Precise Detection in Densely Packed Scenes

Eran Goldman[1,3⋆] Roei Herzig[2⋆] Aviv Eisenschtat[3⋆] Jacob Goldberger[1] Tal Hassner[4†]
[1]Bar-Ilan University, [2]Tel Aviv University, [3]Trax Retail, [4]The Open University of Israel

Abstract

Man-made scenes can be densely packed, containing numerous objects, often identical, positioned in close proximity. We show that precise object detection in such scenes remains a challenging frontier even for state-of-the-art object detectors. We propose a novel, deep-learning based method for precise object detection, designed for such challenging settings. Our contributions include: (1) A layer for estimating the Jaccard index as a detection quality score; (2) a novel EM merging unit, which uses our quality scores to resolve detection overlap ambiguities; finally, (3) an extensive, annotated data set, SKU-110K, representing packed retail environments, released for training and testing under such extreme settings. Detection tests on SKU-110K and counting tests on the CARPK and PUCPR+ show our method to outperform existing state-of-the-art with substantial margins. The code and data will be made available on `www.github.com/eg4000/SKU110K_CVPR19`.

1. Introduction

Recent deep learning–based detectors can quickly and reliably detect objects in many real world scenes [15, 16, 19, 27, 30, 36, 37, 38]. Despite this remarkable progress, the common use case of detection in crowded images remains challenging even for leading object detectors.

We focus on detection in such *densely packed* scenes, where images contain many objects, often looking similar or even identical, positioned in close proximity. These scenes are typically man-made, with examples including retail shelf displays, traffic, and urban landscape images. Despite the abundance of such environments, they are under-represented in existing object detection benchmarks. It is therefore unsurprising that state-of-the-art object detectors are challenged by such images.

To understand what makes these detection tasks difficult, consider two identical objects placed in immediate proximity, as is often the case for items on store shelves (Fig. 1).

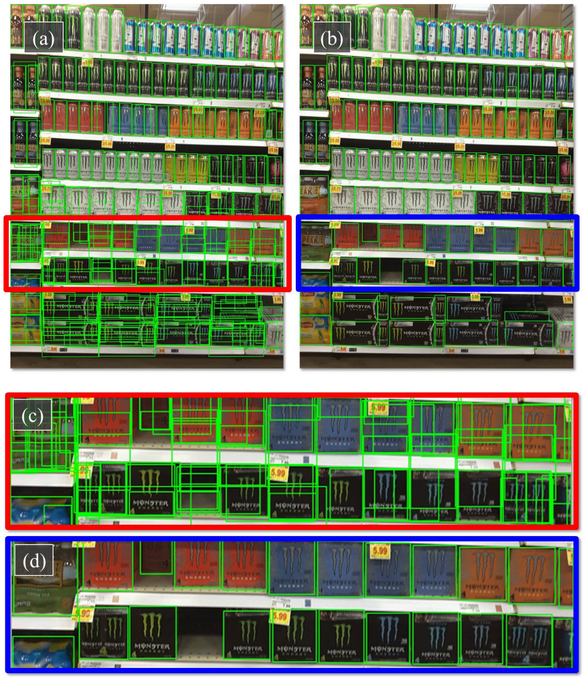

Figure 1. **Detection in packed domains.** A typical image in our SKU-110K, showing densely packed objects. (Top) (a) Detection results for the state-of-the-art RetinaNet [27], showing incorrect and overlapping detections, especially for the dark objects at the bottom which are harder to separate. (b) Our results showing far fewer misdetections and better fitting bounding boxes. (Bottom) Zoomed-in views for (c) RetinaNet [27] and (d) our method.

The challenge is to determine where one object ends and the other begins; minimizing overlaps between their adjacent bounding boxes. In fact, as we show in Fig. 1(a,c), the state-of-the-art RetinaNet detector [27], often returns bounding boxes which partially overlap *multiple objects* or detections of adjacent object regions as separate objects.

We describe a method designed to accurately detect objects, even in such densely packed scenes (Fig. 1(b,d)). Our method includes several innovations. We propose learning the Jaccard index with a *soft Intersection over Union (Soft-IoU)* network layer. This measure provides valuable information on the quality of detection boxes. We explain how detections can be represented as a *Mixture of Gaussians* (MoG), reflecting their locations and their Soft-IoU

[⋆] Equal Contribution.
[†] Work done while at the University of Southern California.

Name	**#Img.**	**#Obj./img.**	**#Cls.**	**#Cls./img.**	**Dense.**	**Idnt.**	**BB**
UCSD (2008) [8]	2000	24.9	1	1	✓	✗	✗
PACAL VOC (2012) [13]	22,531	2.71	20	2	✗	✗	✓
ILSVRC Detection (2014) [12]	516,840	1.12	200	2	✗	✗	✓
COCO (2015) [28]	328,000	7.7	91	3.5	✗	✗	✓
Penguins (2016) [2]	82,000	25	1	1	✓	✗	✗
TRANCOS (2016) [34]	1,244	37.61	1	1	✓	✓	✗
WIDER FACE (2016) [49]	32,203	12	1	1	✗	✗	✓
CityPersons (2017) [51]	5000	6	1	1	✗	✗	✓
PUCPR+ (2017) [22]	125	135	1	1	✓	✓	✓
CARPK (2018) [22]	1448	61	1	1	✓	✓	✓
Open Images V4 (2018) [25]	**1,910,098**	8.4	600	2.3	✗	✓	✓
Our SKU-110K	11,762	**147.4**	**110,712**	**86**	✓	✓	✓

Table 1. **Key properties for related benchmarks. #Img.**: Number of images. **#Obj./img.**: Average items per image. **#Cls.**: Number of object classes (more implies a harder detection problem due to greater appearance variations). **#Cls./img.**: Average classes per image. **Dense**: Are objects typically densely packed together, raising potential overlapping detection problems? **Idnt**: Do images contain multiple identical objects or hard to separate object sub-regions? **BB**: Bounding box labels available for measuring detection accuracy?

scores. An Expectation-Maximization (EM) based method is then used to cluster these Gaussians into groups, resolving detection overlap conflicts.

To summarize, our novel contributions are as follows:

- **Soft-IoU layer**, added to an object detector to estimate the Jaccard index between the detected box and the (unknown) ground truth box (Sec. 3.2).
- **EM-Merger unit**, which converts detections and Soft-IoU scores into a MoG, and resolves overlapping detections in packed scenes (Sec. 3.3).
- **A new data set and benchmark**, the store keeping unit, 110k categories (SKU-110K), for item detection in store shelf images from around the world (Sec. 4).

We test our detector on SKU-110K. Detection results show our method to outperform state-of-the-art detectors. We further test our method on the related but different task of *object counting*, on SKU-110K and the recent CARPK and PUCPR+ car counting benchmarks [22]. Remarkably, although our method was not designed for counting, it offers a considerable improvement over state-of-the-art methods.

2. Related work

Object detection. Work on this problem is extensive and we refer to a recent survey for a comprehensive overview [29]. Briefly, early detectors employed sliding window–based approaches, applying classifiers to window contents at each spatial location [10, 14, 45]. Later methods narrow this search space by determining *region proposals* before applying sophisticated classifiers [1, 7, 35, 44, 52].

Deep learning–based methods now dominate detection results. To speed detection, proposal-based detectors such as R-CNN [15] and Fast R-CNN [16] were developed, followed by Faster R-CNN [38] which introduced a *region proposal network* (RPN), then accelerated even more by R-FCN [9]. Mask-RCNN [19] later added segmentation output and better detection pooling [38]. We build on these methods, claiming no advantage in standard object detection tasks. Unlike us, however, these *two-stage* methods were not designed for crowded scenes where small objects appear in dense formations.

Recently, some offered *proposal-free* detectors, including YOLO [36], SSD [30], and YOLO9000 [37]. To handle scale variance, feature pyramid network (FPN) [26] added up-scaling layers. RetinaNet [27] utilized the same FPN model, introducing a *Focal Loss* to dynamically weigh hard and easy samples for better handling of class imbalances that naturally occur in detection datasets. We extend this approach, introducing a new detection overlap measure, allowing for precise detection of tightly packed objects.

These methods use hard-labeled log-likelihood detections to produce confidences for each candidate image region. We additionally predict a Soft-IoU confidence score which represents detection bounding box accuracy.

Merging duplicate detections. Standard *non-maximum suppression* (NMS) remains a *de-facto* object detection duplicate merging technique, from Viola & Jones [45] to recent deep detectors [27, 37, 38]. NMS is a hand-crafted algorithm, applied at test time as post-processing, to greedily select high scoring detections and remove their overlapping, low confidence neighbors.

Existing NMS alternatives include mean-shift [10, 46], agglomerative [5], and affinity propagation clustering [31], or heuristic variants [4, 40, 23]. GossipNet [21] proposed to perform duplicate-removal using a learnable layer in the detection network. Finally, others bin IoU values into five categories [43]. We instead take a probabilistic interpretation of IoU prediction and a very different general approach.

Few of these methods showed improvement over simple, greedy NMS, with some also being computationally demanding [21]. In densely packed scenes, resolving detec-

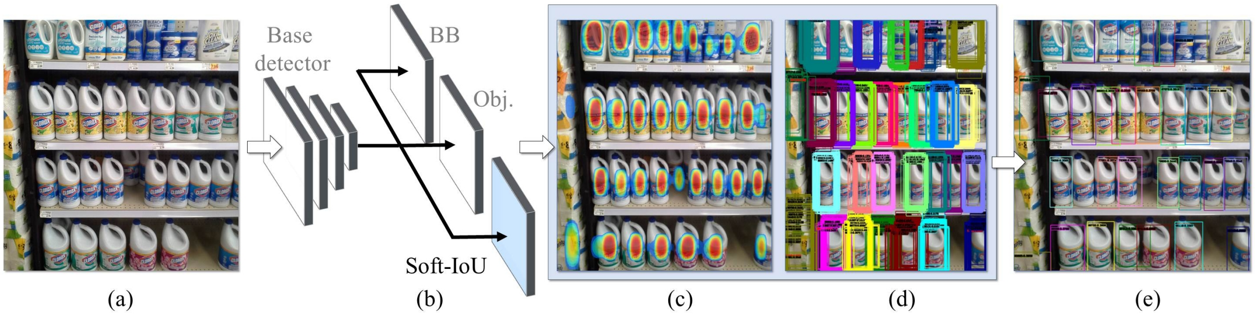

Figure 2. **System diagram.** (a) Input image. (b) A base network, with bounding box (BB) and objectness (Obj.) heads (Sec. 3.1), along with our novel Soft-IoU layer (Sec. 3.2). (c) Our EM-Merger converts Soft-IoU to Gaussian heat-map representing (d) objects captured by multiple, overlapping bounding boxes. (e) It then analyzes these box clusters, producing a single detection per object (e) (Sec. 3.3).

tion ambiguities is exacerbated due to the many overlapping detections. We propose an unsupervised method, designed for clustering duplicate detection in cluttered regions.

Crowded scene benchmarks. Many benchmarks were designed for testing object detection or counting methods and we survey a few in Table 1. Importantly, we are unaware of detection benchmarks intended for densely packed scenes, such as those of interest here.

Popular object detection sets include ILSVRC [12], PASCAL VOC [13] detection challenges, MS COCO [28], and the very recent Open Images v4 [25]. None of these provides scenes with packed items. A number of recent benchmarks emphasize crowded scenes, but are designed for counting, rather than detection [2, 8, 34].

As evident from Table 1, our new SKU-110K dataset, described in Sec. 4, provides *one to three orders of magnitude more items per image* than nearly all these benchmarks (the only exception is the PUCPR+ [22] which offers two orders of magnitude *fewer* images, and a single object class to our more than 110k classes). Most importantly, our enormous, per image, object numbers imply that all our images contain very crowded scenes, which raises the detection challenges described in Sec. 1. Moreover, identical or near identical items in SKU-110K are often positioned closely together, making detection overlaps a challenge. Finally, the large number of classes in SKU-110K implies appearance variations which add to the difficulty of this benchmark, even in challenges of object/non-object detection.

3. Deep IoU detection network

Our approach is illustrated in Fig. 2. We build on a standard detection network design, described in Sec. 3.1. We extend this design in two ways. First, we define a novel Soft-IoU layer which estimates the overlap between predicted bounding boxes and the (unknown) ground truth (Sec. 3.2). These Soft-IoU scores are then processed by a proposed EM-Merger unit, described in Sec. 3.3, which resolves ambiguities between overlapping bounding boxes, returning a single detection per object.

3.1. Base detection network

Our base detector is similar to existing methods [26, 27, 30, 38]. We first detect objects by building a FPN network [26] with three upscaling-layers, using ResNet-50 [20] as a backbone. The proposed model provides three fully-convolutional output heads for each RPN [38]: Two heads are standard and used also by previous work [27, 37] (our novel third head is described in Sec. 3.2).

The first is a *detection head* which produces a bounding box regression output for each object, represented as 4-tuples: (x, y, h, w) for the 2D coordinates of a bounding box center, height and width. The second, *classification head* provides an *objectness* score (confidence) label, $c \in [0, 1]$ (assuming an object/no-object detection task with one object class). In practice, we filter detections for which $c \leq 0.1$, to avoid creating a bias towards noisy detections when training our Soft-IoU layer, described next.

3.2. Soft-IoU layer

In non-dense scenes, greedy NMS applied to objectness scores, c, can resolve overlapping detections. In dense images, however, multiple overlapping bounding boxes often reflect multiple, tightly packed objects, many of which receive high objectness scores. As we later show (Sec. 5.2), in such cases, NMS does not adequately discriminate between overlapping detections or suppress partial detections.

To handle these cluttered positive detections, we propose predicting an additional value for each bounding box: The IoU (*i.e.*, Jaccard index) between a regressed detection box and the object location. This Soft-IoU score, $c^{iou} \in [0, 1]$, is estimated by a fully-convolutional layer which we add as a third head to the end of each RPN in the detector.

Given N predicted detections, the IoU between a predicted bounding box $\mathbf{b}_i$, $i \in \{1..N\}$ and its ground truth bounding box, $\hat{\mathbf{b}}_i$, is defined as:

$$IoU_i = \frac{Intersection(\hat{\mathbf{b}}_i, \mathbf{b}_i)}{Union(\hat{\mathbf{b}}_i, \mathbf{b}_i)}. \quad (1)$$

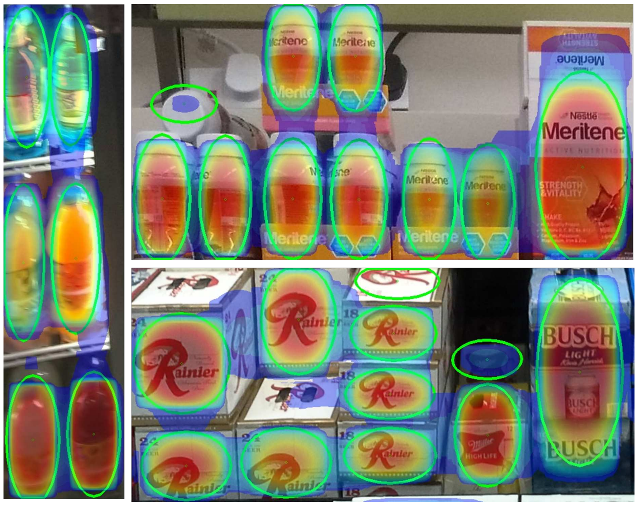

Figure 3. **Visualizing the output of the EM-Merger unit.** Raw detections on these images (not shown) contain many overlapping bounding boxes. Our approach of representing detections as a MoG (Eq. (5)), visualized here as heat maps, provides clear signals for where items are located. The simplified MoG of Eq. (7) is visualized as green ellipsoids. See Sec. 3.3 for details.

We chose $\hat{\mathbf{b}}_i$ to be the closest annotated box to $\mathbf{b}_i$ (in image coordinates). If the two do not overlap, then $IoU_i = 0$. Both $Intersection(\cdot)$ and $Union(\cdot)$ count pixels.

We take a probabilistic interpretation of Eq. (1), learning it with our Soft-IoU layer using a binary cross-entropy loss:

$$\mathcal{L}_{\text{sIoU}} = \qquad (2)$$

$$-\frac{1}{n}\sum_{i=1}^{n}[IoU_i \log{(c_i^{iou})} + (1-IoU_i)\log{(1-c_i^{iou})}],$$

where n is the number of samples in each batch.

The loss used to train each RPN in the detection network is therefore defined as:

$$\mathcal{L} = \mathcal{L}_{\text{Classification}} + \mathcal{L}_{\text{Regression}} + \mathcal{L}_{\text{sIoU}}. \qquad (3)$$

Here, $\mathcal{L}_{\text{Classification}}$ and $\mathcal{L}_{\text{Regression}}$ are the standard cross-entropy and euclidean losses, respectively [16, 36, 38], and $\mathcal{L}_{\text{sIoU}}$ is defined in Eq. (2).

Objectness vs. Soft-IoU. The objectness score used in previous methods predicts object/no-object labels whereas our Soft-IoU predicts the IoU of a detected bounding box and its ground truth. So, for instance, a bounding box which partially overlaps an object can still have a high objectness score, c, signifying high confidence that the object appears in the bounding box. For the same detection, we expect c^{iou} to be low, due to the partial overlap.

In fact, object/no-object classifiers are trained to be invariant to occlusions and translations. A good objectness classifier would therefore *be invariant* to the properties which our Soft-IoU layer is sensitive to. Objectness and Soft-IoU could thus be considered reflecting complementary properties of a detection bounding box.

3.3. EM-Merger unit for inference

We now have N predicted bounding box locations, each with its associated objectness, c, and Soft-IoU, c^{iou}, scores. Bounding boxes, especially in crowded scenes, often *clump* together in clusters, overlapping each other and their item locations. Our EM-Merger unit filters, merges, or splits these overlapping detection clusters, in order to resolve a single detection per object. We begin by formally defining these detection clusters.

Detections as Gaussians. We consider the N bounding boxes produced by the network as a set of 2D Gaussians:

$$\mathbf{F} = \{f_i\}_{i=1}^{N} = \{\mathcal{N}(\mathbf{p};\mu_i,\Sigma_i)\}_{i=1}^{N}, \qquad (4)$$

with $\mathbf{p} \in \mathbb{R}^2$, a 2D image coordinate. The i-th detection is thus represented by a 2D mean, the central point of the box, $\mu_i = (x_i, y_i)$, and a diagonal covariance, $\Sigma_i = [(h_i/4)^2, 0; , 0, (w_i/4)^2]$, reflecting the box size, (h_i, w_i).

We represent these Gaussians, jointly, as a single Mixture of Gaussians (MoG) density:

$$f(\mathbf{p}) = \sum_{i=1}^{N}\alpha_i f_i(\mathbf{p}), \qquad (5)$$

where the mixture coefficients, $\alpha_i = \frac{c_i^{iou}}{\sum_{k=1}^{N} c_k^{iou}}$, reflecting our confidence that the bounding box overlaps with its ground truth, are normalized to create a MoG.

Fig. 3 visualizes the density of Eq. (5) as heat-maps, translating detections into spatial region maps representing our per-pixel confidences of detection overlaps; each region weighted by the accumulated Soft-IoU.

Selecting predictions: formal definition. We next resolve our N Gaussians (detections) into precise, non-overlapping bounding box detections by using a MoG clustering method [6, 17, 18, 50].

We treat the problem of resolving the final detections as finding a set of $K << N$ Gaussians,

$$\mathbf{G} = \{g_j\}_{j=1}^{K} = \{\mathcal{N}(\mathbf{p};{\mu'}_j,{\Sigma'}_j)\}_{j=1}^{K} \qquad (6)$$

such that when aggregated, the selected Gaussians approximate the original MoG distribution f of Eq. (5), formed by all N detections. That is, if g is defined by

$$g(\mathbf{p}) = \sum_{j=1}^{K}\beta_j g_j(\mathbf{p}), \qquad (7)$$

then we seek a mixture of K Gaussians, $\mathbf{G}$, for which

$$d(f,g) = \sum_{i=1}^{N}\alpha_i \min_{j=1}^{K} \text{KL}(f_i||g_j), \qquad (8)$$

is minimized, where KL is the KL-divergence [24] used as a non-symmetric distance between two detection boxes.

An EM-approach for selecting detections. We approximate a solution to minimization of Eq. (8) using an EM-based method. The E-step assigns each box to the nearest box cluster, where box similarity is defined by a KL distance between the corresponding Gaussians. E-step assignments are defined as:

$$\pi(i) = \arg \min_{j=1}^{K} \mathrm{KL}(f_i || g_j). \tag{9}$$

The M-step then re-estimates the model parameters by:

$$\begin{aligned} \beta_j &= \sum_{i \in \pi^{-1}(j)} \alpha_i \\ \mu'_j &= \frac{1}{\beta_j} \sum_{i \in \pi^{-1}(j)} \alpha_i \mu_i \\ \Sigma'_j &= \frac{1}{\beta_j} \sum_{i \in \pi^{-1}(j)} \alpha_i \left(\Sigma_i + (\mu_i - \mu'_j)(\mu_i - \mu'_j)^\top \right). \end{aligned} \tag{10}$$

Note that these matrix computations are fast in 2D space. Moreover, all our Gaussians represent axis-aligned detection and so they all have diagonal covariances. In such cases, the KL distance between two Gaussians has a simpler form which is even more efficient to compute.

General EM theory guarantees that the iterative process described in Eq. (9)–(10), is monotonically decreasing in the value of Eq. (8) and converging to a local minimum [11]. We determine convergence when the value of Eq. (8) is smaller than $\epsilon_{EM} = 1e - 10$. We found this process to nearly always converge within ten iterations and so we set a maximum number of iterations at that number.

EM parameters are often initialized using fast clustering to prevent convergence to poor local minima. We initialize it with an agglomerative, hierarchical clustering [39], where each detection initially represents a cluster of its own and clusters are successively merged until K clusters remain.

We note in passing that there have been several recent attempts to develop deep clustering methods [47, 48]. Such methods are designed for clustering high-dimensional data, training autoencoders to map input data into a low-dimensional feature space where clustering is easier. We instead use EM, as these methods are not relevant in our settings, where the original data is two-dimensional.

Gaussians as detections. Once EM converged, the estimated Gaussians represent a set of K detections. As an upper bound for the number of detections, we use $K = size(\mathbf{I})/(\mu_w \mu_h)$, approximating the amount of non-overlapping, mean-sized boxes that fit into the image. As post-processing, we suppress less confident Gaussians which overlap other Gaussians by more than a predefined threshold. This step can be viewed as model selection and it determines the actual number of detected objects, $K' \leq K$.

To extract the final detections, for each of the K' Gaussians, we consider the ellipse at two standard deviations around its center, visualized in Fig. 3 in green. We then search the original set of N detections (Sec. 3.1) for those whose center, $\mu = (x, y)$, falls inside this ellipse. A Gaussian is converted to a detection window by taking the median dimensions of the detections in this set.

4. The SKU-110K benchmark

We assembled a new labeled data set and benchmark containing images of supermarket shelves. We focus on such retail environments for two main reasons. First, to maximize sales and store real-estate usage, shelves are regularly optimized to present many items in tightly packed, efficient arrangements [3, 33]. Our images therefore represent extreme examples of dense environments; precisely the type of scenes we are interested in.

Second, retail items naturally fall into product, brand, and sub-brand object classes. Different brands and products are designed to appear differently. A typical store can sell hundreds of products, thereby presenting a detector with many inter-class appearance variations. Sub-brands, on the other hand, are often distinguishable only by fine-grained packaging differences. These subtle appearance variations increase the range of nuisances that detectors must face (*e.g.*, spatial transformations, image quality, occlusion).

As we show in Table 1, SKU-110K is very different from existing alternatives in the numbers and density of the objects appearing in each image, the variability of its item classes, and, of course, the nature of its scenes. Example images from SKU-110K are provided in Fig. 1, 2, and 5.

Image collection. SKU-110K images were collected from thousands of supermarket stores around the world, including locations in the United States, Europe, and East Asia. Dozens of paid associates acquired our images, using their personal cellphone cameras. Images were originally taken at no less than five mega-pixel resolution but were then JPEG compressed at one megapixel. Otherwise, phone and camera models were not regulated or documented. Image quality and view settings were also unregulated and so our images represent different scales, viewing angles, lighting conditions, noise levels, and other sources of variability.

Bounding box annotations were provided by skilled annotators. We chose experienced annotators over unskilled, Mechanical Turkers, as we found the boxes obtained this way were more accurate and did not require voting schemes to verify correct annotations [28, 42]. We did, however, visually inspect each image along with its detection labels, to filter obvious localization errors.

Benchmark protocols. SKU-110K images were parti-

Method	FPS	DPS
Faster-RCNN (2015) [38]	2.37	93
YOLO9000 (2017) [37]	5	317
RetinaNet (2018) [27]	0.5	162
Base detector	0.5	162
+ Soft-IoU	0.5	162
+ EM-Merger (on the CPU)	0.23	73

Table 2. **Detection runtime comparison on SKU-110K.**

tioned into train, test, and validate splits. Training consists of 70% of the images ($8,233$ images) and their associated $1,210,431$ bounding boxes; 5% of the images (588), are used for validation (with their $90,968$ bounding boxes). The rest, $2,941$ images ($432,312$ bounding boxes) were used for testing. Images were selected at random, ensuring that the same shelf display from the same shop does not appear in more than one of these subsets.

Evaluation. We adopt evaluation metrics similar to those used by COCO [28], reporting the average precision (AP) at IoU=.50:.05:.95 (their *primary challenge metric*), AP at IoU=.75, AP$^{.75}$ (their *strict metric*), and average recall (AR)^{300} at IoU=.50:.05:.95 (300 is the maximal number of objects). We further report the value sampled from the precision-recall curve at recall $= 0.5$ for IoU=0.75 (P$^{R=.5}$).

The many, densely packed items in our images are reminiscent of the settings in counting benchmarks [2, 22]. We capture both detection and counting accuracy, by borrowing the error measures used for those tasks: If $\{K'_i\}_{i=1}^n$ is the predicted numbers of objects in each test image, $i \in [1, n]$, and $\{t_i\}_{i=1}^n$are the per image ground truth numbers, then the mean absolute error (MAE) is $\frac{1}{n}\sum_i^n |K'_i - t_i|$ and the root mean squared error (RMSE) is $\sqrt{\frac{1}{n}\sum_i^n (K'_i - t_i)^2}$.

5. Experiments

5.1. Run-time analysis

Table 2 compares average frames per second (FPS) and detections per second (DPS) for baseline methods and variations of our approach. Runtimes were measured on the same machine using an Intel(R) Core(TM) i7-5930K CPU @3.50GHz GeForce and a GTX Titan X GPU.

Our base detector is modeled after RetinaNet [27] and so their runtimes are identical. Adding our Soft-IoU layer does not affect runtime. EM-Merger is slower despite the optimizations described in Sec. 3.3, mostly because of memory swapping between GPU and CPU/RAM. Our initial tests suggest that a GPU optimized version will be nearly as fast as the base detector.

5.2. Experiments on the SKU-110K benchmark

Baseline methods. We compare the detection accuracies of our proposed method and recent state-of-the-art on the SKU-110K benchmark. All methods, with the exception of the Monkey detector, were trained on the training set portion SKU-110K.

Method	AP	AP$^{.75}$	AR^{300}	P$^{R=.5}$	MAE	RMSE
Monkey	.000	0	.010	0	N/A	N/A
Faster-RCNN [38]	.045	.010	.066	0	107.46	113.42
YOLO9000^{opt} [37]	.094	.073	.111	0	84.166	97.809
RetinaNet [27]	.455	.389	.530	.544	16.584	30.702
Base & NMS	.413	.384	.484	.491	24.962	34.382
Soft-IoU & NMS	.418	.386	.483	.492	25.394	34.729
Base & EM-Merger	.482	.540	.553	.802	23.978	283.971
Our full approach	**.492**	**.556**	**.554**	**.834**	**14.522**	**23.992**

Table 3. **Detection on SKU-110K.** Bold numbers are best results.

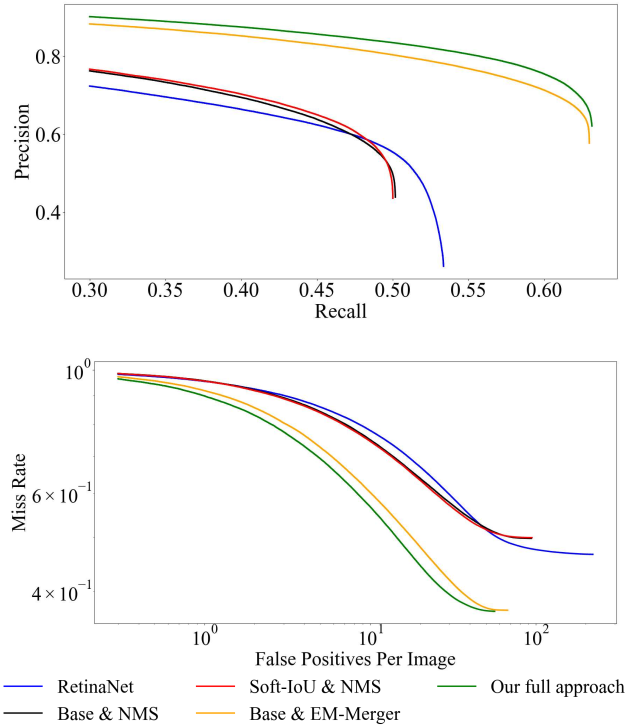

Figure 4. **Result Curves.** (a) PR Curves on SKU-110K with IoU=0.75 (*higher* curve is better). (b) The log-log curve of miss rate vs False Positives Per Image [51] (*lower* curve is better).

The following two baseline methods were tested using the original implementations released by their authors: **RetinaNet** [27] and **Faster-RCNN** [38]. YOLO9000 [37] is not suited for images with more than 50 objects. We offer results for **YOLO9000**^{opt}, which is YOLO9000 with its loss function optimized and retrained to support detection of up to 300 boxes per image.

We also report the following ablation studies, detailing the contributions of individual components of our approach.

- **Monkey:** Because of the tightly packed items in SKU-110K images, it is plausible that randomly *tossed* bounding boxes would correctly predict detections by chance. To test this naive approach, we assume we know the object number, K', the mean and standard-

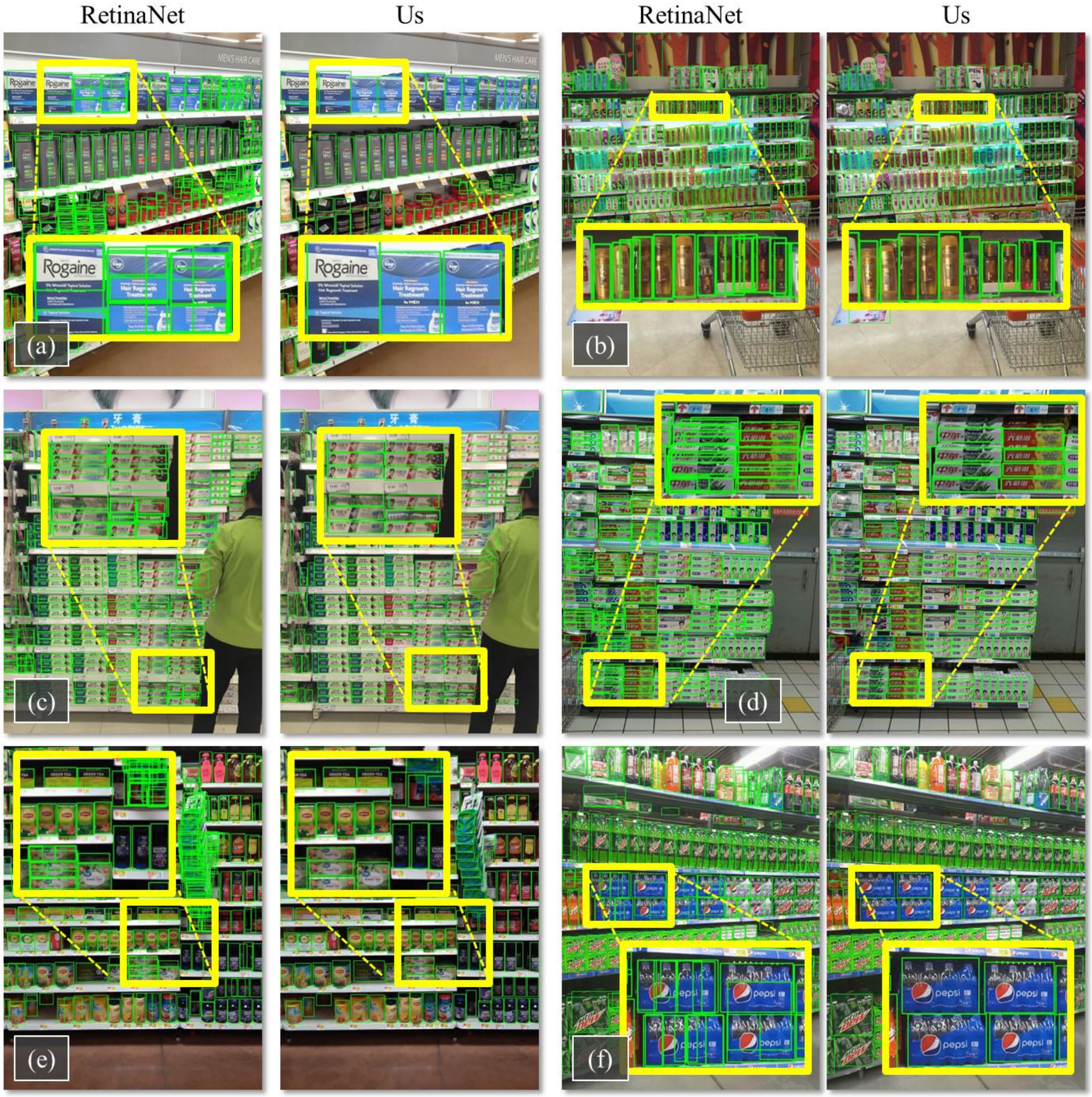

Figure 5. **Qualitative detection results on SKU-110K.** Please see project web-page for more results and images in higher resolutions.

deviation width, μ_w, σ_w, and height, μ_h, σ_h, for these boxes. Monkey samples 2D upper-left corners for the K' bounding boxes from a uniform distribution and box heights and widths from Gaussian distributions $\mathcal{N}(h;\mu_h,\sigma_h)$ and $\mathcal{N}(h;\mu_w,\sigma_w)$, respectively.

- **Base & NMS**: Our basic detector of Sec. 3.1 with standard NMS applied to objectness scores, c.
- **Soft-IoU & NMS**: Base detector with Soft-IoU (Sec. 3.2). Standard NMS applied to Soft-IoU scores, c^{iou}, instead of objectness scores.
- **Base & EM-Merger**: Our basic detector, now using EM-Merger of Sec. 3.3, but applying it to original objectness scores, c.
- **Our full approach**: Applying the EM-Merger unit to Deep-IoU scores, c^{iou}.

To test MAE and RMSE we report the number of detected objects, K', and compare it with the true number of items per image. In *RetinaNet* the number of detections is extremely high so we first filter detections with low confidences. This confidence threshold was determined using cross-validation to optimize the results of this baseline.

Detection results on SKU-110K. Quantitative detection results are provided in Table 3, result curves are presented in Fig. 4, and a selection of qualitative results, comparing

Method	MAE	RMSE
Counting results on CARPK		
Faster R-CNN (2015) [38]	24.32	37.62
YOLO (2016) [36]	48.89	57.55
One-Look Regression (2016) [32]	59.46	66.84
LPN Counting (2017) [22]	23.80	36.79
YOLO9000^{opt} (2017) [37]	45.36	52.02
RetinaNet (2018) [27] [22]	16.62	22.30
IEP Counting (2019) [41]	51.83	-
Our full approach	**6.77**	**8.52**
Counting results on PUCPR+		
Faster R-CNN (2015) [38]	39.88	47.67
YOLO (2016) [36]	156.00	200.42
One-Look Regression (2016) [32]	21.88	36.73
LPN Counting (2017) [22]	22.76	34.46
YOLO9000^{opt} (2017) [37]	130.40	172.46
RetinaNet (2018) [27]	24.58	33.12
IEP Counting (2019) [41]	15.17	-
Our full approach	**7.16**	**12.00**

Table 4. **CARPK and PUCPR+ counting results** [22].

our full approach with RetinaNet [27], the best performing baseline system, is offered in Fig. 5.

Apparently, despite the packed nature of our scenes, randomly tossing detections fails completely, as evident by the near zero accuracy of Monkey. Both Faster-RCNN [38] and YOLO9000^{opt} [37] are clearly unsuited for detecting so many tightly packed objects. RetinaNet [27], performs much better, in fact outperforming our base network despite sharing a similar design (Sec. 3.1). This could be due to the better framework optimization of RetinaNet.

Our full system outperforms all its baselines with wide margins. Much of its advantage seems to come from our EM-Merger (Sec. 3.3). Comparing the accuracy of EM-Merger applied to either objectness scores or our Soft-IoU demonstrates the added information provided by Soft-IoU. This contribution is especially meaningful when examining the counting results, which show that Soft-IoU scores provide a much better means of filtering detection boxes than objectness scores.

It is further instructional to compare detection accuracy with counting accuracy. The counting accuracy gap between our method and the closest runner up, RetinaNet, is greater than the gap in detection accuracy (though both margins are wide). The drop in counting accuracy can at least partially be explained by their use of greedy NMS compared with our EM-Merger. In fact, Fig. 5 demonstrates the many overlapping and/or mis-localized detections produced by RetinaNet compared to the single detections per item predicted by our approach (see, in particular, Fig. 5(a,e)).

Finally, we note that our best results remain far from perfect: The densely packed settings represented by SKU-110K images appear to be highly challenging, leaving room for further improvement.

5.3. Experiments on CARPK and PUCPR+

We test our method on data from other benchmarks, to see if our approach generalizes well to other domains beyond store shelves and retail objects. To this end, we use the recent CARPK and PUCPR+ [22] benchmarks. Both data sets provide images of parking lots from high vantage points. We use their test protocols, comparing the number of detections per image to the ground truth numbers made available by these benchmarks. Accuracy is reported using MAE and RMSE, as in our SKU-110K (Sec. 4).

Counting results. We compare our method with results reported by others [22, 41]: **Faster R-CNN** [38], **YOLO** [36], and **One-Look Regression** [32]. Existing baselines also include two methods designed and tested for counting on these two benchmarks: **LPN Counting** [22] and **IEP Counting** [41]. In addition, we trained and tested counting accuracy with **YOLO9000**^{opt} [37] and **RetinaNet** [27].

Table 4 reports the MAE and RMSE for all tested methods. Despite not being designed for counting, our method is more accurate than recent methods designed for that task. A significant difference between these counting datasets and our SKU-110K is in the much closer proximity of the objects in our images. This issue has a significant impact on baseline detectors, as can be seen in Tables 4 and 3. Our model suffers a much lower degradation in performance due to better filtering of these overlaps.[1]

6. Conclusions

The performance of modern object/no-object detectors on existing benchmarks is remarkable yet still limited. We focus on densely packed scenes typical of every-day retail environments and offer SKU-110K, a new benchmark of such retail shelf images, labeled with item detection boxes. Our tests on this benchmark show that such images challenge state-of-the-art detectors.

To address these challenges, along with our benchmark, we offer two technical innovations designed to raise detection accuracy in such settings: The first is a Soft-IoU layer for estimating the overlap between predicted and (unknown) ground truth boxes. The second is an EM-based unit for resolving bounding box overlap ambiguities, even in tightly packed scenes where these overlaps are common.

We test our approach on SKU-110K and two existing benchmarks for counting, and show it to surpass existing detection and counting methods. Still, even the best results on SKU-110K are far from saturated, suggesting that these densely packed scenes remain a challenging frontier for future work.

[1] See project web-page for qualitative results on these benchmarks.

References

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *Trans. Pattern Anal. Mach. Intell.*, 34(11):2189–2202, 2012.

[2] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *European Conf. Comput. Vision*, 2016.

[3] Judy Bell and Kate Ternus. *Silent selling: best practices and effective strategies in visual merchandising*. Bloomsbury Publishing USA, 2017.

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS-improving object detection with one line of code. In *Proc. Int. Conf. Comput. Vision*. IEEE, 2017.

[5] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *European Conf. Comput. Vision*. Springer, 2010.

[6] Pierrick Bruneau, Marc Gelgon, and Fabien Picarougne. Parsimonious reduction of Gaussian mixture models with a variational-bayes approach. *Pattern Recognition*, 43(3):850–858, 2010.

[7] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *Trans. Pattern Anal. Mach. Intell.*, 34(7):1312–1328, 2012.

[8] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2008.

[9] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object detection via region-based fully convolutional networks. In *Neural Inform. Process. Syst.*, 2016.

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2005.

[11] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2009.

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, 2010.

[14] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2010.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2014.

[16] Ross B. Girshick. Fast R-CNN. In *Proc. Int. Conf. Comput. Vision*. IEEE Computer Society, 2015.

[17] Jacob Goldberger, Hayit K Greenspan, and Jeremie Dreyfuss. Simplifying mixture models using the unscented transform. *Trans. Pattern Anal. Mach. Intell.*, 30(8):1496–1502, 2008.

[18] Jacob Goldberger and Sam T Roweis. Hierarchical clustering of a mixture model. In *Neural Inform. Process. Syst.*, 2005.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. Int. Conf. Comput. Vision*, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[21] Jan Hendrik Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[22] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. Drone-based object counting by spatially regularized regional proposal network. In *Proc. Int. Conf. Comput. Vision*, 2017.

[23] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *European Conf. Comput. Vision*, 2018.

[24] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[25] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018.

[26] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[27] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *Trans. Pattern Anal. Mach. Intell.*, 2018.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conf. Comput. Vision*. Springer, 2014.

[29] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018.

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *European Conf. Comput. Vision*, 2016.

[31] Damian Mrowca, Marcus Rohrbach, Judy Hoffman, Ronghang Hu, Kate Saenko, and Trevor Darrell. Spatial semantic regularisation for large scale object detection. In *Proc. Int. Conf. Comput. Vision*, 2015.

[32] T Nathan Mundhenk, Goran Konjevod, Wesam A Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *European Conf. Comput. Vision*. Springer, 2016.

[33] Jens Nordfält, Dhruv Grewal, Anne L Roggeveen, and Krista M Hill. Insights from in-store marketing experiments. In *Shopper Marketing and the Role of In-Store Marketing*, pages 127–146. Emerald Group Publishing Limited, 2014.

[34] Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *European Conf. Comput. Vision*. Springer, 2016.

[35] Esa Rahtu, Juho Kannala, and Matthew Blaschko. Learning a category independent object detection cascade. In *Proc. Int. Conf. Comput. Vision*. IEEE, 2011.

[36] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[37] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Inform. Process. Syst.*, pages 91–99. 2015.

[39] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.

[40] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[41] Tobias Stahl, Silvia L Pintea, and Jan C van Gemert. Divide and count: Generic object counting by image divisions. *Trans. Image Processing*, 28(2):1035–1044, 2019.

[42] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *AAAI Conf. on Artificial Intelligence workshops*, 2012.

[43] Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness NMS and bounded IoU loss. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2018.

[44] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *Int. J. Comput. Vision*, 104(2):154–171, 2013.

[45] Paul Viola and Michael J Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.

[46] Christian Wojek, Gyuri Dorkó, André Schulz, and Bernt Schiele. Sliding-windows for rapid object class localization: A parallel technique. In *Joint Pattern Recognition Symposium*. Springer, 2008.

[47] J. Xie, R. Girshick, and A. Farhad. Unsupervised deep embedding for clustering analysis. In *Int. Conf. Mach. Learning*, 2016.

[48] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. Towards K-means-friendly spaces: Simultaneous deep learning and clustering. In *Int. Conf. Mach. Learning*, 2017.

[49] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[50] Kai Zhang and James T Kwok. Simplifying mixture models through function approximation. In *Neural Inform. Process. Syst.*, 2007.

[51] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[52] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conf. Comput. Vision*. Springer, 2014.