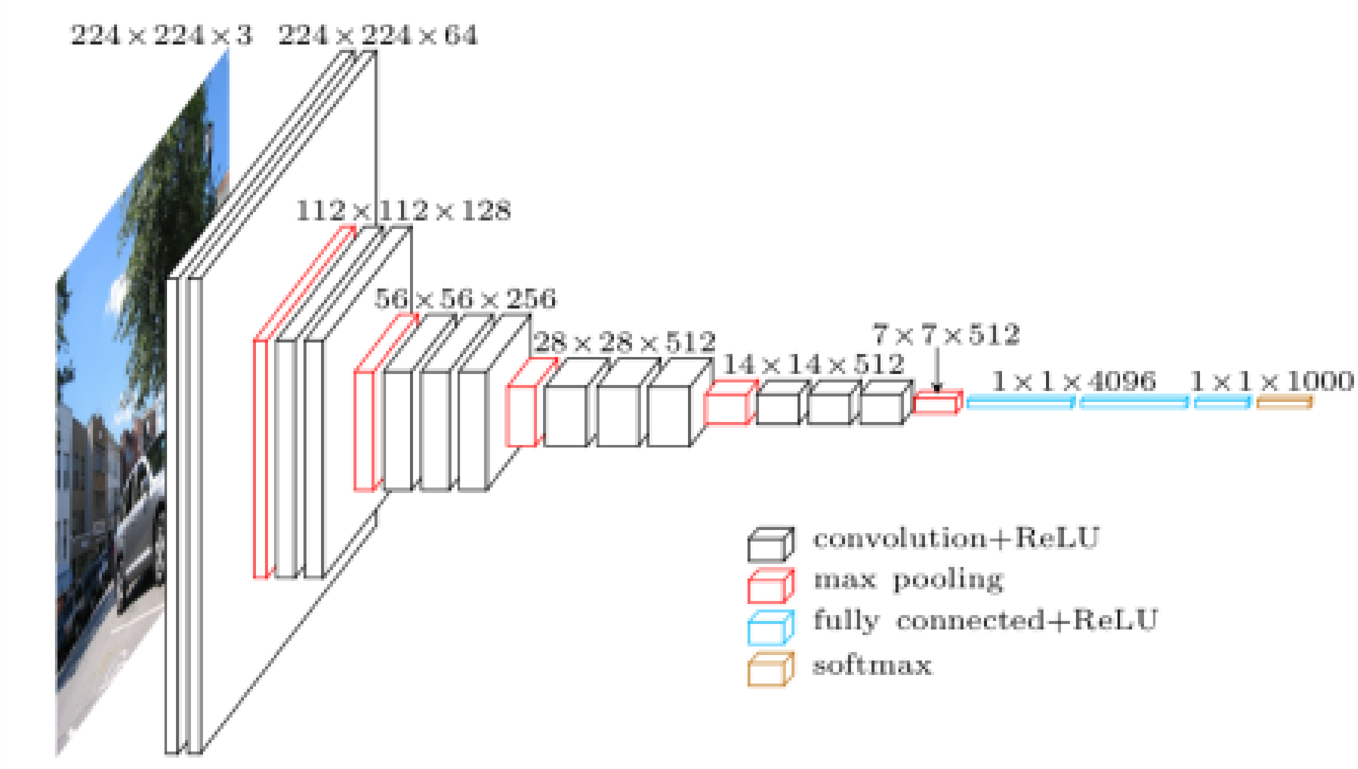
DL2.0 Bootcamp Object Detection

By Kingsley Kuan

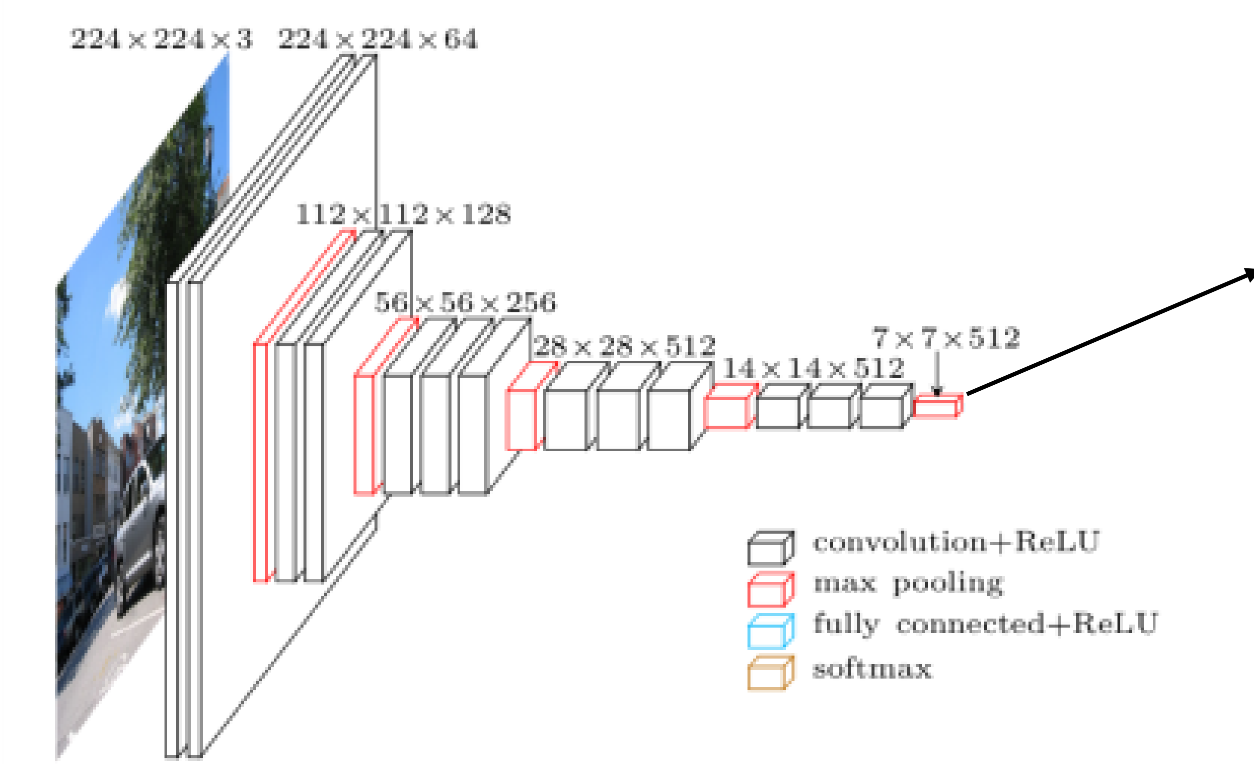
CNNs for Image Classification

- Can we reuse image classification CNNs for object detection?



Repurposing a CNN for Object Detection

- What if we chop the last layers off an image classification CNN?

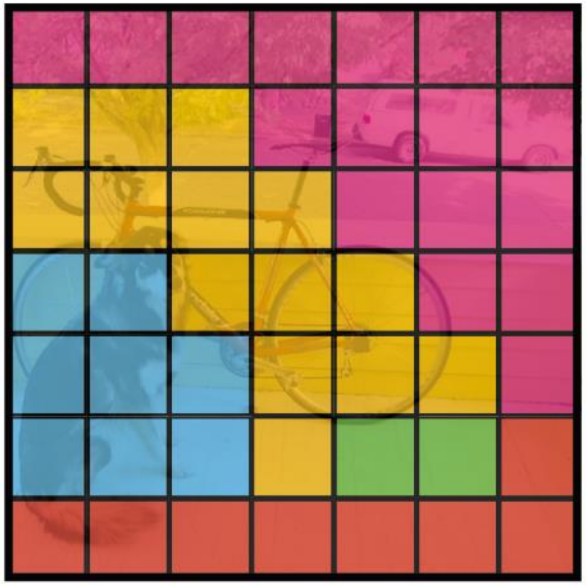


0	1	0	0	1	1	0
1	1	0	1	0	0	0
1	1	1	1	1	0	0
0	1	1	0	1	0	1
1	0	0	0	0	1	1
0	0	1	1	0	1	0
0	0	1	1	1	1	0

- **Problem:** This feature map is implicitly learned during training

CNNs for Object Detection - Object Classes

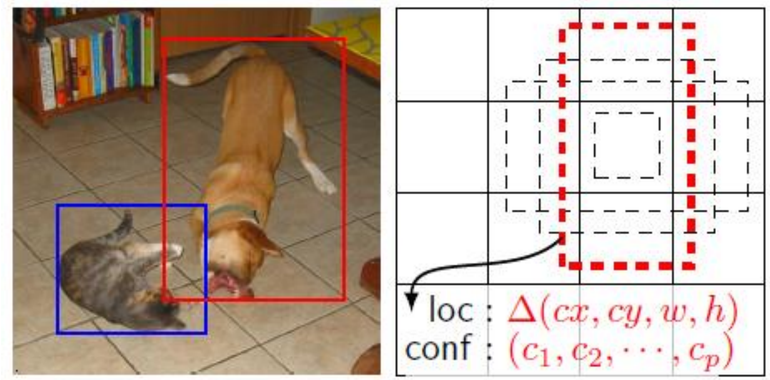
- **Solution:** Explicitly redefine the target output during training
- Groundtruth consists of a (m x n) score map where each feature map cell contains (1, ..., c) class scores
- Output of network becomes a (m x n x c) feature map
- Ie. Network predicts score of an object class occurring at the associated position in the image
- Use standard cross entropy loss for each feature map cell



CNNs for Object Detection - Object Bounding Box

- Define additional outputs to refine the shape of the object's bounding box by:

1. Adding more default bounding box sizes to each feature map cell
   - These are known as anchors or default boxes in different frameworks



2. Compute regression targets relative to default boxes that closely match groundtruth bounding boxes

$$t_{\mathrm{x}} = (x - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}} = (y - y_{\mathrm{a}})/h_{\mathrm{a}},$$
$$t_{\mathrm{w}} = \log(w/w_{\mathrm{a}}), \quad t_{\mathrm{h}} = \log(h/h_{\mathrm{a}}),$$

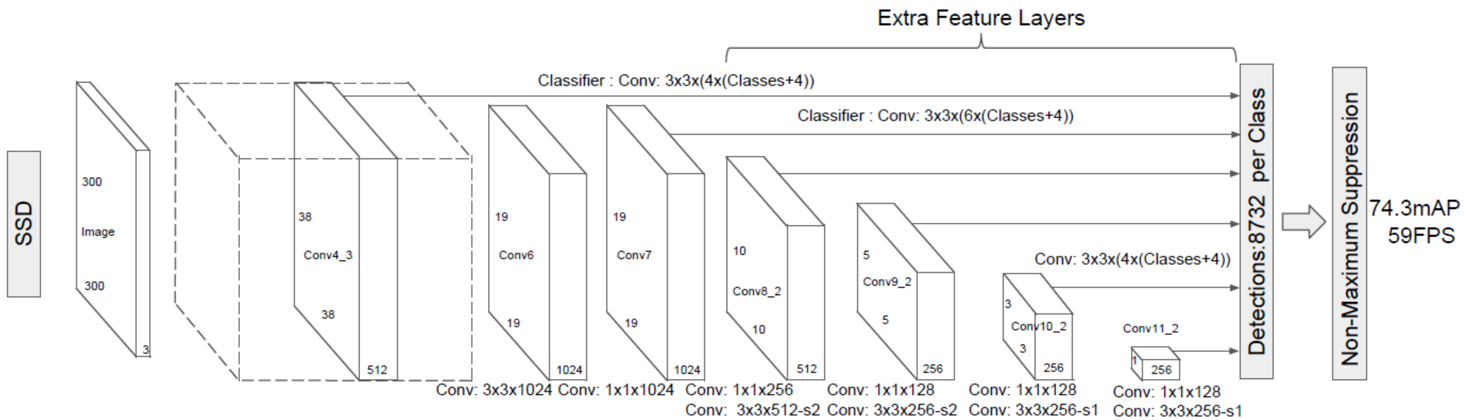
Output of network becomes a (m x n x k x (c + 4)) feature map

Use smooth L1 loss to optimize regression

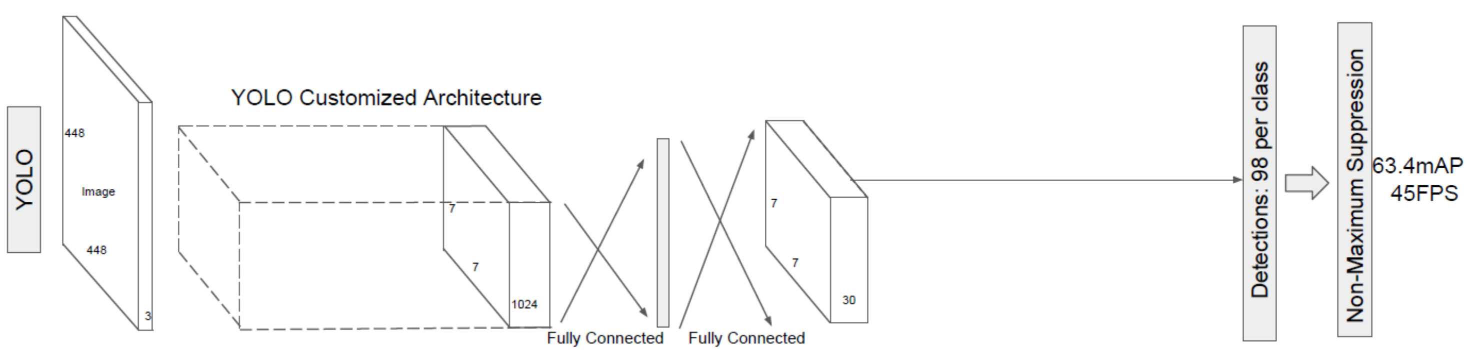
$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Frameworks I - Speed over Performance

- SSD - Produces outputs directly from feature maps of different scales

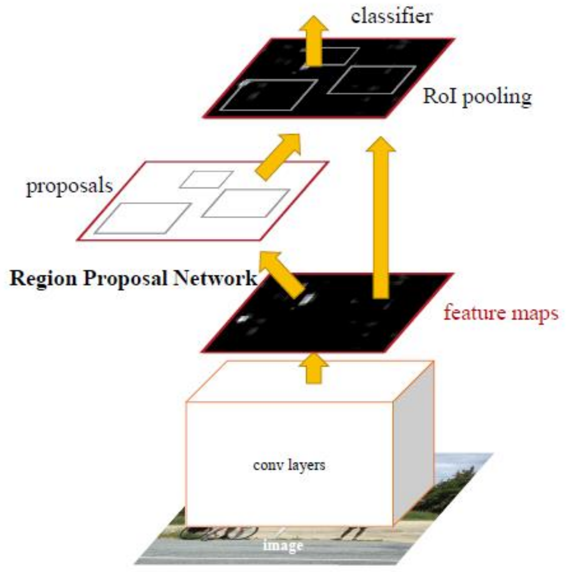


- YOLO - Uses fully connected layer before final output feature map

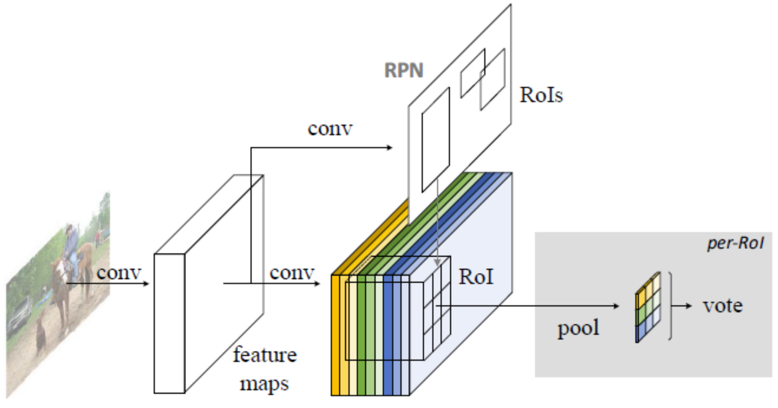


Frameworks II - Performance over Speed

- Faster-RCNN - Network branches into two:
  - Region proposal network proposes ROIs with 2 classes (object / no object)
  - Classifier layers classifies features cropped and scaled to a fixed size using proposals

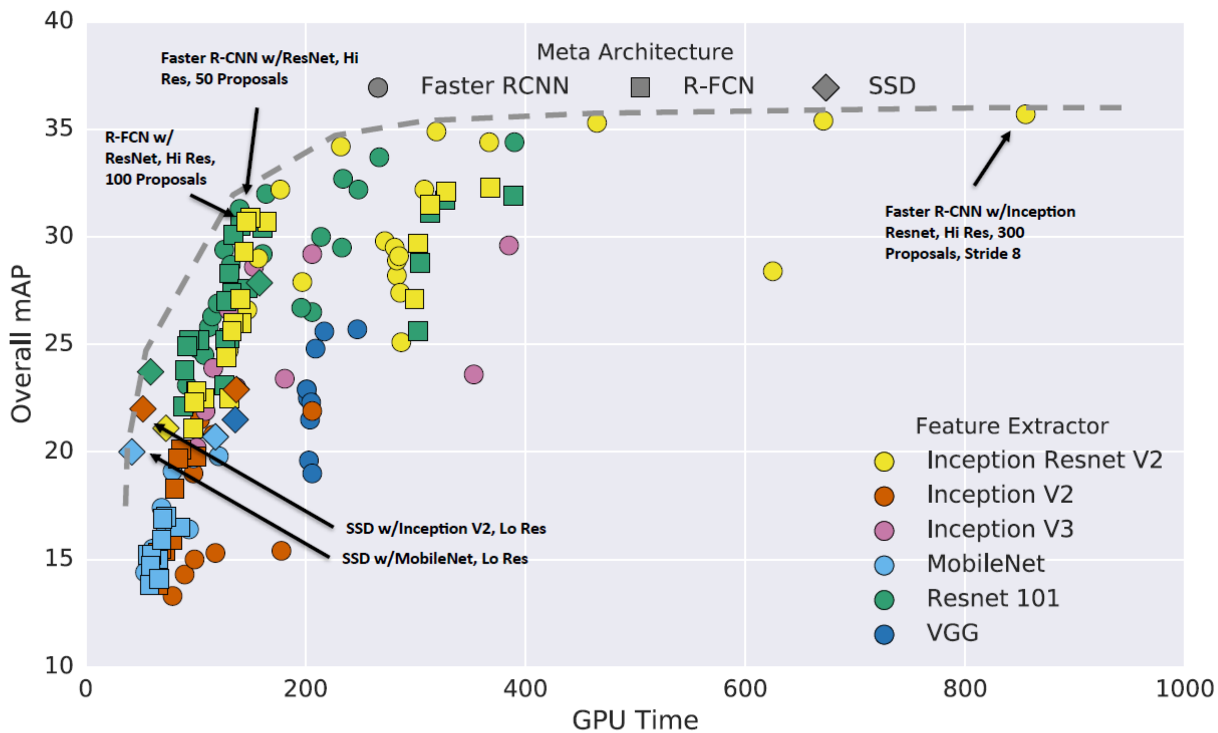


- R-FCN - Uses region proposal sub-network but only uses it to pool from position sensitive output feature map



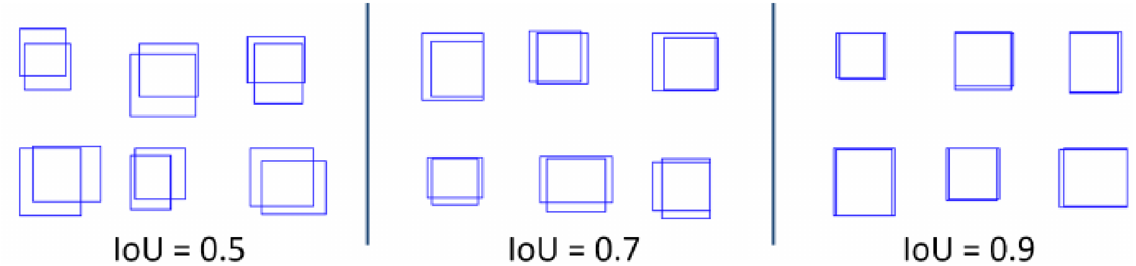
Frameworks III

- Use any image classification CNN as base of the framework
- Mix and match different base CNNs and frameworks for speed-accuracy trade-offs



Additional Techniques

- Anchors / Default Boxes can be manually defined or discovered through clustering groundtruth bounding boxes
- Significant imbalance between negative and positive feature map cells can be addressed through proper sampling during training
- Match between two boxes can be computed with intersection over union (iou)



Code Walkthrough

Applying object detection to Kitti dataset (autonomous driving images)

Code is based on a very simplified version of SSD+ResNet-18 with only one default box per feature map cell and one feature map scale

References

- Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016. https://arxiv.org/pdf/1512.02325.pdf
- Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. https://arxiv.org/pdf/1506.02640.pdf
- Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015. https://arxiv.org/pdf/1506.01497.pdf
- Li, Yi, Kaiming He, and Jian Sun. "R-fcn: Object detection via region-based fully convolutional networks." Advances in Neural Information Processing Systems. 2016. https://arxiv.org/pdf/1605.06409.pdf
- Huang, Jonathan, et al. "Speed/accuracy trade-offs for modern convolutional object detectors." arXiv preprint arXiv:1611.10012 (2016). https://arxiv.org/pdf/1611.10012