DeNet: Scalable Real-time Object Detection with Directed Sparse Sampling
Supplementary Material

Lachlan Tychsen-Smith, Lars Petersson
CSIRO (Data61)
7 London Circuit, Canberra, ACT, 2601
Lachlan.Tychsen-Smith@data61.csiro.au, Lars.Petersson@data61.csiro.au

1. Instance Estimation (Non-Max Supression)

From the classifier output $\Pr(s|B_S)$ we cluster *detector hits* of the same object to identify unique instances. Following standard practice, this operation is performed via a variant of the Non-Max Suppression (NMS) algorithm. In this algorithm, if two detector hits have the same class and a $\text{IoU} > 0.5$ the hit with the lowest confidence is discarded. An optimized C++ implementation was employed.

2. Cost Function Constants (Λ)

Below we provide the constants used for the training cost function:

$$\Lambda_t = BWH\ln(2) \tag{1}$$
$$\Lambda_s = BN^2\ln(C+1) \tag{2}$$
$$\Lambda_b = BN^2 \tag{3}$$

where B is the batch size, (W, H) is the width and height of the corner distribution (i.e. 64×64), C is the number of classes and N^2 is the number of sampling bounding boxs.

3. Corner Layer Formulation

The **corner** layer performs the following transformation on the input activations $\alpha_{t,y,x}$ to obtain the corner distribution $\Pr(s|k,y,x)$ and sampling feature map $F_{q,y,x}$.

$$F_{q,y,x} = \sum_t \omega^F_{q,t}\alpha_{t,y,x} \tag{4}$$
$$\beta_{k,y,x} = \sum_t \omega^P_{k,t}\alpha_{t,y,x} \tag{5}$$
$$\Pr(s|k,y,x) = \frac{\exp(s\beta_{k,y,x})}{\exp(\beta_{k,y,x}) + \exp(-\beta_{k,y,x})} \tag{6}$$

where ω^F and ω^P are learnt parameters, and $s = \{-1, 1\}$ indicates the presence of a corner . In practice both linear transforms are performed with a single operation.

4. Sparse Layer Formulation

As input the **sparse** layer takes the feature map $F_{q,y,x}$ and a set of bounding boxs $B_{n,m} = (x_{n,m}, y_{n,m}, w_{n,m}, h_{n,m})$ where x, y, w, h are defined as spatial positions in the sampling feature map. Subsequently, the **sparse** layer produces the output activations $\alpha_{p,n,m}$ where:

$$\alpha_{p,n,m} = \begin{cases} F_{\bar{\sigma}(n,m,p)} & p < |P| - 2 \\ w_{n,m} & p = |P| - 2 \\ h_{n,m} & p = |P| - 1 \end{cases} \tag{7}$$

where $|P| = QP_WP_H + 2$ is the size of the feature vector, Q the sampling feature map depth and (P_W, P_H) is the feature's spatial width and height. Therefore, apart from the last two values which provide the bounding box width and height, each value in the feature vector is mapped to a location in the sampling feature map via the function $\bar{\sigma}(n, m, p) = \sigma_q(p), \sigma_y(n, m, p), \sigma_x(n, m, p)$ defined below.

$$i(p) = \lfloor p/Q \rfloor \tag{8}$$

$$\sigma_q(p) = p \bmod Q \tag{9}$$

$$\sigma_y(n, m, p) = y_{n,m} + \frac{i(p) \bmod P_W}{P_H - 1} h_{n,m} \tag{10}$$

$$\sigma_x(n, m, p) = x_{n,m} + \frac{\lfloor i(p)/P_W \rfloor}{P_W - 1} w_{n,m} \tag{11}$$

During back propagation the indexing function $\bar{\sigma}(n, m, p)$ is held constant.