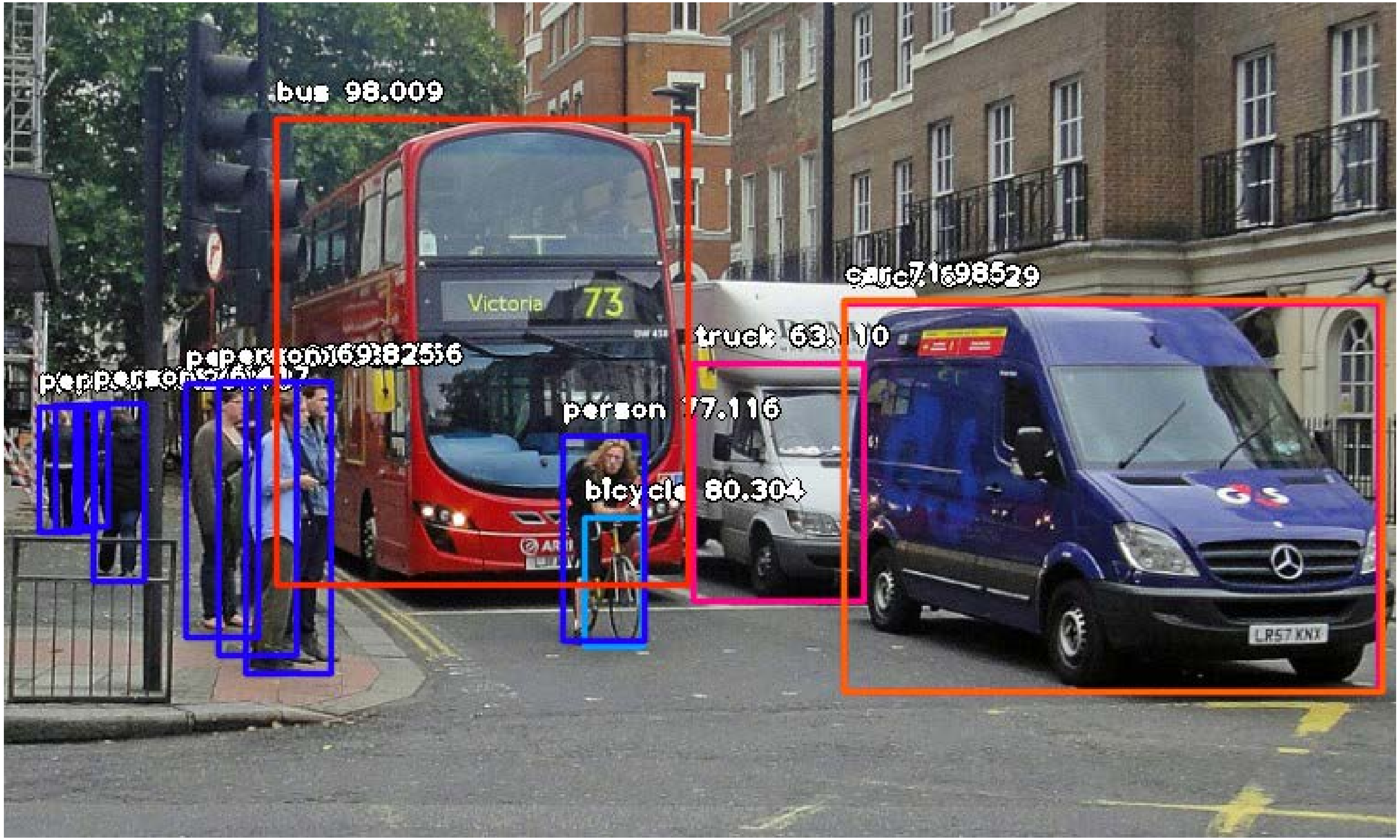
# SSD : Single Shot MultiBox Detector

ECCV 2016

# Main Task : Object Detection




Localization + Classification

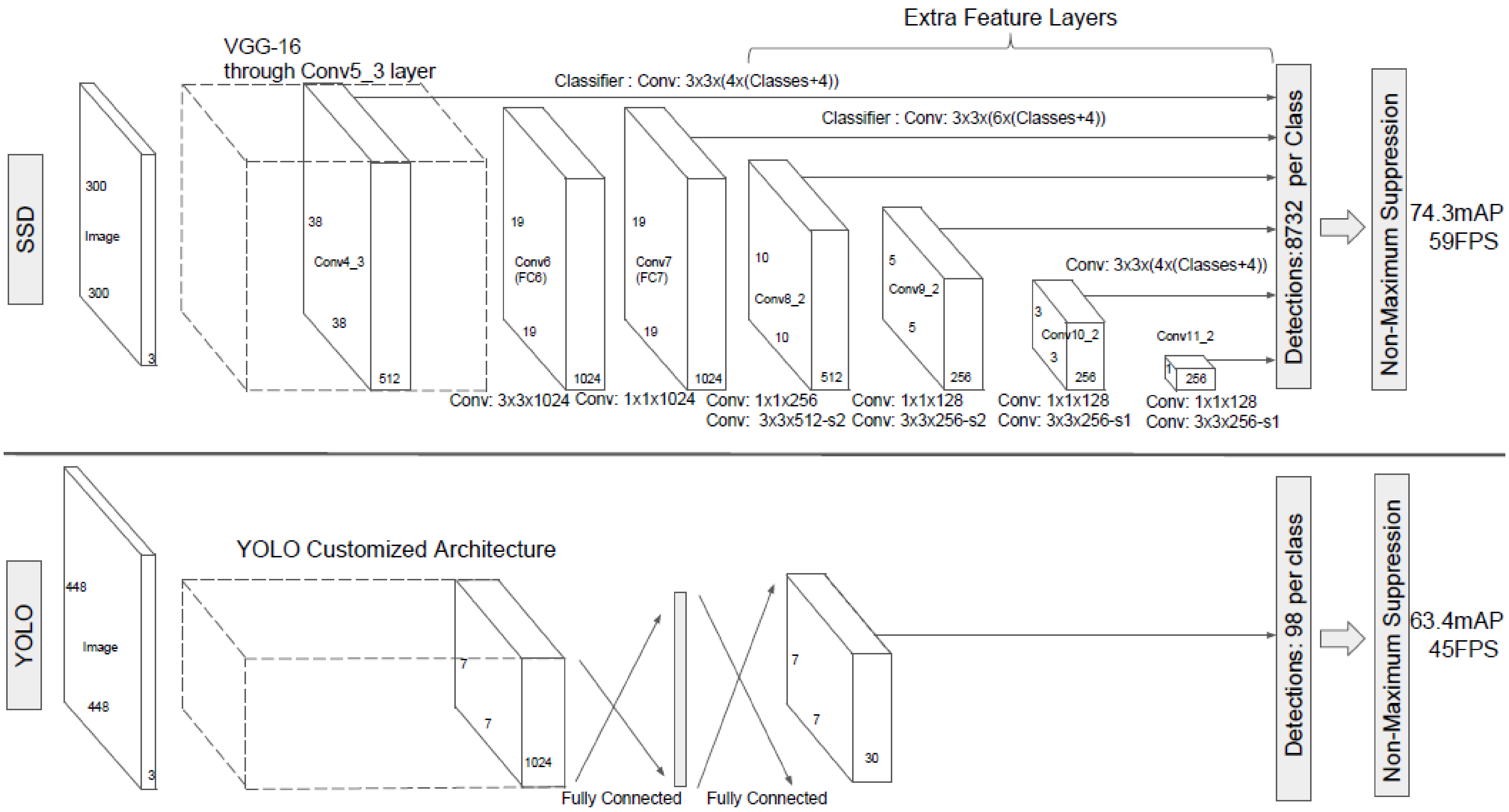
# Main Problem

- But, conventional object detection network has several problems

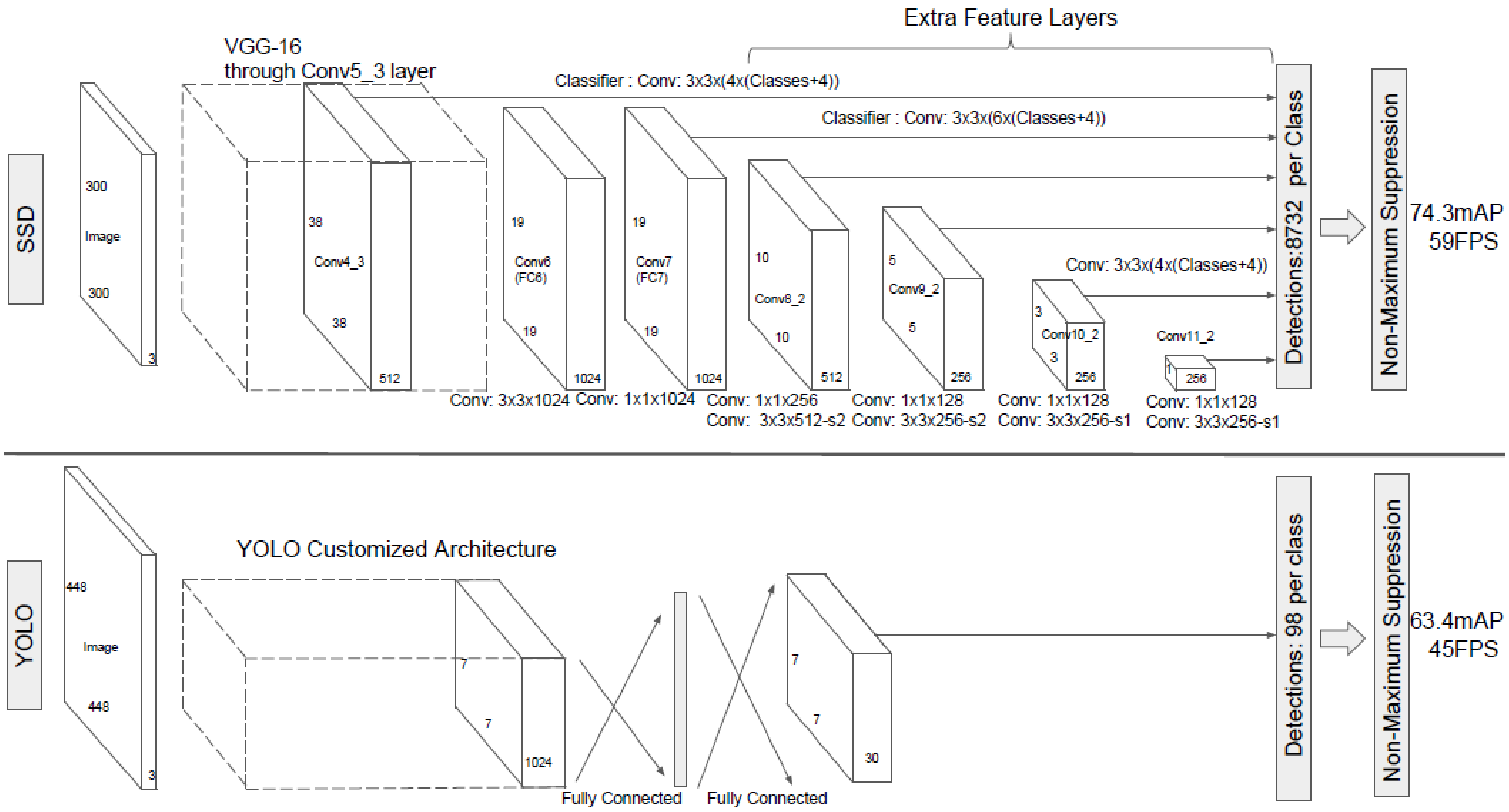
- R-CNN series
  - Good performance
  - Too slow

- Yolo
  - So fast
  - Low performance

# Related researches

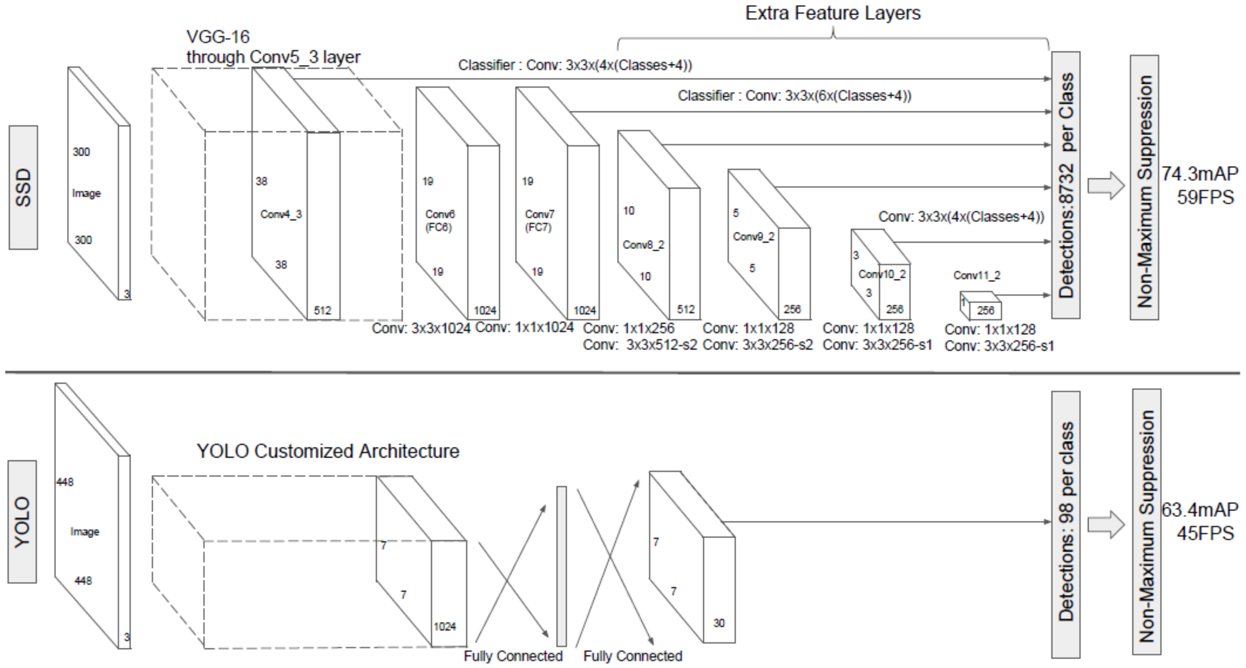


# Solution : SSD architecture

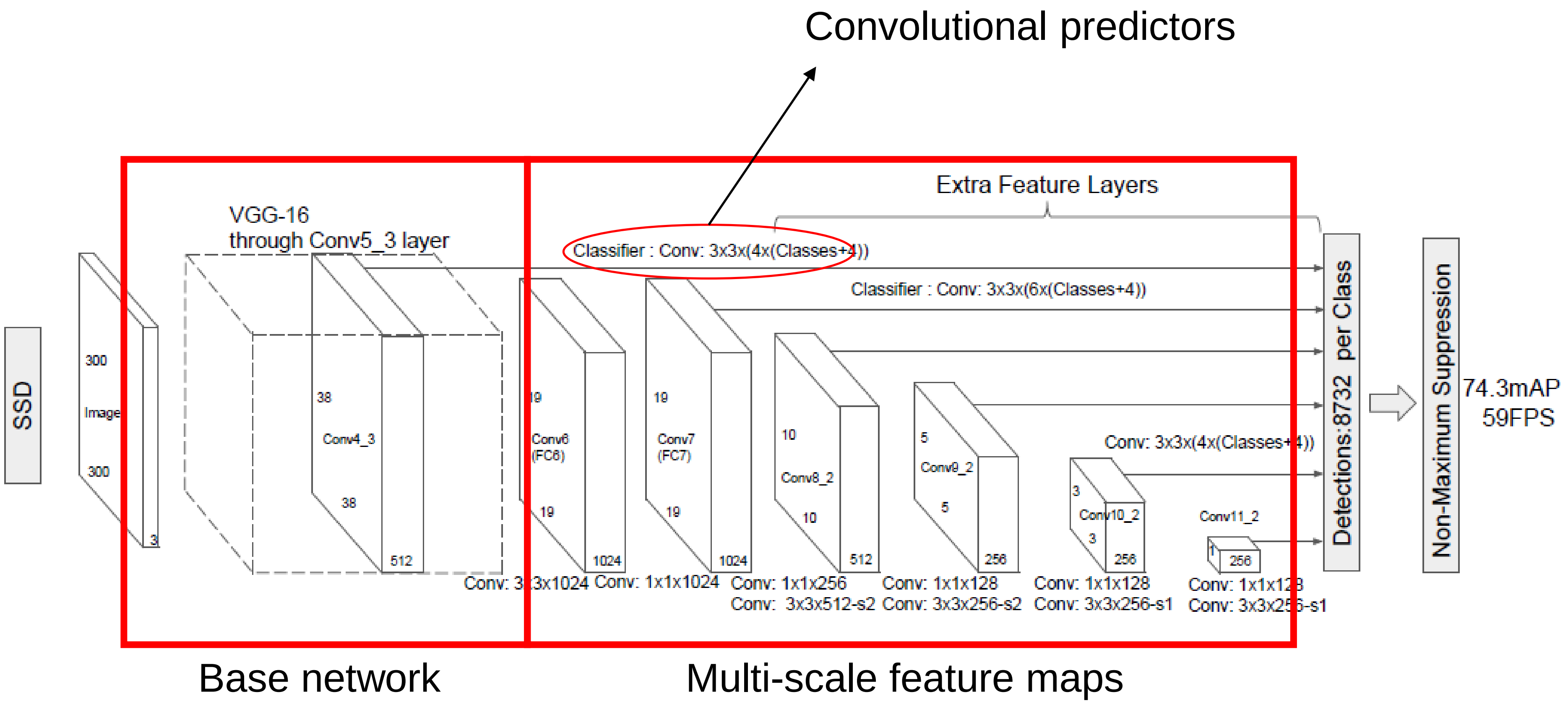


# SSD : Main Contributions

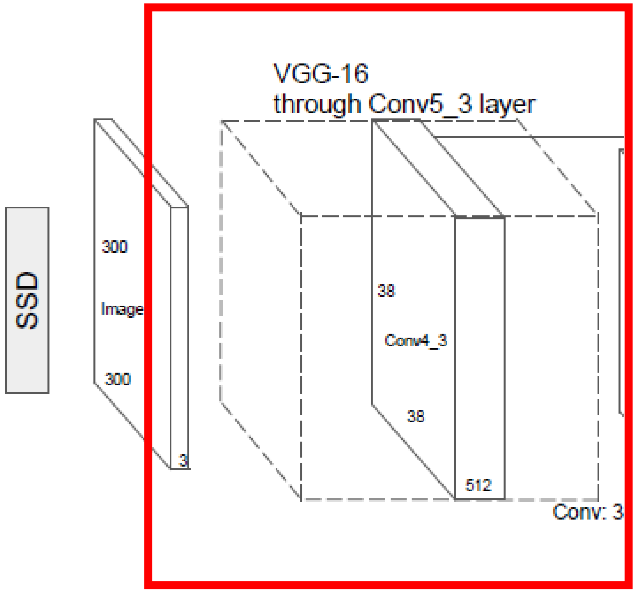
- Propose faster than YOLO, and more accurate, comparable with R-CNN series
- Use box offsets for more faster&accurate prediction
- Use multi-scale prediction scheme



# SSD : Architecture



# SSD : Architecture




Base network

- Backbone : VGG16
- Used for high-quality image classification
  - Learned common features of images
- Truncated at Conv5_3 layer
  - 300x300 → 38x38

# SSD : Architecture

<Multi-scale feature maps>

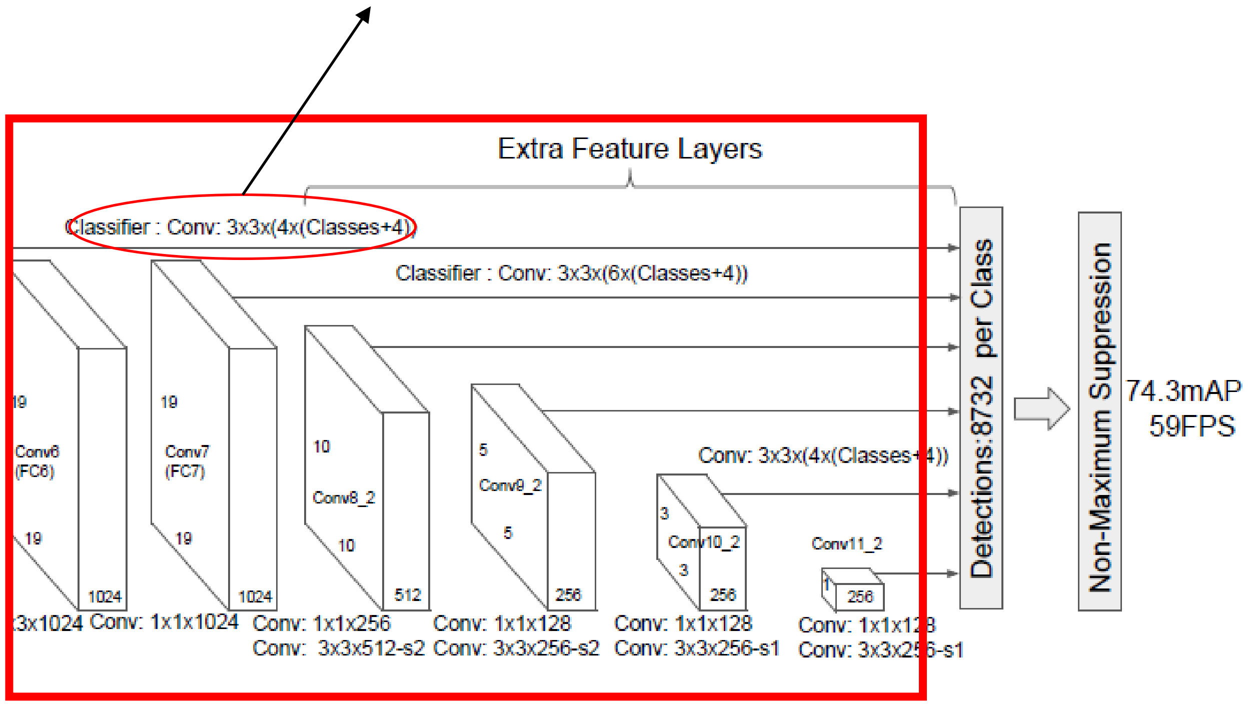
- Added auxiliary convolutional layers for Multi-scale detections

<Convolutional predictors>

- Produce a fixed set of detection predictions by using CNN

- Applied to feature maps for each scales

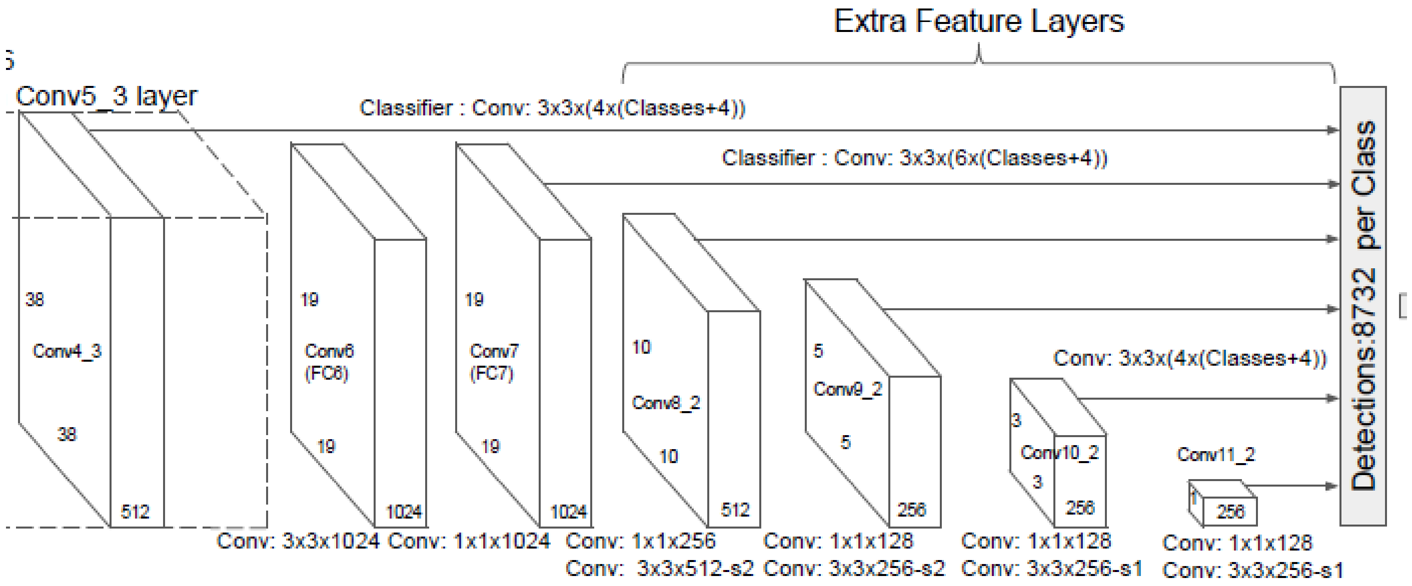
Convolutional predictors




Multi-scale feature maps

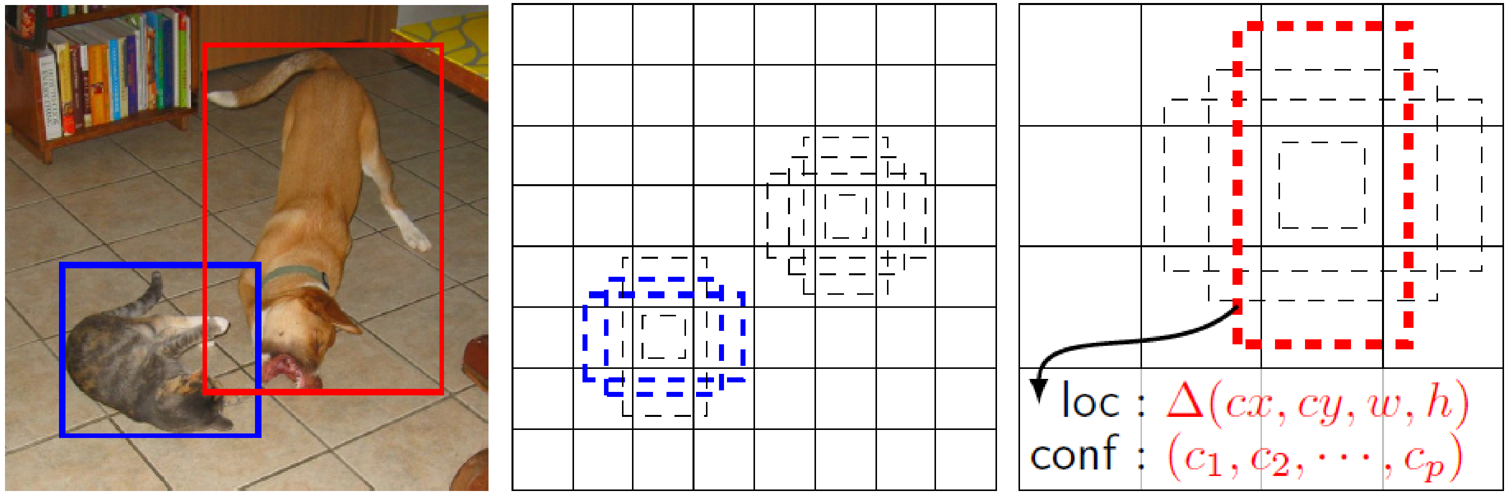
# SSD : Architecture

- Produce fixed number of predictions for each feature map
- If we predict k bounding box region, we should use (C+4)k channel CNN kernel
  - For mxn feature map, produce (C+4)kmn predictions
- Predict the offsets relative to the default box shapes in the cell



# SSD : Details

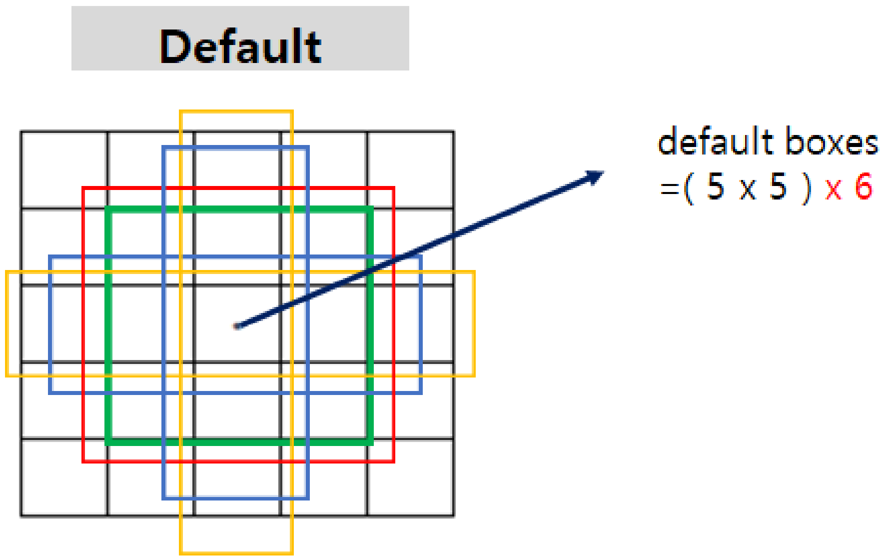
- Discretize space of the possible output box shapes
- Allocated default box set to each scales
- Useful to predict bounding box




(a) Image with GT boxes (b) $8 \times 8$ feature map (c) $4 \times 4$ feature map

# SSD : Details

- Match default boxes to ground truth and select boxes with threshold (IOU 0.5)
- Use multiple boxes, calculate loss function
- Useful to predict bounding box

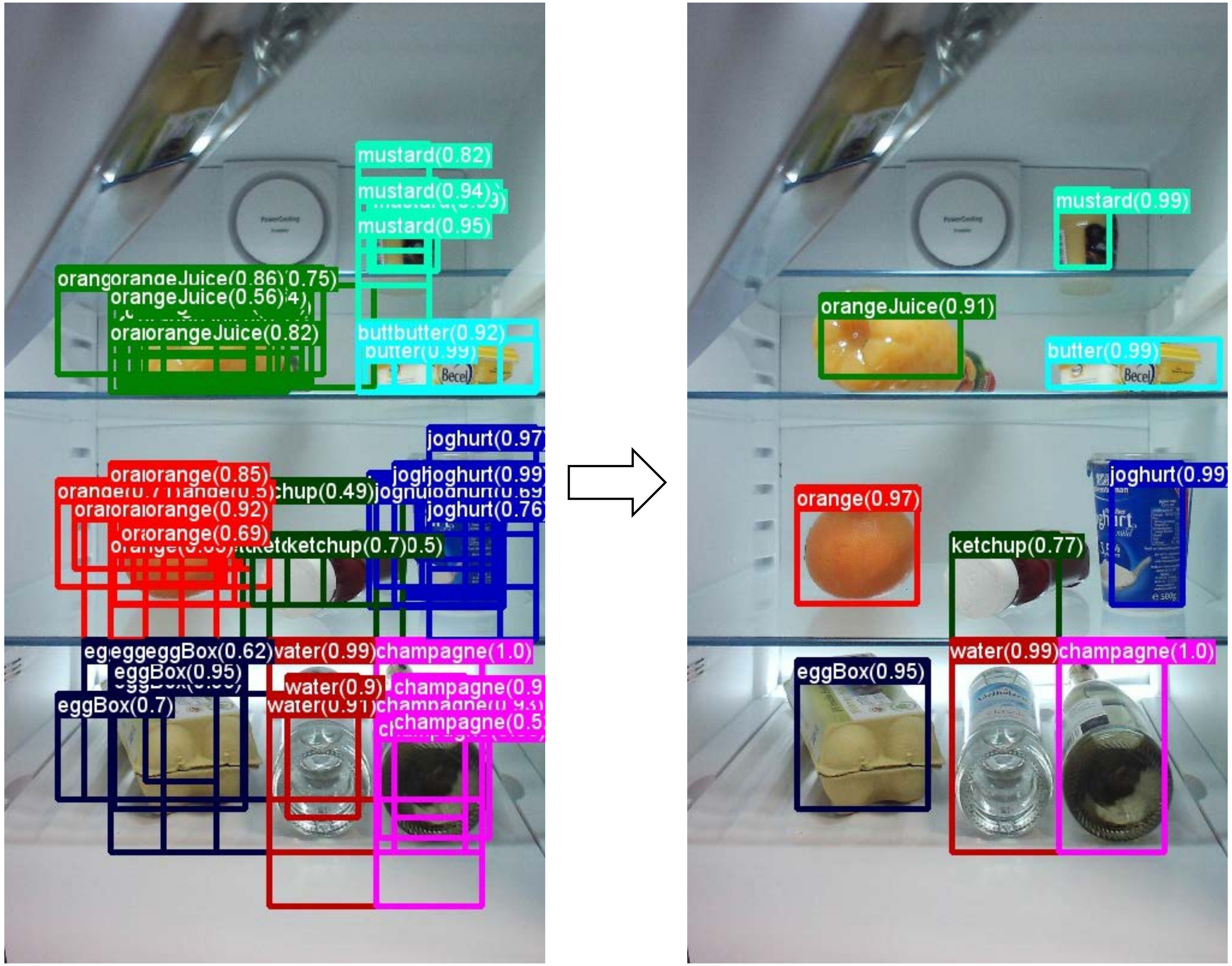


# SSD : Non-maximum suppression

- Remove overlapped bounding boxes

- Sorting list of bounding box by confidence

- Eliminate overlapped bounding box, determine by IOU threshold



# SSD : Training

- Summarized form

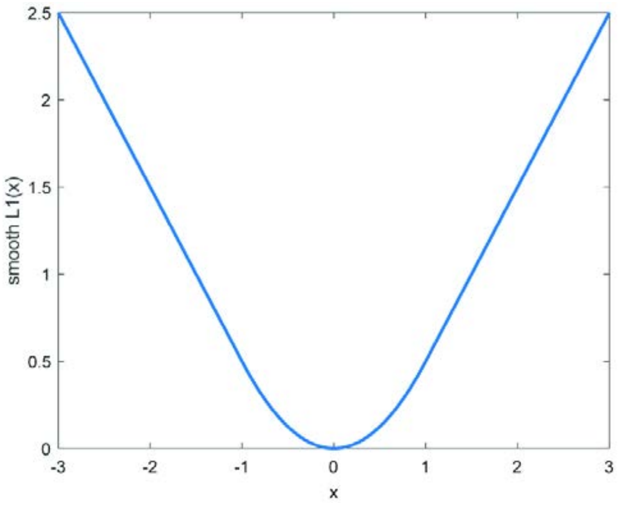
$$L(x,c,l,g) = \underbrace{\frac{1}{N}}_{\text{Number of matched default boxes}}(\underbrace{L_{conf}(x,c)}_{\text{Confidence loss}} + \alpha \underbrace{L_{loc}(x,l,g)}_{\text{Localization loss}})$$

# SSD : Training

- Localization loss

$$L_{loc}(x,l,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} \mathrm{smooth}_{\mathrm{L1}}(l_i^m - \hat{g}_j^m)$$

$$\boxed{\begin{array}{cc} \hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w & \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h \\ \hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) & \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \end{array}}$$




- Confidence loss

Classification probability

$$L_{conf}(x,c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

Matching coefficient

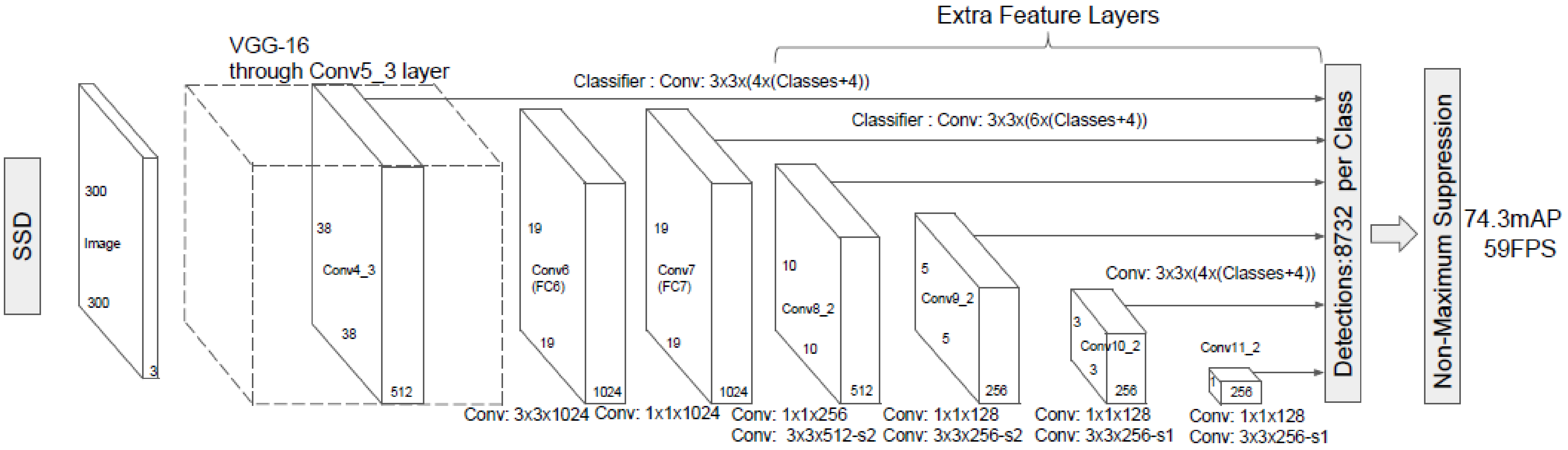
# SSD : Training

$$s_k = s_{\text{min}} + \frac{s_{\text{max}} - s_{\text{min}}}{m-1}(k-1), \quad k \in [1, \underline{m}]$$

Number of feature map

- $S \in [0.2, 0.9]$
- $a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$ → $w_k^a = s_k\sqrt{a_r}$ , $h_k^a = s_k/\sqrt{a_r}$
- Determine scale of default boxes

# SSD : Conclusion




- Proposed new brilliant architecture for object detection
- Faster, but more accurate