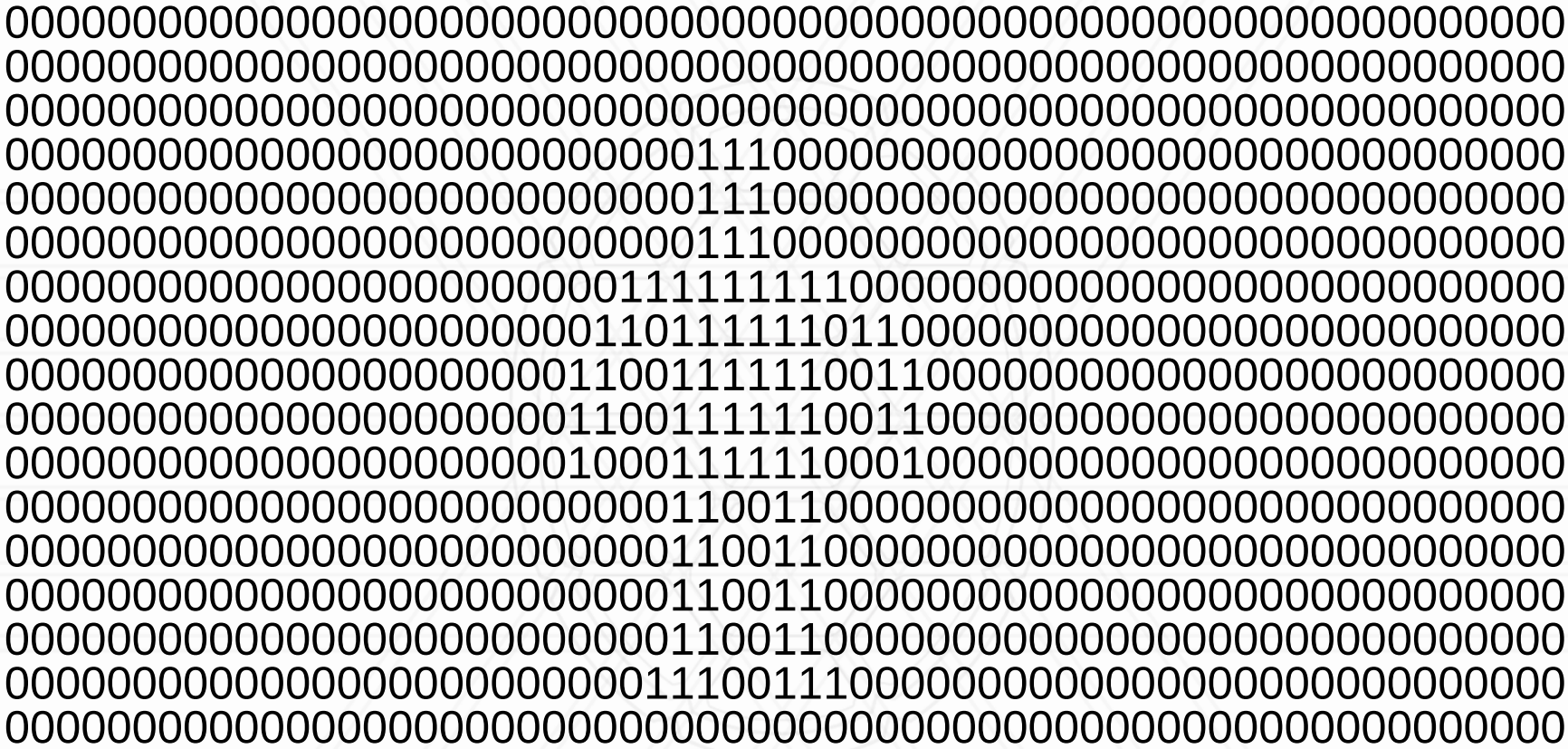
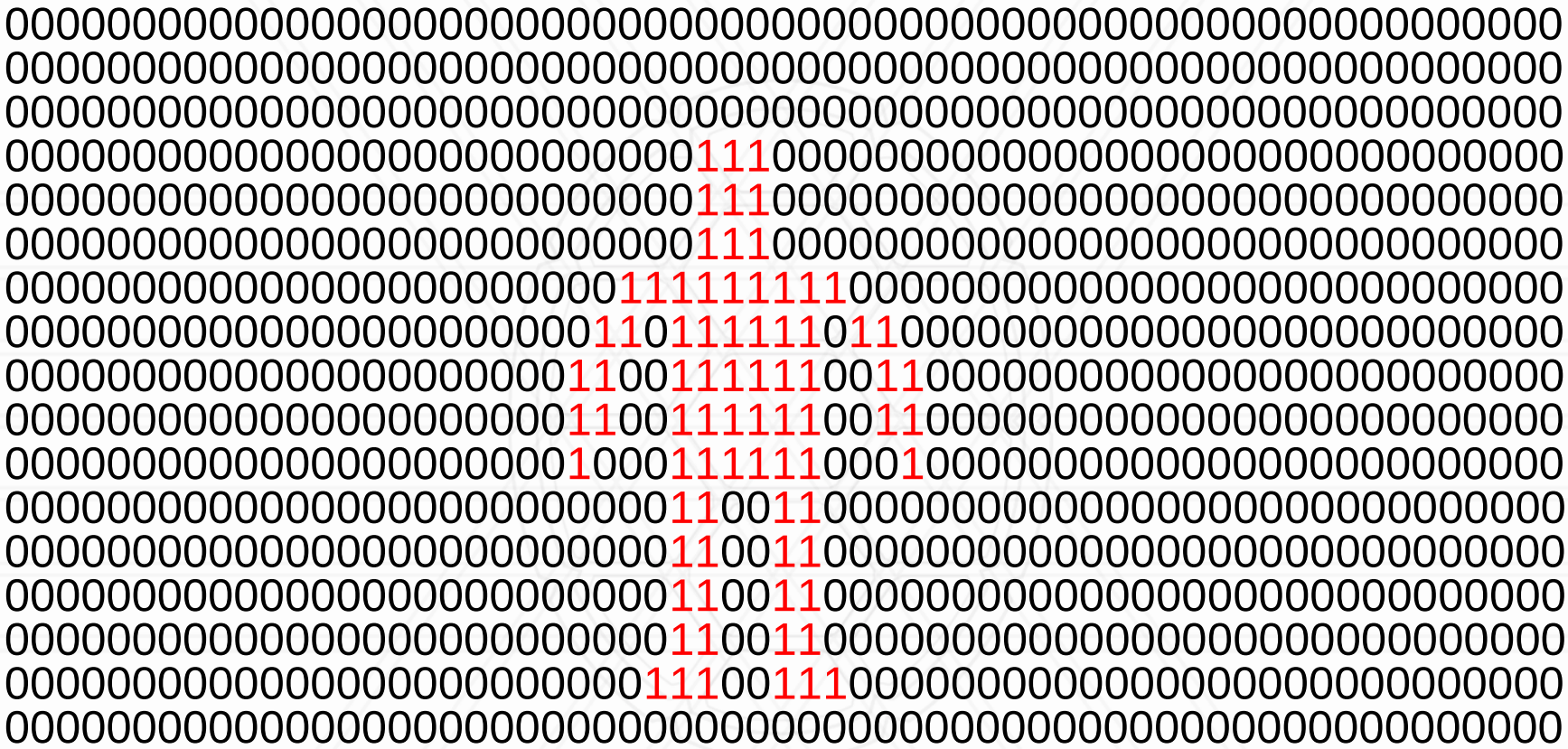
RGB-D Fusion for Real Time Object Detection

Chayan Patodi, Nikhil Mehra and Raghav Nandwani

UNIVERSITY OF
MARYLAND




0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000111000000000000000000000000000000000000
0000000000000000000000000111000000000000000000000000000000000000
0000000000000000000000000111000000000000000000000000000000000000
0000000000000000000000001111111100000000000000000000000000000000
0000000000000000000000011011111101100000000000000000000000000000
0000000000000000000000011001111110011000000000000000000000000000
0000000000000000000000011001111110011000000000000000000000000000
0000000000000000000000010001111110001000000000000000000000000000
0000000000000000000000000011001100000000000000000000000000000000
0000000000000000000000000011001100000000000000000000000000000000
0000000000000000000000000011001100000000000000000000000000000000
0000000000000000000000000011001100000000000000000000000000000000
0000000000000000000000000011100111000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000


1369.8 FPS

Motivation

Exploring RGB+Depth Fusion for Real-Time Object Detection

By: Tanguy Ophoff, Kristof Van Beeck and Toon Goedemé

Objective: Whether fusion of depth data with the RGB data can help increase the performance of current state of the art single shot networks.

Model Architecture

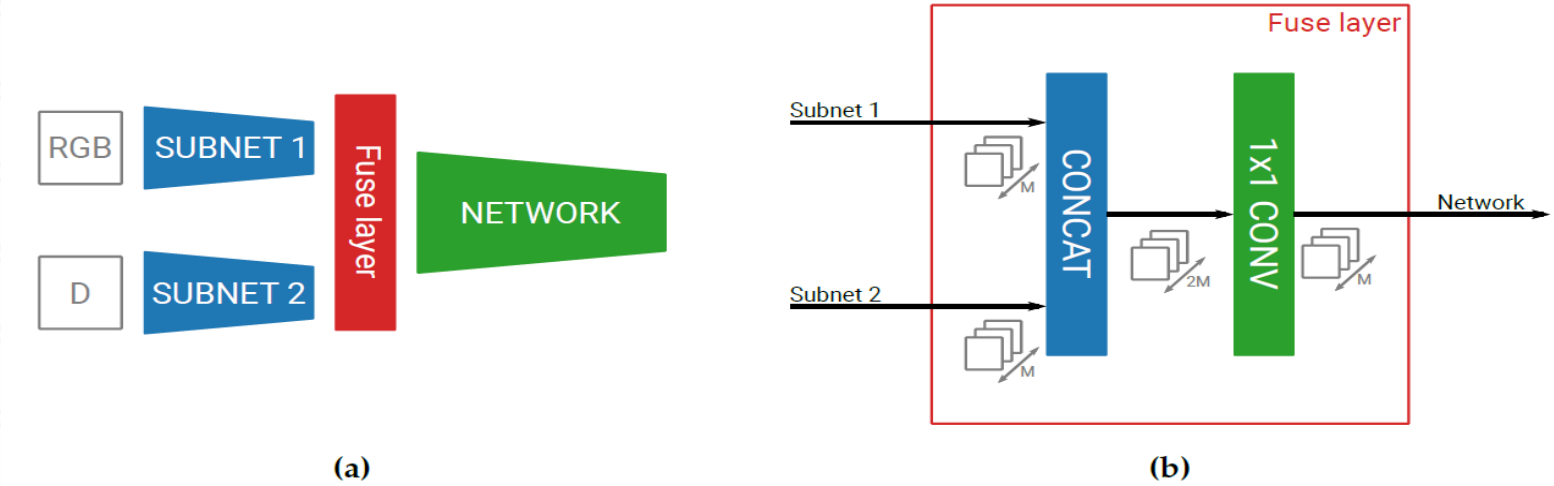


Figure 1. The main building blocks of our parameterizable fusion network. (**a**) The fuse layer can be transparently implemented after any arbitrary layer, allowing for a parameterizable fusion level. (**b**) The fuse layer combines both information streams and divides the number of output channels by two.

Network training

- Used ImageNet pretrained weights instead of random initialization
- Same weights for depth network, just removed the weights of first layer

Network training

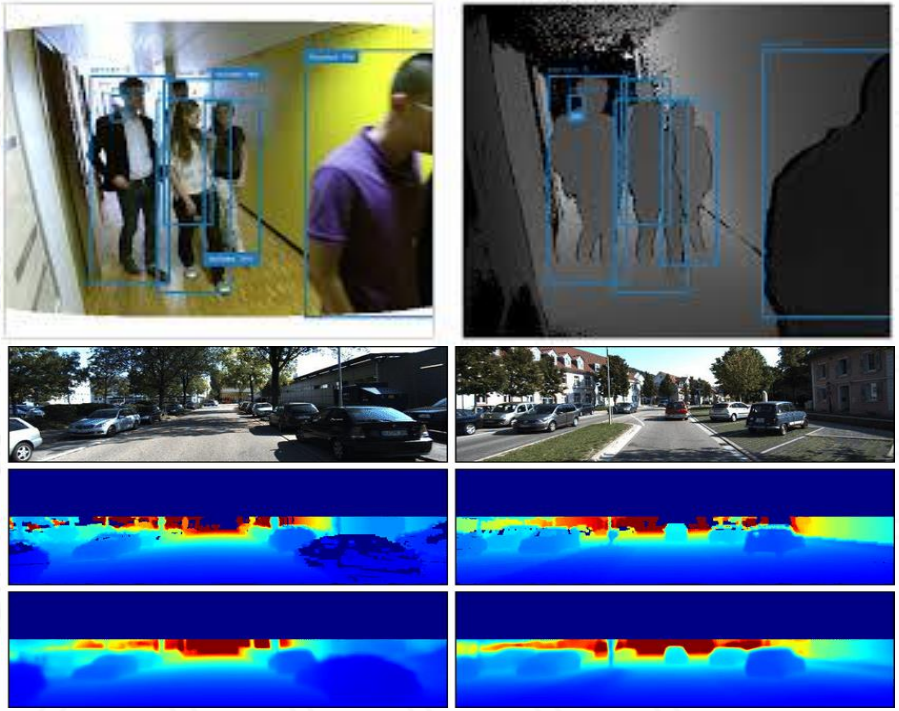
- Used ImageNet pretrained weights instead of random initialization
- Same weights for depth network, just removed the weights of first layer

Why ??

- The networks looks for similar features in both RGB and Depth images
- If the depth subnetwork does not provide any substantial information compared to the RGB network, the fusion layer could possibly ignore those feature maps.

Dataset

- EPFL pedestrian depth dataset
- KITTI depth map



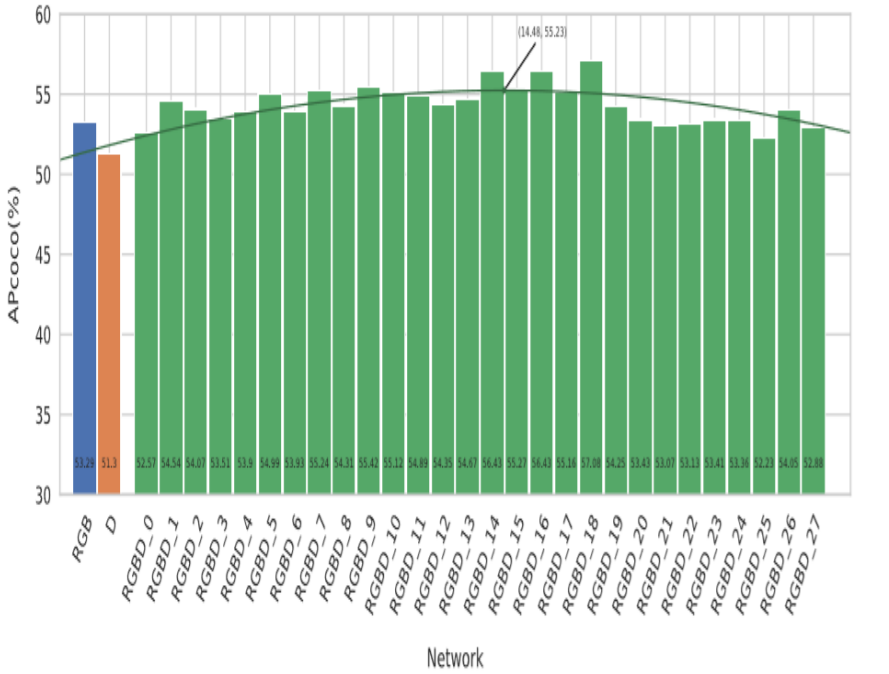
Evaluation

The main advantages of fusing depth data is in that the clearly distinguishable **silhouettes** in the **depth maps** allow for more accurate bounding boxes

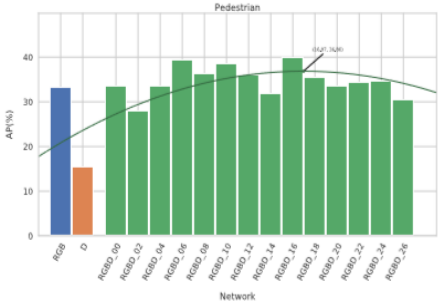
AP of the networks was measured using COCO IoU thresholding scheme, which is defined as follows.

$$AP = \frac{\sum_{IoU \in I} AP_{IoU}(Annotations, Detections)}{I} ; I = \{0.50, 0.55, 0.60, \ldots, 0.95\}$$

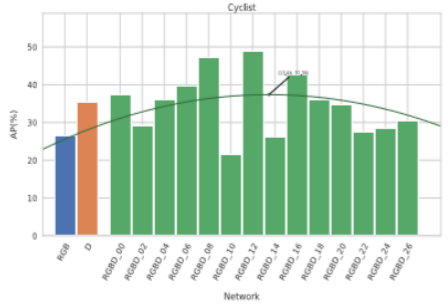
Results


APcoco(%)
60
55
50
45
40
35
30
(14.48, 55.23)
53.29 51.3 52.57 54.54 54.07 53.51 53.9 54.99 53.93 55.24 54.31 55.42 55.12 54.89 54.35 54.67 56.43 55.27 56.43 55.16 57.08 54.25 53.43 53.07 53.13 53.41 53.36 52.23 54.05 52.88
RGB
D
RGBD_0
RGBD_1
RGBD_2
RGBD_3
RGBD_4
RGBD_5
RGBD_6
RGBD_7
RGBD_8
RGBD_9
RGBD_10
RGBD_11
RGBD_12
RGBD_13
RGBD_14
RGBD_15
RGBD_16
RGBD_17
RGBD_18
RGBD_19
RGBD_20
RGBD_21
RGBD_22
RGBD_23
RGBD_24
RGBD_25
RGBD_26
RGBD_27
Network

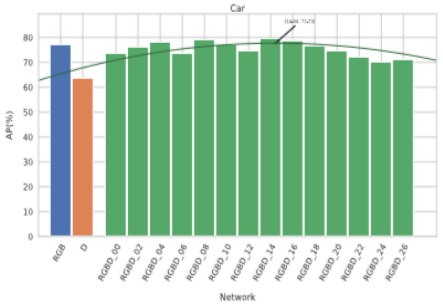
Results Continued



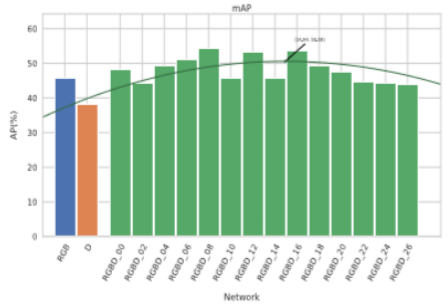
(a)



(b)

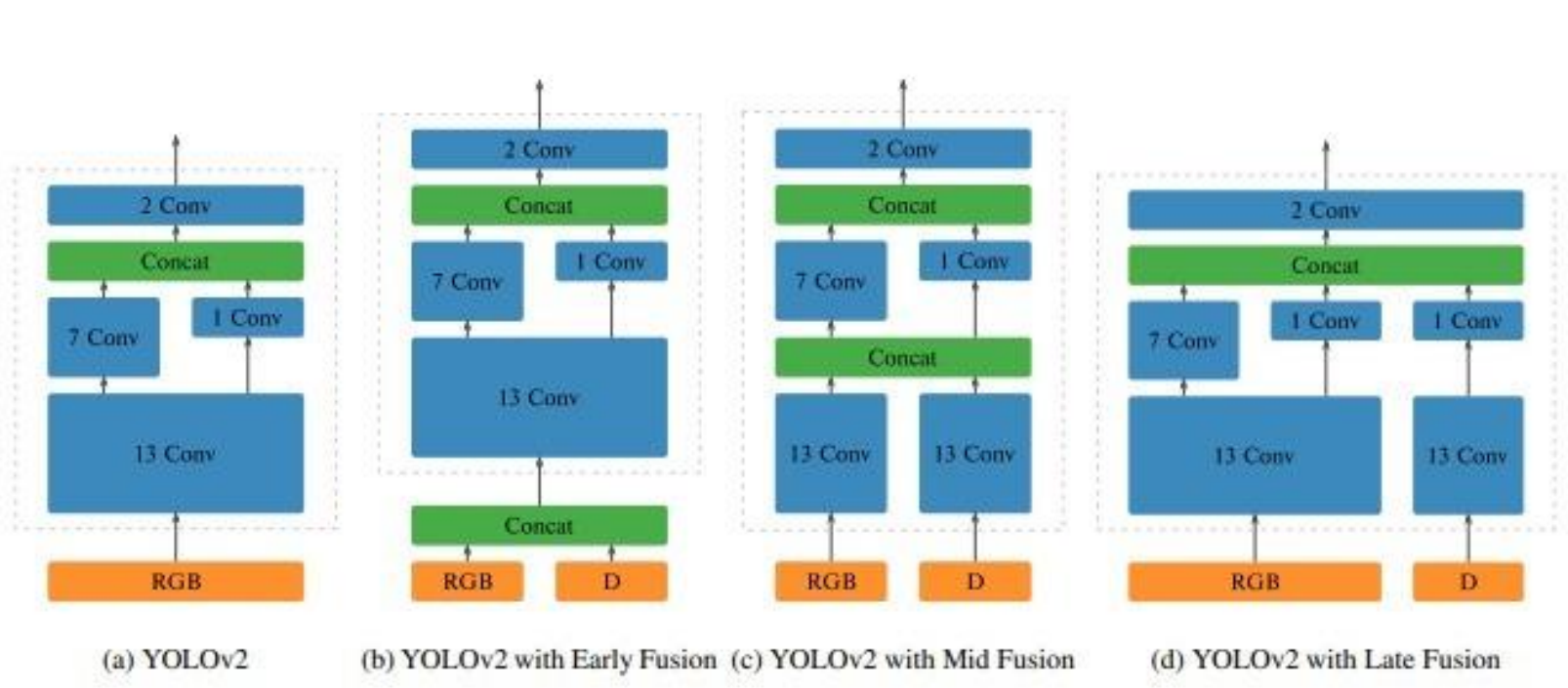


(c)



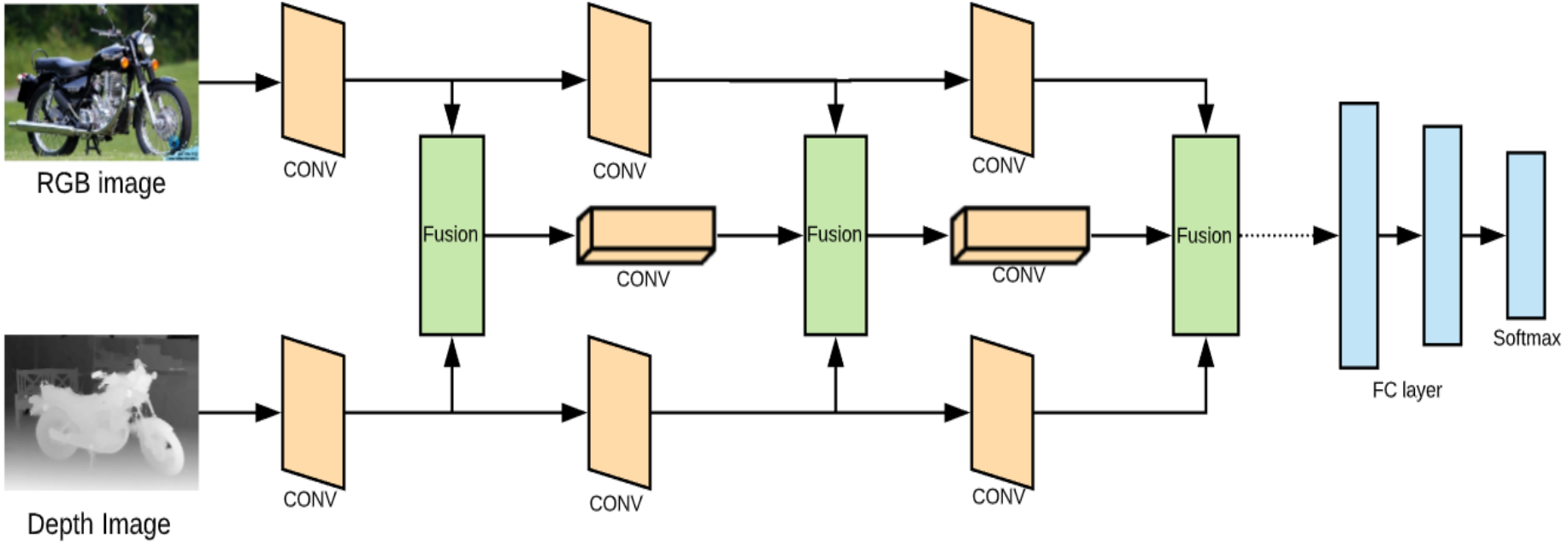
(d)

Our Approach

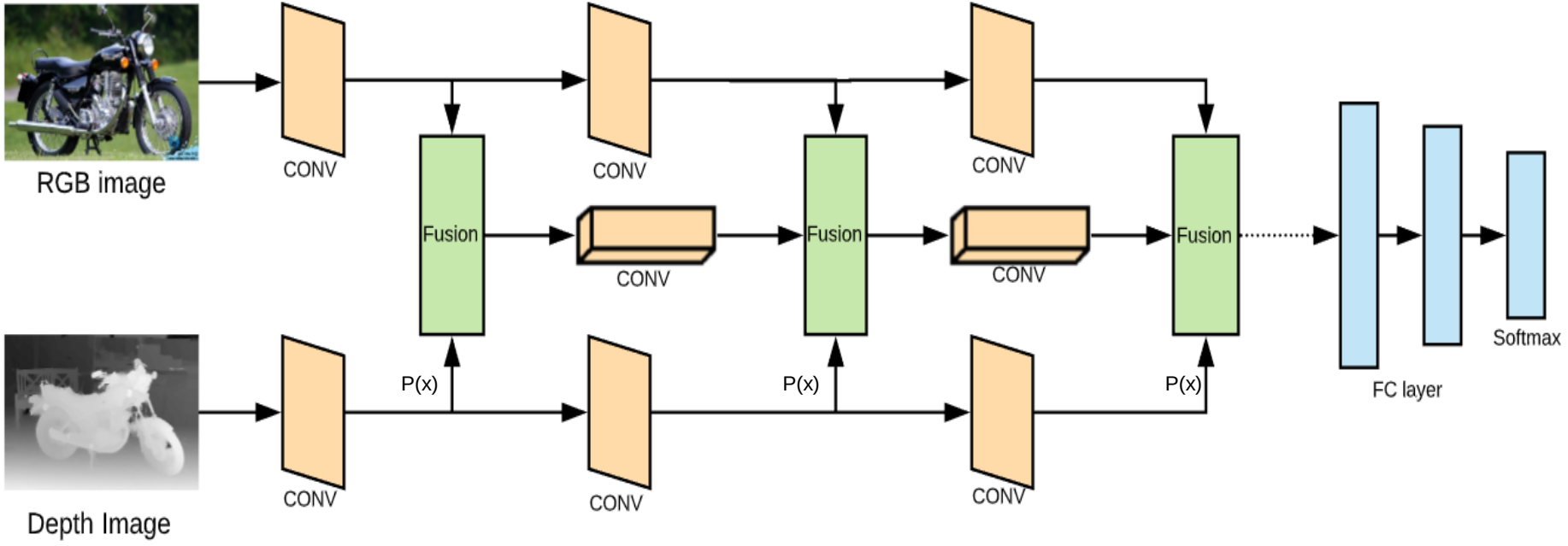


(a) YOLOv2 (b) YOLOv2 with Early Fusion (c) YOLOv2 with Mid Fusion (d) YOLOv2 with Late Fusion

Proposed Architecture



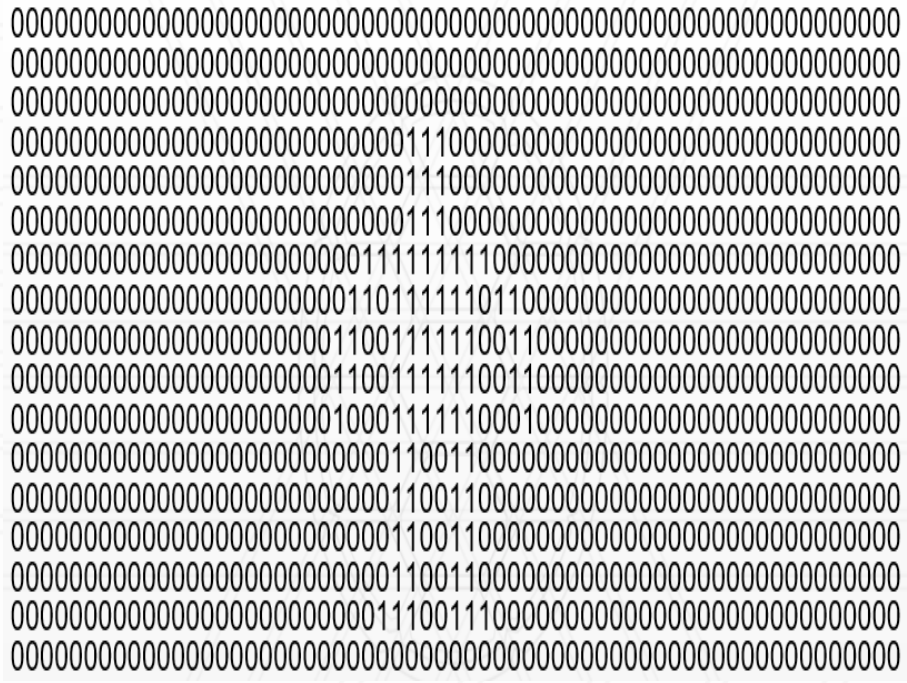
Proposed Arch. Continued

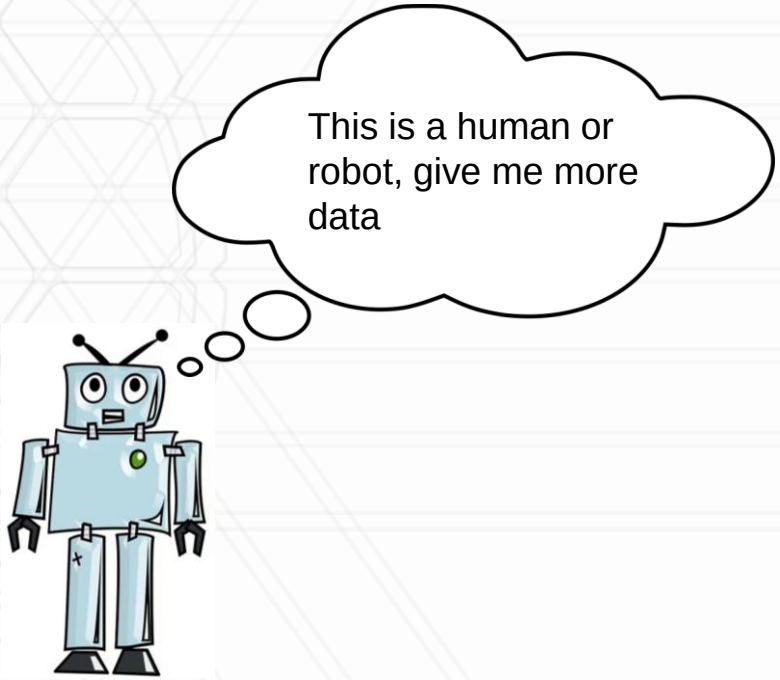


Expectation

- Time taken for a detection will increase, as the number of fusion layer increases.
- Difficulty in training as the number of parameters increases.
- Might get increase in the accuracy/AP metric.

Future Work





UNIVERSITY OF
MARYLAND

Thank You