# [Supplementary Material]
# Improving Occlusion and Hard Negative Handling for Single-Stage Pedestrian Detectors

Junhyug Noh  Soochan Lee  Beomsu Kim  Gunhee Kim
Department of Computer Science and Engineering
Seoul National University, Seoul, Korea
{jh.noh, soochan.lee}@vision.snu.ac.kr, {123bskim, gunhee}@snu.ac.kr

The contents of this supplementary material are as follows.

- Section 1 present the detail settings of four base models. Section 1.1 shows the backbone networks and datasets for pre-training. Section 1.2 discusses details of anchor settings. Section 1.3 explains the localization and confidence loss for each base model. Section 1.4 shows the default coefficients for the other types of losses.
- Section 2 shows some statistics about the ground truths of part confidence maps.
- Section 3 explains the details of training and inference process of our method.
- Section 4–5 present results of ablation experiments on our occlusion and hard negative handling methods, respectively. We measure the performance variation according to the configurations of our key components, including the size changes of the part confidence map, doubling the layer depths of occlusion handling methods, and different uses of the grid confidence map of the grid classifiers.
- Section 6 compare our results with state-of-the-art models.
- Section 7 illustrates more qualitative results that visualize effects of occlusion and hard negative handling.

## 1. Detail Settings of Base Models

### 1.1. Backbone Networks

Table 1 shows the backbone network and dataset on which the pre-trained weights were trained for each model.

| Model | Backbone | Dataset for pre-training |
|---|---|---|
| SqueezeDet+ [13] | SqueezeNet+ [6] | ImageNet classification [2] |
| YOLOv2 [9] | DarkNet-19 [9] | PASCAL VOC object detection [4] |
| SSD [8] | VGG16 [10] | MS COCO object detection [7] |
| DSSD [5] | VGG16 [10] | MS COCO object detection [7] |

Table 1: Backbone networks and datasets for pre-training of baseline models.

### 1.2. Anchor Settings

Table 2 shows the setting of default anchor boxes for four base models.

### 1.3. Loss Formulations

Single-stage models in this paper have two types of losses: localization and confidence loss. The classification loss, which is generally specified separately, is omitted since there is only one class in our task. There are slight differences in the formulae

| Model | Shape of Output Tensors $\{W, H, K\}$ | Anchors | |
|---|---|---|---|
| | | Relative Height | Aspect Ratio |
| SqueezeDet+ [13] | {38, 28, 9} | {0.10, 0.14, 0.18, 0.24, 0.31, 0.41, 0.54, 0.71, 0.94} | {0.41} |
| YOLOv2 [9] | {20, 15, 9} | {0.10, 0.14, 0.18, 0.24, 0.31, 0.41, 0.54, 0.71, 0.94} | {0.41} |
| SSD [8] | {40, 30, 4} | {0.05, 0.09, 0.13, 0.18} | {0.41} |
| | {20, 15, 3} | {0.18, 0.25, 0.36} | {0.41} |
| | {10, 8, 3} | {0.36, 0.50, 0.72} | {0.41} |
| | {8, 6, 2} | {0.72} | {0.41} |
| | {6, 4, 1} | {0.90} | {0.41} |
| DSSD [5] | {1, 1, 3} | {0.94} | {0.20, 0.41, 0.60} |
| | {6, 4, 3} | {0.88} | {0.20, 0.41, 0.60} |
| | {8, 6, 6} | {0.75, 0.81} | {0.20, 0.41, 0.60} |
| | {10, 8, 6} | {0.63, 0.69} | {0.20, 0.41, 0.60} |
| | {20, 15, 3} | {0.33, 0.42, 0.53} | {0.41} |
| | {40, 30, 3} | {0.17, 0.21, 0.26} | {0.41} |
| | {80, 60, 3} | {0.08, 0.10, 0.13} | {0.41} |
| | {160, 120, 1} | {0.04} | {0.41} |

Table 2: Specification of anchors for four base models. The relative height is a height value with respect to the image height, and the aspect ratio is a ratio of the width to the height.

for each model. Specifically, YOLOv2 and SqueezeDet+ use the same formulations which differs from the ones of SSD and its derivative DSSD.

**YOLOv2 / SqueezeDet+**. For the localization loss of each anchor box, YOLOv2 and SqueezeDet+ use $\ell_2$ loss in Eq.(1). The target of confidence is the IOU between predicted box and ground truth. Therefore, they use $\ell_2$ loss between the IOU and predicted confidence as confidence loss as in Eq.(2). In the equations, $\mathbb{I}^{+}_{(ijk)} = 1$ indicates that the $(ijk)$-th anchor is a positive example while $\mathbb{I}^{-}_{(ijk)} = 1$ indicates a negative example.

$$\mathcal{L}_{l,(ijk)} = \mathbb{I}^{+}_{(ijk)} \left\{ \sum_{l \in \{x,y,w,h\}} \left( \delta_{l,(ijk)} - \hat{\delta}_{l,(ijk)} \right)^2 \right\} \tag{1}$$

$$\mathcal{L}_{c,(ijk)} = \lambda_c^{+} \mathbb{I}^{+}_{(ijk)} \left( \mathrm{IOU}^{gt}_{(ijk)} - \hat{c}_{(ijk)} \right)^2 + \lambda_c^{-} \mathbb{I}^{-}_{(ijk)} \hat{c}^2_{(ijk)} \tag{2}$$

**SSD / DSSD**. SSD and DSSD use smooth $\ell_1$ loss for the localization loss (Eq.(3)) and log-loss for the confidence loss (Eq.(4)).

$$\mathcal{L}_{l,(ijk)} = \mathbb{I}^{+}_{(ijk)} \left\{ \sum_{l \in \{x,y,w,h\}} \mathrm{s}_{\ell 1} \left( \delta_{l,(ijk)} - \hat{\delta}_{l,(ijk)} \right) \right\} \tag{3}$$

$$\mathcal{L}_{c,(ijk)} = -\lambda_c^{+} \mathbb{I}^{+}_{(ijk)} \log \hat{c}_{(ijk)} - \lambda_c^{-} \mathbb{I}^{-}_{(ijk)} \log(1 - \hat{c}_{(ijk)}) \tag{4}$$

## 1.4. Loss Coefficients

The coefficients of localization loss ($\lambda_l$) and confidence loss ($\lambda_c$, $\lambda_c^{+}$, $\lambda_c^{-}$) are the same as those of original paper for each model. For the other losses, their coefficients are set as shown in Table 3 where the normalizers are defined as in Eq.(5)–(7).

$$N^{+} = \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{K}\mathbb{I}^{+}_{(ijk)}, \qquad N^{-} = \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{K}\mathbb{I}^{-}_{(ijk)} \tag{5}$$

$$N^{+f} = \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{K}\mathbb{I}^{+f}_{(ijk)}, \qquad N^{+o} = \sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{K}\mathbb{I}^{+o}_{(ijk)} \tag{6}$$

$$N^{+}_{g} = \sum_{l=1}^{L}\sum_{i=1}^{w_l}\sum_{j=1}^{h_l}\mathbb{I}^{+}_{l,(ij)}, \qquad N^{-}_{g} = \sum_{l=1}^{L}\sum_{i=1}^{w_l}\sum_{j=1}^{h_l}\mathbb{I}^{-}_{l,(ij)} \tag{7}$$

| Loss | Coefficient | Model | | | |
|---|---|---|---|---|---|
| | | SqueezeDet+ [13] | YOLOv2 [9] | SSD [8] | DSSD [5] |
| $L_p$ | $\lambda_p$ | 1 | 1 | 1 | 1 |
| | $\lambda_p^{+}$ | $75/N^{+}$ | $5/(N^{+}+N^{-})$ | $1/(N^{+}+N^{-})$ | $1/(N^{+}+N^{-})$ |
| | $\lambda_p^{-}$ | $100/N^{-}$ | $1/(N^{+}+N^{-})$ | $1/(N^{+}+N^{-})$ | $1/(N^{+}+N^{-})$ |
| $L_s$ | $\lambda_s$ | 1 | 1 | 1 | 1 |
| | $\lambda_s^{+f}$ | $75/N^{+f}$ | $1/(N^{+f}+N^{+o}+N^{-})$ | $1/(N^{+f}+N^{+o}+N^{-})$ | $1/(N^{+f}+N^{+o}+N^{-})$ |
| | $\lambda_s^{+o}$ | $150/N^{+o}$ | $5/(N^{+f}+N^{+o}+N^{-})$ | $1/(N^{+f}+N^{+o}+N^{-})$ | $1/(N^{+f}+N^{+o}+N^{-})$ |
| | $\lambda_s^{-}$ | $75/N^{-}$ | $1/(N^{+f}+N^{+o}+N^{-})$ | $1/(N^{+f}+N^{+o}+N^{-})$ | $1/(N^{+f}+N^{+o}+N^{-})$ |
| $L_g$ | $\lambda_{g,l}$ | 5 | 1 | 1 | 3 |
| | $\lambda_g^{+}$ | $75/N_g^{+}$ | $5/(N_g^{+}+N_g^{-})$ | $1/(N_g^{+}+N_g^{-})$ | $1/(N_g^{+}+N_g^{-})$ |
| | $\lambda_g^{-}$ | $100/N_g^{-}$ | $1/(N_g^{+}+N_g^{-})$ | $1/(N_g^{+}+N_g^{-})$ | $1/(N_g^{+}+N_g^{-})$ |

Table 3: Default coefficients for each type of loss.

## 2. Distribution of Ground Truths of Part Confidence Maps

In this section, we show some statistics of ground truth labels that are used to train occlusion handling models. We limit the ground truth that are labeled as *person* and its occluded fraction is less than $0.8$, which amounts to 125,623 and 15,371 instances for Caltech and CityPersons dataset respectively. Table 4–5 show how *part confidence maps* are distributed according to its grid size ($M \times N$). For Caltech pedestrian dataset, more than $80\%$ of examples are fully visible, and the examples where below parts are occluded are the most common cases among partially visible examples. In contrast, we can see that more occluded cases and patterns are found in CityPersons dataset.

## 3. Details of Training and Test (Inference) Process

**Training process**. For each ground truth bounding box, we associate the following anchors as positive examples: (i) the anchor box with the highest IOU value, and (ii) the anchor boxes whose IOU values are over $0.5$. We select anchor boxes whose IOU values are less than 0.4 as negative examples. We ignore anchors whose IOU Values are between 0.4 and 0.5 for calculating the loss. Because all base models define many anchor boxes as default, there are overwhelming many negative examples compared to positive examples. Therefore, instead of including them all negative examples in the loss calculation, we select only the negative examples that cause the highest loss for each loss type (confidence, part, score, and grid loss). We use the ratio of negative to positive examples as a hyperparameter. (Only for score loss $L_s$, we use the ratio to occluded examples.) We set 3 for SSD and 10 for the other models. To optimize the loss function, we use the Adam optimizer with $\ell_2$ regularization.

**Inference process**. At test time, for a given image, we first compute the output tensor via forward pass, and then obtain the final prediction results by applying the grid classifiers and non-maximum suppression (NMS). For fast inference, we apply these two steps to only top 256 predicted boxes with the highest confidences. We set the IOU threshold of NMS to 0.6 for Caltech and 0.5 for CityPersons dataset.

## 4. Ablation Experiments of Occlusion Handling Methods

In this section, we present more experimental results about our occlusion handling methods, in which we measure the performance changes by varying the configuration of key components.

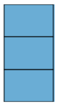
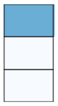
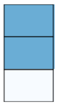
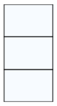
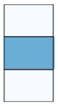
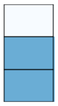
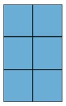
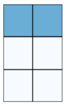
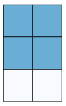
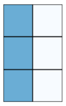
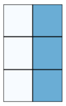
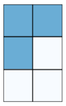
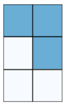
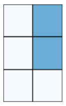
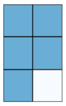
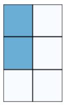
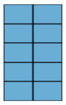
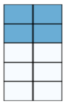
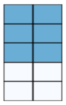
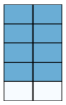
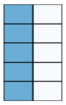
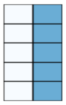
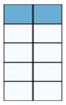
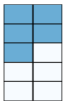
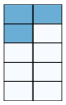
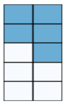
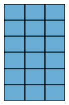
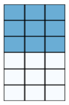
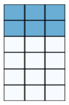
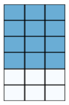
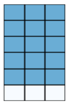
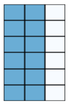
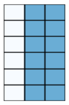
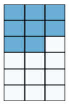
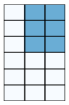
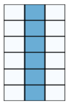
| $M \times N$ | Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $1 \times 3$ | Part<br>$(n = 6)$ |  |  |  |  |  |  | | | | |
| | Freq. | 105615 | 10537 | 8139 | 1007 | 170 | 155 | | | | |
| | Percent. | 84.07 | 8.39 | 6.48 | 0.80 | 0.14 | 0.12 | | | | |
| | Cum. | 84.07 | 92.46 | 98.94 | 99.74 | 99.88 | 100.0 | | | | |
| $2 \times 3$ | Part<br>$(n = 28)$ |  |  |  |  |  |  |  |  |  |  |
| | Freq. | 103935 | 9248 | 6572 | 938 | 891 | 692 | 633 | 467 | 366 | 351 |
| | Percent. | 82.47 | 7.36 | 5.23 | 0.75 | 0.71 | 0.55 | 0.50 | 0.37 | 0.29 | 0.28 |
| | Cum. | 82.47 | 90.10 | 95.33 | 96.08 | 96.78 | 97.34 | 97.84 | 98.21 | 98.50 | 98.78 |
| $2 \times 5$ | Part<br>$(n = 55)$ |  |  |  |  |  |  |  |  |  |  |
| | Freq. | 102925 | 8367 | 4384 | 2805 | 789 | 766 | 597 | 378 | 372 | 367 |
| | Percent. | 81.93 | 6.66 | 3.49 | 2.23 | 0.63 | 0.61 | 0.48 | 0.30 | 0.30 | 0.29 |
| | Cum. | 81.93 | 88.59 | 92.08 | 94.31 | 94.94 | 95.55 | 96.03 | 96.33 | 96.62 | 96.92 |
| $3 \times 6$ | Part<br>$(n = 171)$ |  |  |  |  |  |  |  |  |  |  |
| | Freq. | 101162 | 4128 | 3746 | 2369 | 1348 | 1256 | 1111 | 823 | 707 | 652 |
| | Percent. | 80.53 | 3.29 | 2.98 | 1.89 | 1.07 | 1.00 | 0.88 | 0.66 | 0.56 | 0.52 |
| | Cum. | 80.53 | 83.81 | 86.80 | 88.68 | 89.76 | 90.75 | 91.64 | 92.29 | 92.86 | 93.38 |

Table 4: Distribution of *part confidence maps* on Caltech *train* dataset. $n$ is the total number of parts. The $blue$ areas represent visible parts.

**Max part score**. For the *max part score* method, we test four different settings of the grid size ($M \times N$) of the *part confidence map* to find the most proper size.

**Soft part score**. For the *soft part score* method, we measure the performance variation according to the grid size ($M \times N$) of the *part confidence map*, and the depth of additional layers for calculating the part score. The reason for considering the depth is to check that single layer is sufficient to interpret the *part confidence map*. We test four different combinations between $2 \times 5$ and $3 \times 6$ for the grid size, and single and double layers. Beyond the single layer setting we proposed at the main draft (Eq.(8)), we also tested two fully connected layers to obtain the final detection score (Eq.(9)),

$$s_{\text{person}} = \sigma\left(\mathbf{W}_{s1,2}^{\top} \max\left(\mathbf{0}, \mathbf{W}_{s1,1}^{\top} \widehat{\mathbf{V}}\right)\right) \tag{8}$$

$$s_{\text{person}} = \sigma\left(\mathbf{W}_{s2,3}^{\top} \max\left(\mathbf{W}_{s2,2}^{\top} \max\left(\mathbf{0}, \mathbf{W}_{s2,1}^{\top} \widehat{\mathbf{V}}\right)\right)\right) \tag{9}$$

where $\sigma$ is a sigmoid function, and parameters to learn include $\mathbf{W}_{s1,1}, \mathbf{W}_{s2,1} \in \mathbb{R}^{(M \times N) \times S}$, and $\mathbf{W}_{s1,2} \in \mathbb{R}^{S \times 1}$, $\mathbf{W}_{s2,2} \in \mathbb{R}^{S \times S'}$, $\mathbf{W}_{s2,3} \in \mathbb{R}^{S' \times 1}$. In our experiments, number of nodes per layer is fixed to $S = 6$ for single layer, and $(S, S') = (45, 64)$ for double layer setting.

Table 6–9 show the results of occlusion handling methods for each model. For all models, soft part score method shows the best performance in general. This generally means that the information in the semantic part is more meaningful than the information in the basic part (each grid of part confidence map). However, the max part score method shows better results on *heavy* subset. Since heavily occluded person is visible only for small area, its confidence is highly correlated to the score of basic part.

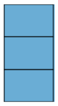
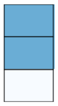
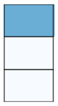
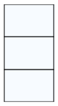
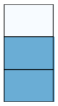
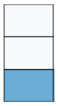
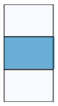
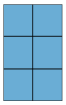
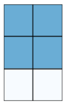
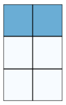
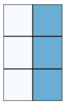
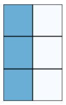
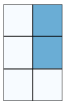
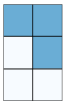
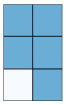
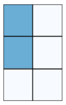
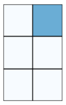
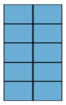
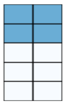
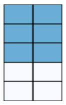
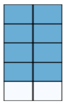
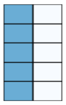
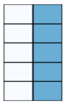
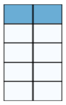
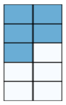
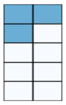
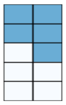
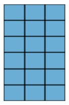
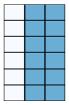
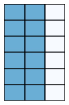
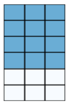
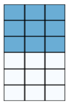
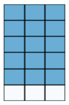
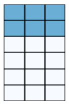
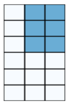
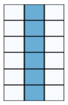
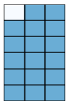
| $M \times N$ | Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $1 \times 3$ | Part<br>$(n = 7)$ |  |  |  |  |  |  |  | | | |
| | Freq. | 11981 | 1823 | 1172 | 197 | 144 | 32 | 22 | | | |
| | Percent. | 77.95 | 11.86 | 7.62 | 1.28 | 0.94 | 0.21 | 0.14 | | | |
| | Cum. | 77.95 | 89.81 | 97.43 | 98.71 | 99.65 | 99.86 | 100.0 | | | |
| $2 \times 3$ | Part<br>$(n = 30)$ |  |  |  |  |  |  |  |  |  |  |
| | Freq. | 10933 | 1382 | 895 | 748 | 366 | 204 | 175 | 95 | 93 | 90 |
| | Percent. | 71.13 | 8.99 | 5.82 | 4.87 | 2.38 | 1.33 | 1.14 | 0.62 | 0.61 | 0.59 |
| | Cum. | 71.13 | 80.12 | 85.94 | 90.81 | 93.19 | 94.52 | 95.65 | 96.27 | 96.88 | 97.46 |
| $2 \times 5$ | Part<br>$(n = 57)$ |  |  |  |  |  |  |  |  |  |  |
| | Freq. | 10607 | 928 | 749 | 690 | 601 | 332 | 144 | 138 | 109 | 80 |
| | Percent. | 69.01 | 6.04 | 4.87 | 4.49 | 3.91 | 2.16 | 0.94 | 0.90 | 0.71 | 0.52 |
| | Cum. | 69.01 | 75.04 | 79.92 | 84.41 | 88.32 | 90.48 | 91.41 | 92.31 | 93.02 | 93.54 |
| $3 \times 6$ | Part<br>$(n = 163)$ |  |  |  |  |  |  |  |  |  |  |
| | Freq. | 8865 | 1624 | 631 | 537 | 489 | 336 | 311 | 243 | 205 | 183 |
| | Percent. | 57.67 | 10.57 | 4.11 | 3.49 | 3.18 | 2.19 | 2.02 | 1.58 | 1.33 | 1.19 |
| | Cum. | 57.67 | 68.24 | 72.34 | 75.84 | 79.02 | 81.20 | 83.23 | 84.81 | 86.14 | 87.33 |

Table 5: Distribution of *part confidence maps* on CityPersons *train* dataset. $n$ is the total number of parts. The *blue* areas represent visible parts.

| Method | | Height $\geq$ 50 | | | | |
|---|---|---|---|---|---|---|
| Part score | Structure | Reasonable | All | None | Partial | Heavy |
| Baseline | | 23.37 | 32.83 | 21.58 | 36.07 | 63.65 |
| Max | $(1 \times 3)$ | 23.47 | 33.53 | 21.10 | 38.43 | 65.87 |
| | $(2 \times 3)$ | 22.07 | 32.40 | 19.51 | 38.37 | 65.83 |
| | $(2 \times 5)$ | 22.08 | 30.30 | 19.46 | 40.14 | **56.60** |
| | $(3 \times 6)$ | 22.92 | 32.15 | 21.01 | 33.65 | 62.35 |
| Soft | $(2 \times 5)$ - 6 | **20.30** | 31.80 | 18.45 | **33.28** | 68.25 |
| | $(3 \times 6)$ - 6 | 22.56 | 31.68 | 20.52 | 35.88 | 60.71 |
| | $(2 \times 5)$ - 45 - 64 | 22.20 | 31.82 | 20.04 | 37.27 | 62.69 |
| | $(3 \times 6)$ - 45 - 64 | 20.78 | **30.18** | **18.76** | 34.65 | 59.87 |

Table 6: Detailed breakdown performance of occlusion handling methods at *SqueezeDet+* on Caltech *test* dataset (lower is better).

## 5. Ablation Experiments of Hard Negative Handling Methods

We perform ablation studies on grid classifiers by changing two configurations in the model. First, we change the size of convolutional filter that is used as a classifier ($1 \times 1$ and $3 \times 3$). Second, we compare the performances of different uses of the grid confidence map as follows.

- *Baseline*: The results of base models.
- *Loss only*: The result of using the grid confidence map for training.

| Method | | Height ≥ 50 | | | | |
|---|---|---|---|---|---|---|
| Part score | Structure | Reasonable | All | None | Partial | Heavy |
| Baseline | | 20.83 | 29.35 | 18.97 | 34.37 | 57.55 |
| Max | (1 × 3) | 18.33 | 28.02 | 16.58 | **30.60** | 60.93 |
| | (2 × 3) | 20.74 | 29.03 | 18.75 | 33.82 | 58.43 |
| | (2 × 5) | 19.31 | 27.56 | 17.40 | 31.69 | **53.90** |
| | (3 × 6) | 20.58 | 31.11 | 18.52 | 33.71 | 67.05 |
| Soft | (2 × 5) - 6 | 18.91 | 28.17 | 16.90 | 31.26 | 60.07 |
| | (3 × 6) - 6 | **18.29** | **27.16** | **16.12** | 31.94 | 57.02 |
| | (2 × 5) - 45 - 64 | 18.77 | 28.51 | 16.83 | 30.61 | 61.78 |
| | (3 × 6) - 45 - 64 | 18.98 | 27.93 | 16.51 | 33.85 | 57.42 |

Table 7: Detailed breakdown performance of occlusion handling methods at *YOLOv2* on Caltech *test* dataset (lower is better).

| Method | | Height ≥ 50 | | | | | Height ≥ 20 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Part score | Structure | Reasonable | All | None | Partial | Heavy | All | None | Partial | Heavy |
| Baseline | | 16.36 | 25.18 | 14.55 | 27.89 | 53.80 | 60.19 | 52.21 | **67.96** | 76.47 |
| Max | (1 × 3) | 15.81 | 24.09 | 13.86 | 29.40 | 51.18 | 60.18 | 53.02 | 70.23 | 74.71 |
| | (2 × 3) | 16.19 | 23.87 | 14.86 | 30.03 | **47.68** | 59.53 | 52.71 | 69.63 | **73.11** |
| | (2 × 5) | 15.60 | 23.70 | 13.69 | 27.85 | 50.02 | 59.94 | 53.07 | 69.92 | 73.14 |
| | (3 × 6) | 16.16 | 24.84 | 14.27 | **25.30** | 53.19 | 60.19 | 51.92 | 70.54 | 77.24 |
| Soft | (2 × 5) - 6 | 15.56 | 24.37 | 13.15 | 30.99 | 54.41 | 59.83 | 52.73 | 69.48 | 75.11 |
| | (3 × 6) - 6 | 14.57 | 23.83 | 12.57 | 27.80 | 53.94 | 59.23 | 51.05 | 69.28 | 77.53 |
| | (2 × 5) - 45 - 64 | 15.50 | 23.76 | 13.56 | 28.34 | 51.34 | 59.00 | 51.31 | 68.93 | 75.29 |
| | (3 × 6) - 45 - 64 | **14.23** | **22.53** | **12.22** | 27.52 | 50.46 | **58.94** | **51.71** | 68.85 | 74.37 |

Table 8: Detailed breakdown performance of occlusion handling methods at *SSD* on Caltech *test* dataset (lower is better).

| Method | | Height ≥ 50 | | | | | Height ≥ 20 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Part score | Structure | Reasonable | All | None | Partial | Heavy | All | None | Partial | Heavy |
| Baseline | | 13.25 | 20.53 | 11.23 | 25.23 | 44.13 | 53.03 | 44.72 | 64.15 | 69.59 |
| Max | (1 × 3) | 12.72 | 20.23 | 10.72 | 25.80 | 44.81 | 52.64 | 44.42 | 63.77 | **69.57** |
| | (2 × 3) | 13.04 | 22.44 | 10.82 | 29.42 | 53.66 | 54.39 | 45.16 | 65.18 | 74.60 |
| | (2 × 5) | 12.01 | 20.92 | 10.36 | 22.40 | 49.82 | 53.05 | 43.46 | 62.60 | 74.70 |
| | (3 × 6) | 13.01 | 20.52 | 11.36 | 23.83 | 44.44 | 53.19 | 44.48 | 62.96 | 71.47 |
| Soft | (2 × 5) - 6 | 11.60 | 19.87 | 9.78 | **22.12** | 46.75 | 51.96 | 43.36 | **60.79** | 70.80 |
| | (3 × 6) - 6 | 11.84 | 20.28 | 10.12 | 24.66 | 47.49 | 52.35 | 43.22 | 62.89 | 72.90 |
| | (2 × 5) - 45 - 64 | **10.97** | **18.58** | **8.88** | 26.14 | **44.11** | **50.55** | **41.51** | 61.68 | 69.65 |
| | (3 × 6) - 45 - 64 | 11.99 | 20.63 | 10.06 | 25.49 | 49.33 | 52.91 | 43.82 | 63.98 | 73.53 |

Table 9: Detailed breakdown performance of occlusion handling methods at *DSSD* on Caltech *test* dataset (lower is better).

- *Adjustment*: The result of using the grid confidence map for training and refining the initial confidence.

The parameters of the models used by *loss only* and *adjustment* are the same. The only difference is whether to adjust the initial confidence using the predicted grid confidence map.

Table 10 shows the overall results of ablation experiments of hard negative handling methods. The performance of the *loss only* is always better than the *baseline*. However, in case of *adjustment*, the results are different depending on the base model. The *adjustment* performs the best in SqueezeDet+ and YOLOv2, but the worst in the SSD and DSSD (even worse than the baseline). In the main draft, we mentioned about the two intuitions of why the grid classifiers help improve the performance: i) the refinement by the averaged results from multiple feature maps, and ii) resolving the mismatch between a predicted box and its feature representation in the base models. The SSD and DSSD have layers that care for the object scales; that is, the grid feature representations of the ground truth and its anchor are not significantly mismatched each other because of their similar scales. Therefore, the second effect is not much significant in SSD and DSSD.

| Model | Height $\geq 50$ | | | | | Height $\geq 20$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Reasonable | All | None | Partial | Heavy | All | None | Partial | Heavy |
| SqueezeDet+ [13] | 23.47 | 32.88 | 21.69 | 34.05 | 62.96 | – | – | – | – |
| + 1 × 1 (*loss only*) | 20.87 | 29.23 | 18.88 | 34.20 | 56.56 | – | – | – | – |
| + 1 × 1 (*adjustment*) | **19.58** | **28.72** | **17.79** | **29.68** | **56.53** | – | – | – | – |
| + 3 × 3 (*loss only*) | 21.36 | 31.56 | 19.83 | 30.44 | 65.51 | – | – | – | – |
| + 3 × 3 (*adjustment*) | 20.61 | 30.48 | 18.96 | 30.21 | 62.42 | – | – | – | – |
| YOLOv2 [9] | 20.83 | 29.35 | 18.97 | 34.37 | **57.55** | – | – | – | – |
| + 1 × 1 (*loss only*) | 18.66 | 28.79 | 16.74 | 31.12 | 62.27 | – | – | – | – |
| + 1 × 1 (*adjustment*) | **16.92** | **27.65** | **14.95** | **27.44** | 63.57 | – | – | – | – |
| + 3 × 3 (*loss only*) | 19.88 | 28.54 | 18.24 | 31.94 | 57.76 | – | – | – | – |
| + 3 × 3 (*adjustment*) | 18.80 | 27.86 | 17.06 | 29.46 | 57.61 | – | – | – | – |
| SSD [8] | 16.36 | 25.18 | 14.55 | 27.89 | 53.80 | 60.19 | 52.21 | **67.96** | 76.47 |
| + 1 × 1 (*loss only*) | 14.92 | **23.68** | 12.90 | 27.90 | **52.93** | **59.15** | 51.84 | 69.98 | **73.61** |
| + 1 × 1 (*adjustment*) | 16.94 | 26.20 | 14.90 | 28.20 | 54.51 | 61.46 | 54.53 | 70.18 | 74.16 |
| + 3 × 3 (*loss only*) | **14.04** | 23.79 | **12.03** | **26.52** | 55.10 | 59.66 | **51.60** | 68.93 | 76.04 |
| + 3 × 3 (*adjustment*) | 16.51 | 27.17 | 14.39 | 26.93 | 57.51 | 62.37 | 54.46 | 68.32 | 76.04 |
| DSSD [5] | 13.25 | 20.53 | 11.23 | 25.23 | 44.13 | 53.03 | 44.72 | 64.15 | 69.59 |
| + 1 × 1 (*loss only*) | 11.83 | 19.95 | 9.90 | 26.41 | 47.51 | 50.11 | **41.19** | **60.56** | 69.87 |
| + 1 × 1 (*adjustment*) | 15.28 | 23.74 | 13.19 | 26.80 | 48.73 | 55.08 | 46.95 | 61.67 | 70.17 |
| + 3 × 3 (*loss only*) | **10.85** | **18.20** | **9.00** | 24.28 | 42.42 | **49.24** | 41.32 | 60.74 | 65.99 |
| + 3 × 3 (*adjustment*) | 14.40 | 21.69 | 12.83 | **22.50** | **42.34** | 54.07 | 47.31 | 61.06 | **65.07** |

Table 10: Detailed breakdown performance of hard negative handling methods on Caltech *test* dataset (lower is better).

## 6. Comparison with State-of-the-art Models

The goal of this paper is to propose a lightweight approach that is applicable to single-stage detection models for improving their occlusion and hard negative handling. We do not argue that our approach is integrable with any detection models, but limited to single-stage models, which are often inferior to the two-stage models in performance, but are much faster and lighter. Therefore, we focus on improving the performance of base networks, instead of comparing with state-of-the-art methods. Our final detection accuracies depend on those of base models; thus if the base model is competitive, our method is as well.

Table 11 shows the performance comparison with state-of-the-art models on Caltech *test* dataset. Encouragingly, in some settings (*all*/*heavy* with Height $\geq 50$ and $\geq 20$), our approach with DSSD achieves the best as in Table 11.

| Model | Height $\geq 50$ | | | | | Height $\geq 20$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Reasonable | All | None | Partial | Heavy | All | None | Partial | Heavy |
| DeepParts [11] | 11.89 | 22.79 | 10.64 | 19.93 | 60.42 | 64.78 | 58.43 | 70.39 | 81.81 |
| MS-CNN [1] | 9.95 | 21.53 | 8.15 | 19.24 | 59.94 | 60.95 | 53.67 | 67.16 | 79.51 |
| RPN+BF [14] | 9.58 | 24.01 | 7.68 | 24.23 | 69.91 | 64.66 | 56.38 | 72.55 | 87.48 |
| F-DNN [3] | 8.65 | 19.31 | 7.10 | 15.41 | 55.13 | 50.55 | 40.29 | 60.60 | 76.98 |
| F-DNN+SS [3] | **8.18** | 18.82 | **6.74** | **15.11** | 53.67 | 50.29 | **40.21** | **60.08** | 75.77 |
| DSSD [5] + Ours | 10.85 | **18.20** | 9.00 | 24.28 | **42.42** | **49.24** | 41.32 | 60.74 | **65.99** |

Table 11: Comparison with state-of-the-art models on Caltech *test* dataset (lower is better).

As of now, the best-performing model on CityPersons is RepLoss [12], which is also a two-stage model taking advantage over our single-stage base models. Nonetheless, as shown in Table 12, our approach with DSSD outperforms RepLoss in the *partial* and *heavy* setting.

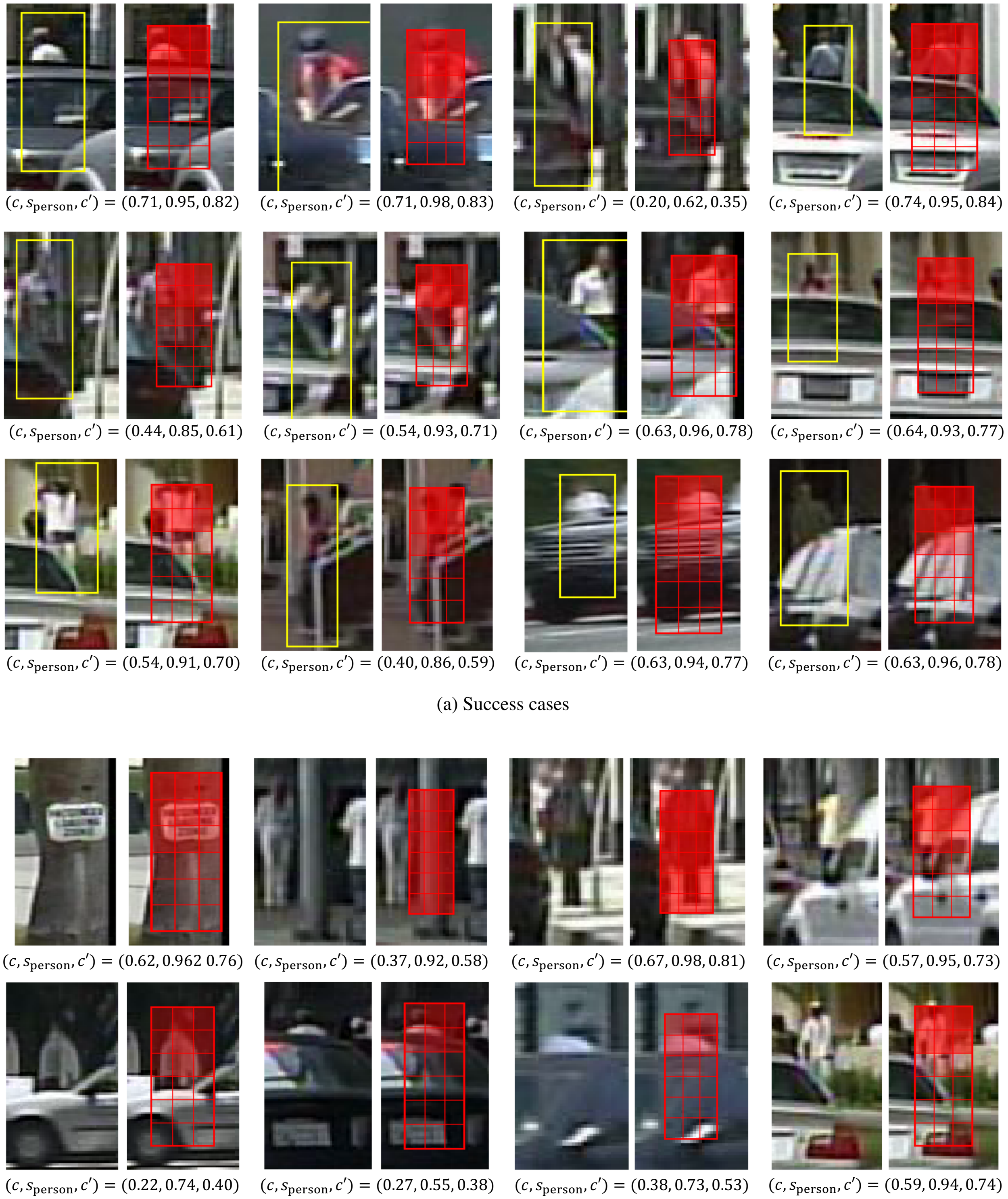
## 7. More Qualitative Results

Figure 1 shows more examples of occlusion handling. Many success examples (Figure 1a) are the cases whose visible areas are upper parts. The models can easily detect those examples, mainly because much occlusion in the training set is such cases as discussed in Section 2. The failure cases (Figure 1b) include hard negative and mislabelled examples. We can see

| Model | Reasonable | All | None | Partial | Heavy |
|---|---|---|---|---|---|
| RepLoss [12] | **13.20** | N/A | **7.60** | 16.80 | 56.90 |
| DSSD [5] + Ours | 16.77 | 31.71 | 11.15 | **16.05** | **48.52** |

Table 12: Comparison with the state-of-the-art model on CityPersons *val* set (lower is better).

the effect of grid classifiers in Figure 2. All examples are adjusted in the right way by the grid classifiers. We can also check the robustness of using multiple layers. If one of the layers predicts its confidence map incorrectly as the initial confidence, the other layer can refine its value.

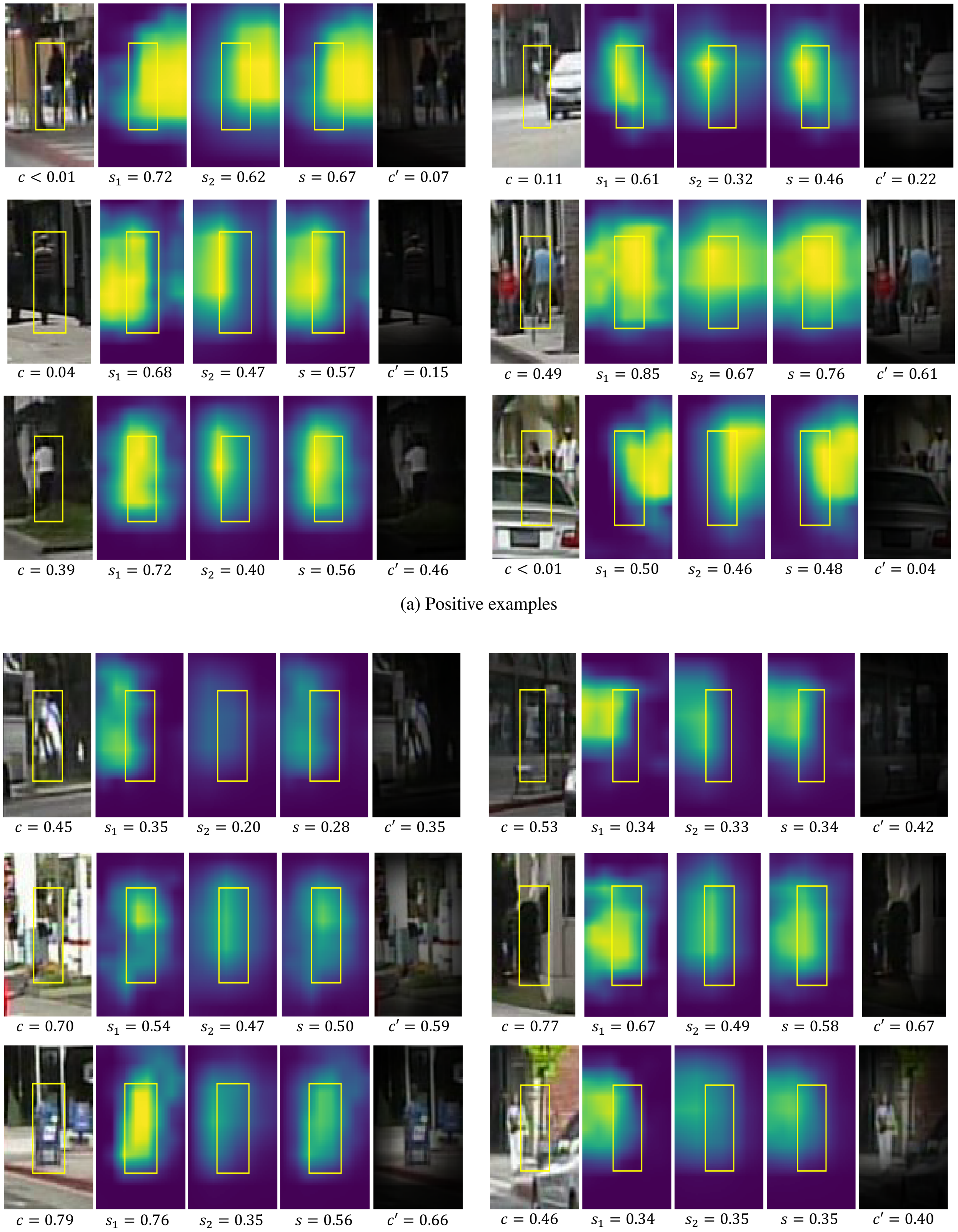

$(c, s_{\text{person}}, c') = (0.71, 0.95, 0.82)$
$(c, s_{\text{person}}, c') = (0.71, 0.98, 0.83)$
$(c, s_{\text{person}}, c') = (0.20, 0.62, 0.35)$
$(c, s_{\text{person}}, c') = (0.74, 0.95, 0.84)$
$(c, s_{\text{person}}, c') = (0.44, 0.85, 0.61)$
$(c, s_{\text{person}}, c') = (0.54, 0.93, 0.71)$
$(c, s_{\text{person}}, c') = (0.63, 0.96, 0.78)$
$(c, s_{\text{person}}, c') = (0.64, 0.93, 0.77)$
$(c, s_{\text{person}}, c') = (0.54, 0.91, 0.70)$
$(c, s_{\text{person}}, c') = (0.40, 0.86, 0.59)$
$(c, s_{\text{person}}, c') = (0.63, 0.94, 0.77)$
$(c, s_{\text{person}}, c') = (0.63, 0.96, 0.78)$

(a) Success cases

$(c, s_{\text{person}}, c') = (0.62, 0.962\ 0.76)$
$(c, s_{\text{person}}, c') = (0.37, 0.92, 0.58)$
$(c, s_{\text{person}}, c') = (0.67, 0.98, 0.81)$
$(c, s_{\text{person}}, c') = (0.57, 0.95, 0.73)$
$(c, s_{\text{person}}, c') = (0.22, 0.74, 0.40)$
$(c, s_{\text{person}}, c') = (0.27, 0.55, 0.38)$
$(c, s_{\text{person}}, c') = (0.38, 0.73, 0.53)$
$(c, s_{\text{person}}, c') = (0.59, 0.94, 0.74)$

(b) Failure cases

Figure 1: More examples of occlusion handling. For better visualization, we crop detection regions from images.



(a) Positive examples

(b) Negative examples

Figure 2: More examples of adjustment by grid classifiers. For better visualization, we crop detection regions from images.

## References

[1] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In *ECCV*, 2016.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[3] X. Du, M. El-Khamy, J. Lee, and L. Davis. Fused DNN: A Deep Neural Network Fusion Approach to Fast and Robust Pedestrian Detection. In *IEEE WACV*. IEEE, 2017.

[4] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[5] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. DSSD: Deconvolutional Single Shot Detector. *arXiv:1701.06659*, 2017.

[6] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and $< 0.5$ MB Model Size. *arXiv:1602.07360*, 2016.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. In *ECCV*, 2016.

[9] J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *CVPR*, 2017.

[10] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*, 2014.

[11] Y. Tian, P. Luo, X. Wang, and X. Tang. Deep Learning Strong Parts for Pedestrian Detection. In *ICCV*, 2015.

[12] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen. Repulsion Loss: Detecting Pedestrians in a Crowd. *arXiv:1711.07752*, 2017.

[13] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. *arXiv:1612.01051*, 2016.

[14] L. Zhang, L. Lin, X. Liang, and K. He. Is Faster R-CNN Doing Well for Pedestrian Detection? In *ECCV*, 2016.