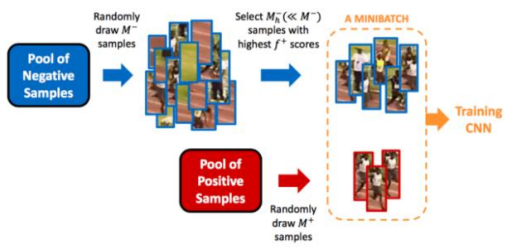
Dilated/Atrous Convolution:

Conv6 and Conv7 layers are dilated convolution layers. Dilated convolution can increase the receptive field keeping the number of parameters lower.

Hard negative mining:

During training, most bounding boxes are going to have low IoU and therefore will be classified as negative training examples, this can lead to an accumulation of disproportionate number of negative examples. Therefore, instead of using all the negative examples, a given ratio of negative:positive examples should be kept. It is important to keep some negative examples for the model to see what constitutes as a negative example.



Example of hard negative mining (from Jamie Kang blog)

SSD sorts all the negatives by their calculated confidence loss (after making a prediction) and picks only to top ones while keeping a ratio of at most 3:1 of positive:negative examples.

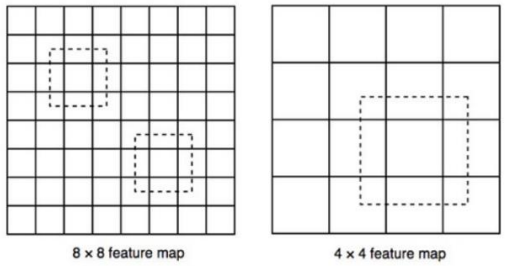
SSD:

A single shot detector that uses VGG-16 for feature map construction, but discards the fully connected layers. Instead of the fully connected layers, they add auxiliary convolutional layers that decrease in size progressively which helps detection of objects in **multiple scales**.

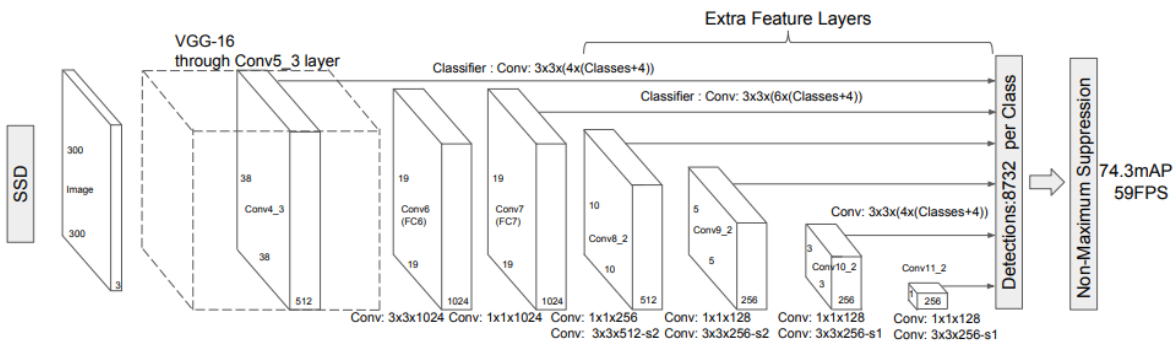
Auxiliary layers and way of traversal over feature layers:
Each feature cell in VGG-16's Conv4_3 layer is assigned with 4 default bounding boxes in different scales. In addition, SSD adds 6 more auxiliary convolutional layers(Conv6 – Conv11_2), 5 of them are used for object detection. Conv7, Conv 8_2 and Conv9_2 layers have 6 bounding boxes, Conv10_2 and Conv11_2 layers have 4 bounding boxes per feature cell. At every feature map cell, an offset value to the default boxes is predicted, in addition to the per-class score that indicates the presence of a class in every box. For every box of the k default boxes, c class predictions and 4 offsets are computed. This results in (c+4)*k filters that are applied around every feature cell**. Non-maximum suppression** is used to produce the final detections.

Predictors:
For a feature layer (ConvX) of size mXn with p channels 3X3Xp small kernels are used to produce either a score for a category (number of classes + 1 because background class is added) or a shape offset relative to the default box coordinates. Lower resolution feature layers are used to detect large objects

and high resolution feature layers detect small objects.



For example: At Conv4_3 we have a feature layer of size 38X38X512 with 4 bounding boxes per feature cell. A [3,3,512,4(21+4)] convolution is used to produce 21 class predictions and 4 offsets for each bounding box.



Default boxes and aspect ratios:
Default bounding boxes sized are scales are chosen manually. Starting from the left, Conv4_3 gets the smallest boxes (0.2) for small object detection and increases linearly until the rightmost layer Conv11_2 (0.9). Starting with 5 target aspect ratios (1, 2, 3, 1/2, 1/3) the width and height of each box is calculated as follows:

$$w = scale \cdot \sqrt{\text{aspect ratio}}$$
$$h = \frac{scale}{\sqrt{\text{aspect ratio}}}$$

Then SSD adds an extra default box with scale:

$$scale = \sqrt{\text{scale} \cdot \text{scale at next level}}$$

and aspect ratio = 1.

Matching strategy:
SSD matches are classified as positive matches and negative matches. SSD uses only the positive matches when calculating the location loss. A match is considered positive if it the corresponding bounding box (not the predicted one) has an IoU greater than 0.5(Matching all of them, not just the highest one).

Loss:

The confidence loss is the Softmax loss over c classes confidences.

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

The confidence loss is smooth L1 Loss between the predicted box(l) and the ground-truth box(g)

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

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^{w} = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^{h} = \log\left(\frac{g_j^h}{d_i^h}\right)$$

Where Xijk is an indicator of matching the i'th bounding box to the j'th ground truth box with category k. (if the model think there is no object' the loss is set to 0).

In total, the loss is defined as:

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))$$

Where N is the number of matched bounding boxes and alpha is the weight for the location loss.

SSD uses hard negative mining and data augmentation strategies during training.