Precise Detection in Densely Packed Scenes

Eran Goldman[1,2] Roei Herzig[4] Aviv Eisenschtat[1] Oria Ratzon[1] Itsik Levi[1]
Jacob Goldberger[2] Tal Hassner[3]

1. Trax Retail, 2. Bar-Ilan University , 3. Open University of Israel, 4. Tel Aviv University

CVPR LONG BEACH CALIFORNIA June 16-20, 2019

Our contributions

- ***New detection dataset with new challenges:*** Densely packed retail images
- ***Soft-IoU layer*** estimating overlaps between detections and objects
- ***EM-Merger unit*** resolving overlapping detections by hierarchical clustering

SKU-110K dataset

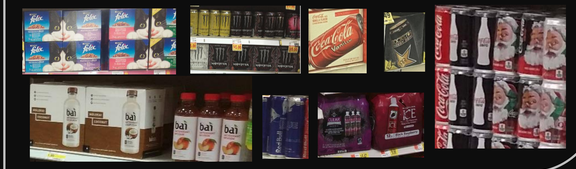
- 1,000,000+ objects
- 200+ of objects per image
- Huge variation in items and appearances
- Structured, Crowded, Fine-grained

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

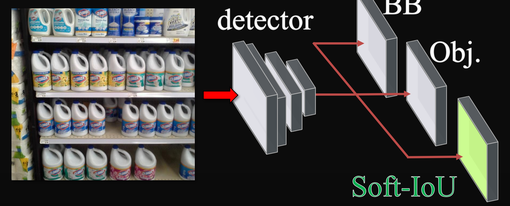
Key insights:

Appearance of detected patch is not always enough.

NMS-like heuristics frequently fail in packed scenes.



Novel IoU layer

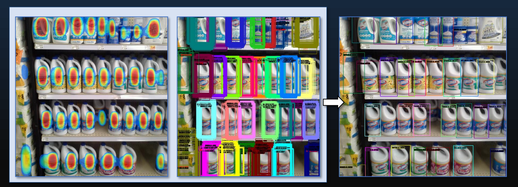


- Predict IoU rates between detected and true boxes
- Take a probabilistic interpretation of the IoU rate function and learn it with cross-entropy loss

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

EM-Merger unit

Resolve overlapping detections



- Model original detections as a GMM
- Use EM to reduce number of GMM components
- Convert the reduced GMM to final detections

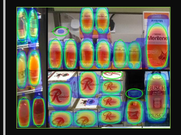
E-step: $\pi(t) = \min_{i \le 1 j \le k} KL(f_t || g_j)$

M-step: $\beta_j = \sum_{t \in \pi^{-1}(j)} \alpha_t$

$$\mu'_j = \frac{1}{\beta_j}\sum_{t \in \pi^{-1}(j)} \alpha_t \mu_t$$

$$\Sigma'_j = \frac{1}{\beta_j}\sum_{t \in \pi^{-1}(j)} \alpha_t \left(\Sigma_t + \left(\mu_t - \mu'_j\right)\left(\mu_t - \mu'_j\right)^T\right)$$

- ✓ Agglomerative initialization
- ✓ 2D matrix computations
- ✓ Fast convergence



Results

Qualitative results

Quantitative results

Method	AP	$AP^{0.75}$	AR^{300}	$P^{R=0.5}$	MAE	$RMSE$
Monkey	0	0	0.01	0	N/A	N/A
Faster-RCNN	0.045	0.01	0.066	0	107.46	113.42
YOLO9000	0.094	0.073	0.111	0	84.166	97.809
RetinaNet	0.455	0.389	0.53	0.544	16.584	30.702
Base & NMS	0.413	0.384	0.484	0.491	24.962	34.382
Soft-IoU & NMS	0.418	0.386	0.483	0.492	25.394	34.729
Base & EM-Merger	0.482	0.54	0.553	0.802	23.978	283.971
Our full approach	**0.492**	**0.556**	**0.554**	**0.834**	**14.522**	**23.992**

For code, data, more results and info, see our project page:

https://tinyurl.com/sku110k