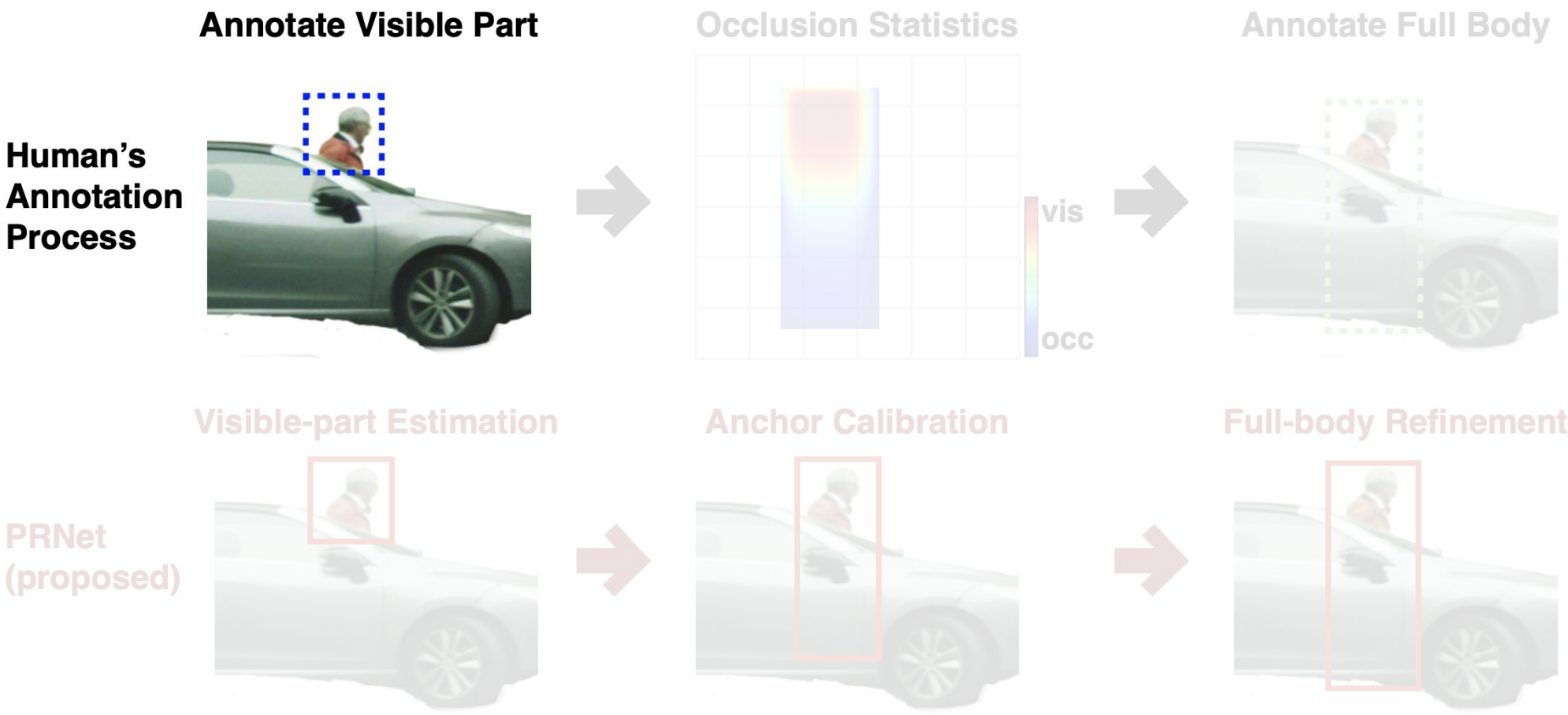
Progressive Refinement Network for Occluded Pedestrian Detection

Xiaolin Song[1*], Kaili Zhao[1*], Wen-Sheng Chu, Honggang Zhang[1], Jun Guo[1]

[1]Beijing University of Posts and Telecommunications, Beijing, China
[*]Equal Contribution

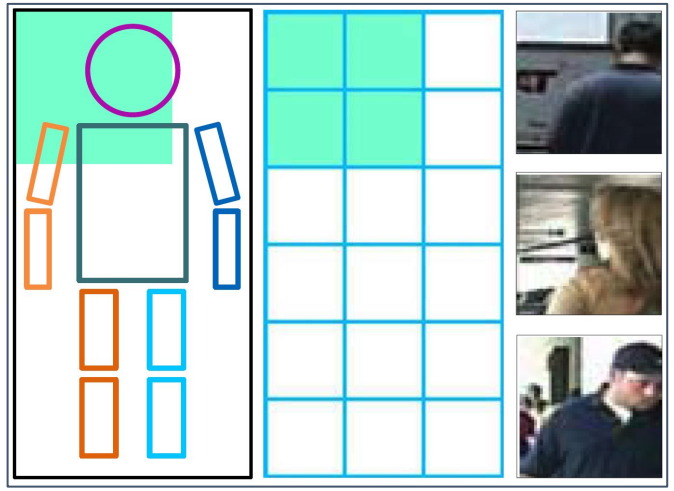
Motivation

Progressive Refinement Network (PRNet) sequentially refines anchors in three phases, as motivated by human's progressive process on annotating occluded pedestrians.



Occlusion-Aware Pedestrian Detection

Independent Detector

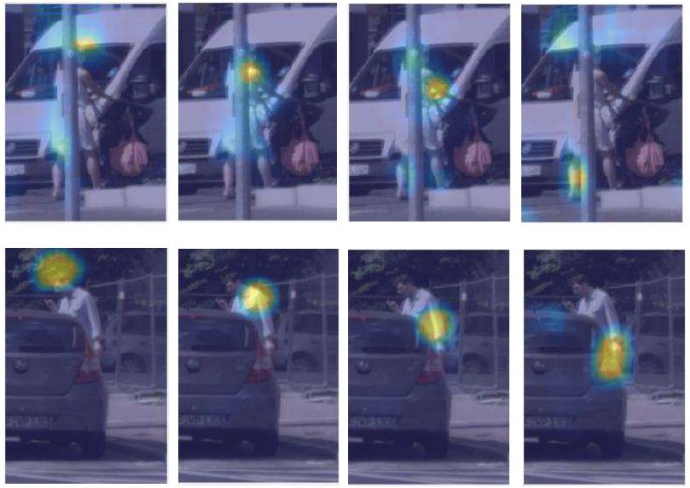


Shet et al (CVPR'07)
Wu et al (ICCV'05)
Ouyang et al (CVPR'12,13)
Tian et al (ICCV'15)

❌ Expensive computation

❌ Exhaustive enumeration

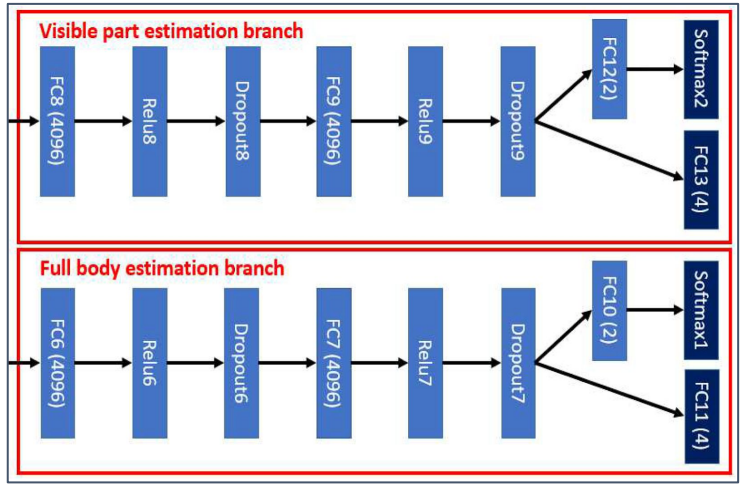
Attention-Based Methods



Zhang et al (CVPR'18)
Pang et al (ICCV'19)

❌ Slow inference

Auxiliary Visibility Classifier

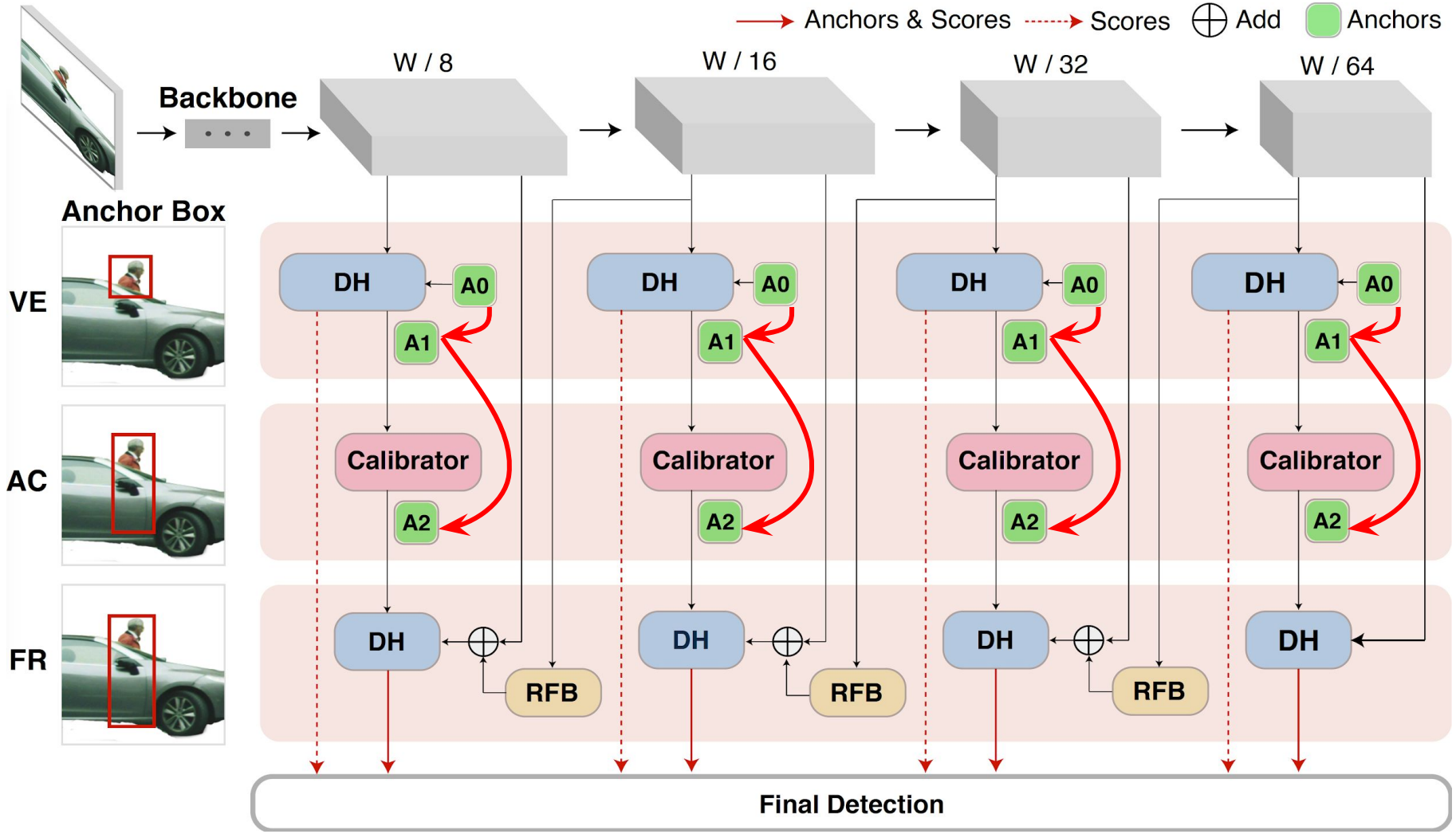


Noh et al (CVPR'18)
Zhou et al (ECCV'18)
Zhou et al (ICCV'19)

❌ No interaction

PRNet Architecture

With a ResNet50 backbone, PRNet embodies VE (Visible-part Estimation) and FR (Full-body Refinement) using two separate detection heads (DH), and introduce AC (Anchor Calibration) to bridge their gaps.



Phase 1: Visible-part Estimation (VE)

Goal: A0 —VE→ A1

VE offers high-confident prediction on visible parts, unlike most detectors trained with occlusions.

Reference VE Prediction

Model: $\mathcal{L}_{VE} = \underbrace{\mathcal{L}_{focal}}_{\text{Classification [1]}} + \lambda_v [y = 1] \underbrace{\mathcal{L}_{smoothL1}}_{\text{Regression [2]}}$

[1] Lin et al., “Focal loss for dense object detection,” CVPR 2017.
[2] Ren et al., “Faster R-CNN,” NIPS 2015

Phase 2: Anchor Calibration (AC)

Goal: A1 —AC→ A2

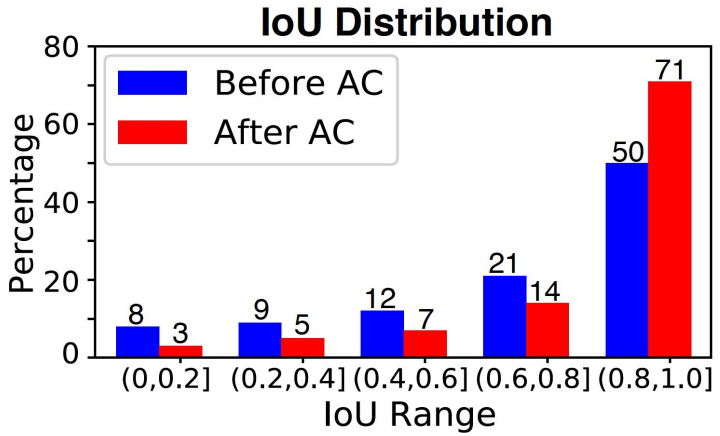
AC is necessary in bridging the gaps between VE and FR tasks (with different regression goals).

Before AC | After AC

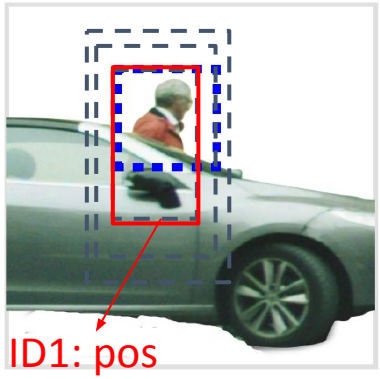
Prediction


1. Aspect-ratio gap

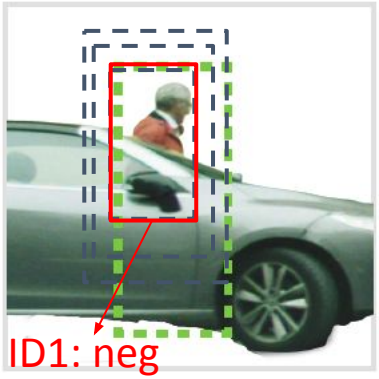
2. IoU gap



3. Anchor-assignment gap

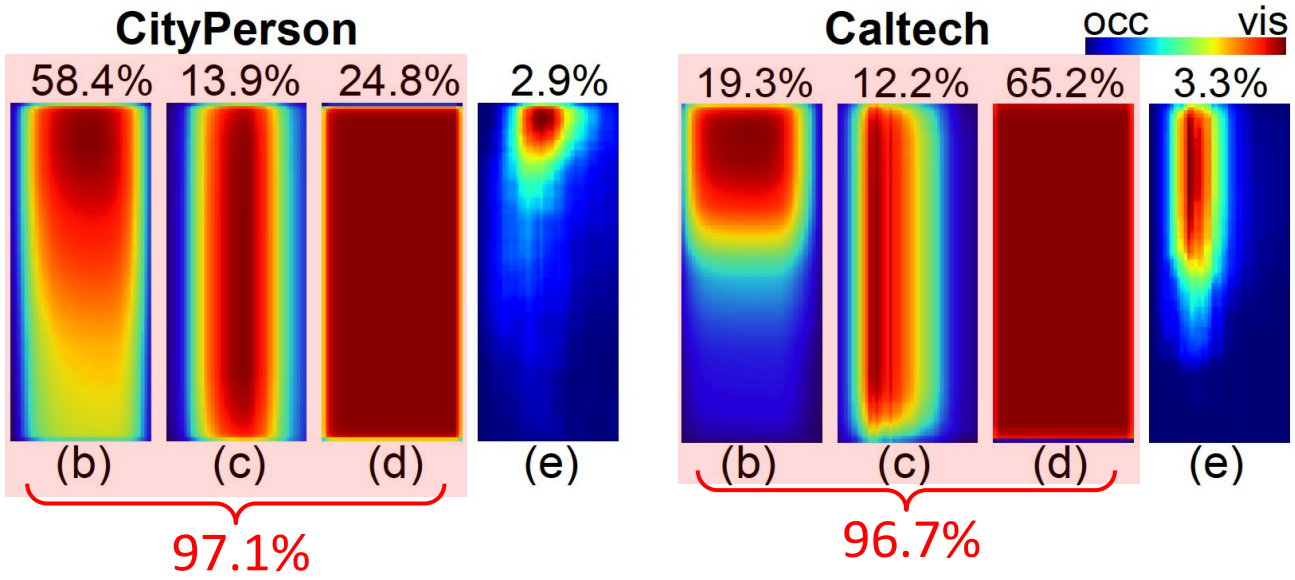


VE



FR

AC based on Occlusion Statistics

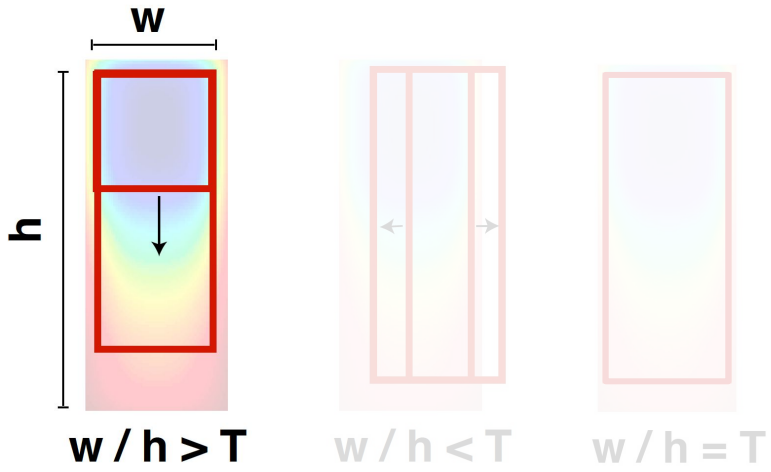


Percentage (%) denotes the likelihood of each occlusion pattern.

(b) Horizontal occlusions
(c) Vertical occlusions
(d) Non-occlusions
(e) Others

Key observation: Two calibrators can cover ~97% occlusion patterns according to occlusion statistics.

Calibrators:
(T=0.41)



Phase 3: Full-body Refinement (FR)

Goal: A2 —FR→ B^*

Different from VE, FR starts to deal with hard positive samples (eg, occlusion).

Model: $\mathcal{L}_{FR} = \underbrace{\mathcal{L}_{focal}}_{\text{Classification}} + \lambda_f [y = 1] \mathcal{L}_{occ}$

Occlusion Loss:

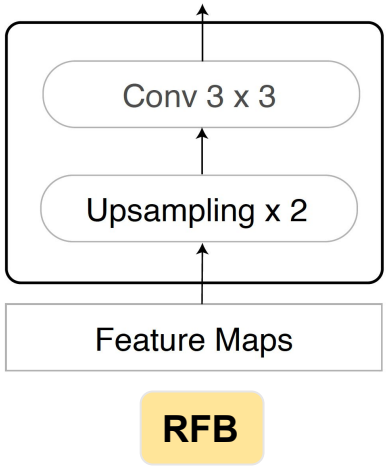
Anchor box from AC: $a \in \mathcal{A}_2$, final full-body box: $b \in \mathcal{B}_{gt}$

$$\mathcal{L}_{occ} = \sum_{a \in \mathcal{A}_2} \underbrace{(1 - \text{IoU}(a, b))}_{\text{Occlusion weight}} \underbrace{\{[|s| < 1]\, 0.5 s^2 + [|s| >= 1]\, (|s| - 0.5)\}}_{\text{Smooth L1 loss}}$$

Receptive Field Backfeed (RFB)

Challenge:

As in most detectors, shallow layer only fire on visible parts or small-size boxes due to limited receptive fields.



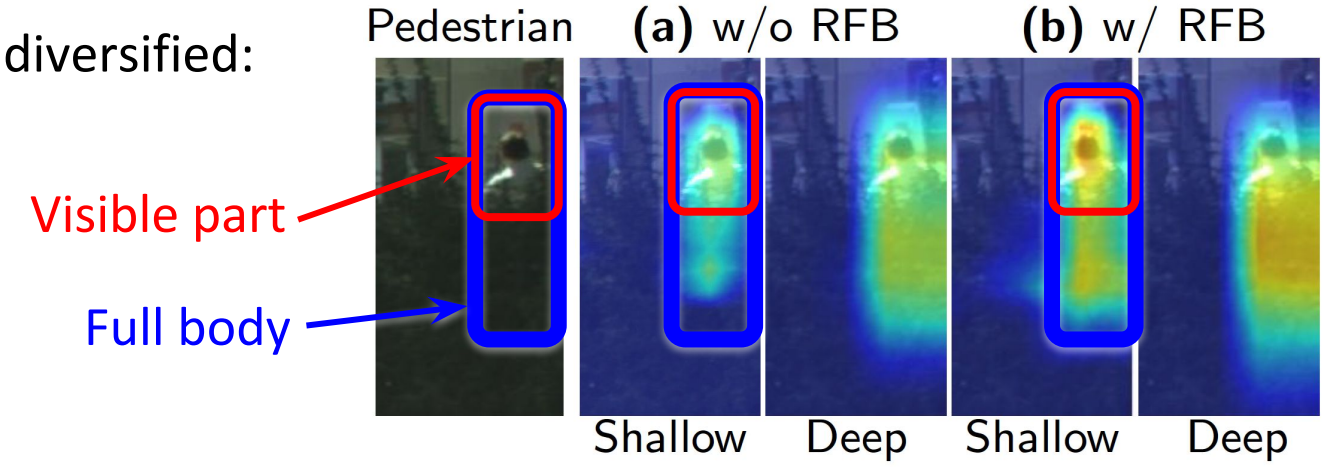
RFB Module:

Diversifies receptive fields by backfeeding deep features to previous layers.

RFB Effects:

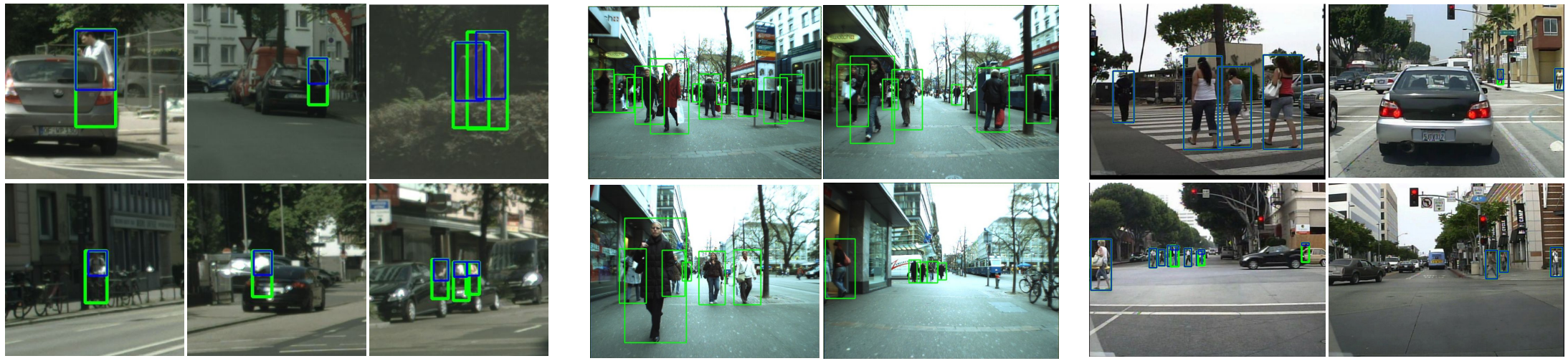
With RFB, receptive field in shallow layers are diversified:

- Visible part region is enhanced.
- Full body region is complemented.



Dataset and Settings

Datasets



Metrics:

MR^{-2} on 6 occlusion subsets with [different visibility ratios].

R: [0.65,1], HO: [0.2, 0.65], R+HO: [0.2, 1], Bare: [0.9, 1.0], Partial: [0.65, 0.9], Heavy: [0, 0.65].

Training: All models reported in this paper are trained on CityPersons (tr) dataset.

Inference: We obtain anchor boxes from FR, and anchor scores by multiplying the scores from VE and FR.

Ablation Study

Three-phase components:
PRNet's three-phase architecture showed a consistent improvement over single-phase and two-phase designs.

Architecture	VE	AC	FR	R	HO
PRNet-F			✓	15.6	45.7
PRNet-VA	✓	✓		11.7	51.3
PRNet-VAF	✓	✓	✓	11.4	45.3

PRNet-F (single phase): A standalone FR
PRNet-VA (2 phases): VE+AC
PRNet-VAF (3 phases): VE+AC+FR

Occlusion loss and RFB module:
PRNet couples occlusion loss and RFB, achieving the most improvement when both modules are included.

Architecture	+Occ.	+RFB	R	HO
PRNet-VAF			11.4	45.3
PRNet-VAF-OCC	✓		11.0	45.7
PRNet-VAF-RFB		✓	11.6	44.9
PRNet (ours)	✓	✓	10.8	42.0

Within-dataset: CityPersons (tr) → CityPersons (val)

Method	Occ.	Scale	R	HO	R+HO	Heavy	Partial	Bare	Time
Adapted FasterRCNN [39]		×1	15.4	64.8	41.45	55.0	18.9	9.3	-
TLL+MRF [32]		×1	14.4	-	-	52.0	15.9	9.2	-
CSP [19]		×1	**11.0**	-	-	[**49.3**]	**10.4**	7.3	0.33
FasterRCNN+ATT [40]	✓	×1	16.0	56.7	38.2	-	-	-	-
RepLoss [35]	✓	×1	13.2	-	-	56.9	16.8	7.6	-
		×1.3	11.6	-	-	55.3	14.8	7.0	-
OR-CNN [41]	✓	×1	12.8	-	-	55.7	15.3	[**6.7**]	-
		×1.3	11.0	-	-	51.3	13.7	5.9	-
MGAN [27]	✓	×1	11.3	[**42.0**]	-	-	-	-	-
FRCN+A+DT [42]	✓	×1.3	11.1	44.3	-	-	11.2	6.9	-
ALFNet [18]		×1	12.0	43.8	**26.3**	**51.9**	11.4	8.4	0.27
Bi-box [44]	✓	×1.3	11.2	44.2	-	-	-	-	-
PRNet (ours)	✓	×1	[**10.8**]	[**42.0**]	[**25.6**]	53.3	[**10.0**]	**6.8**	0.22

Accurate!

Efficient!

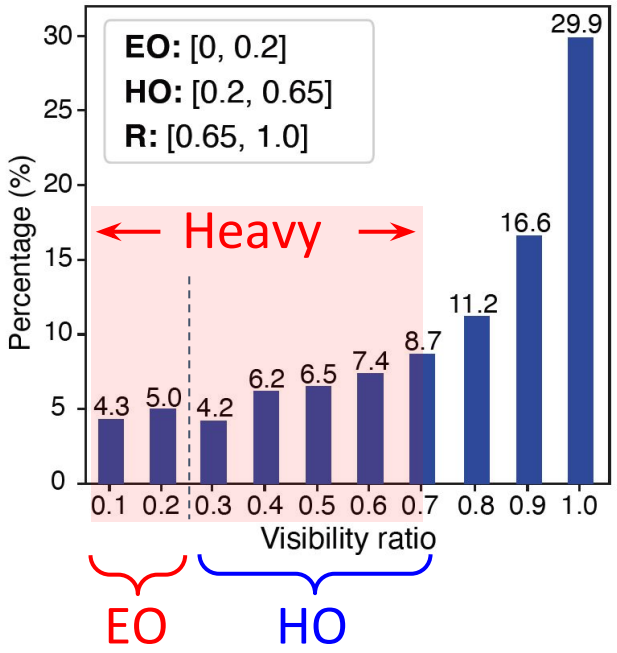
False Positives in alternative methods

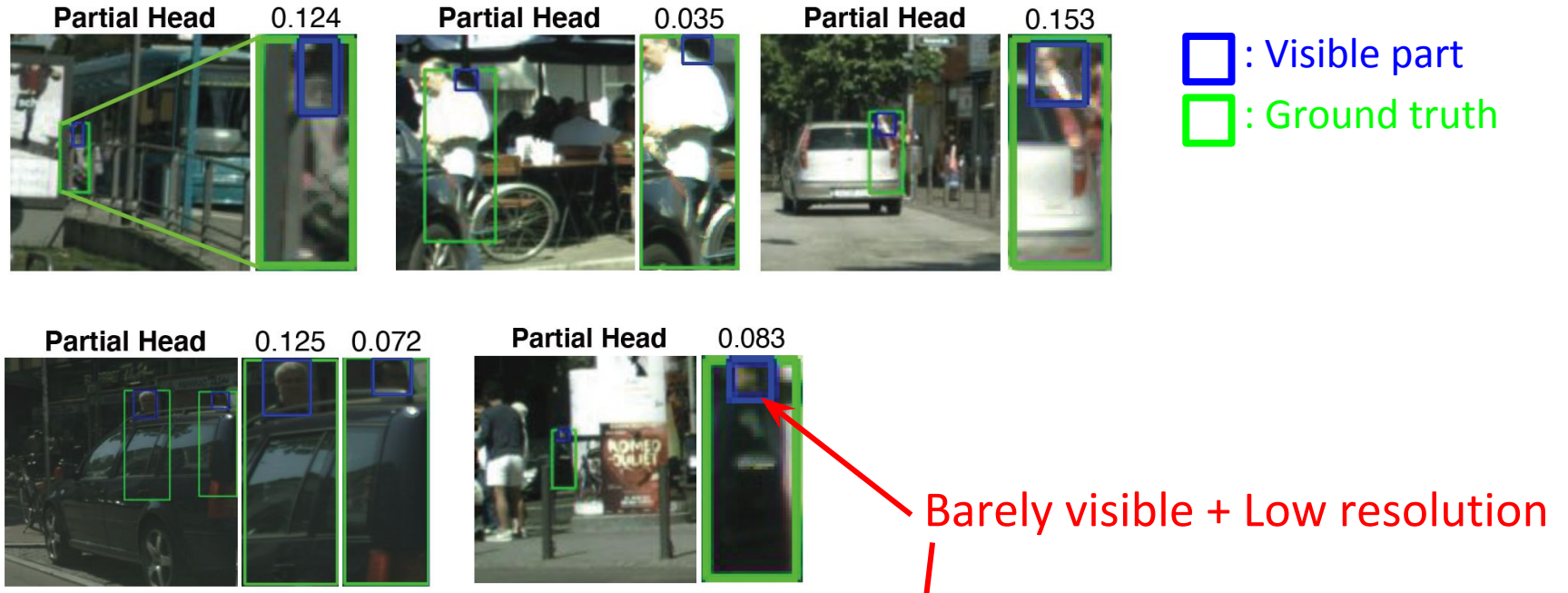
False Negatives in alternative methods

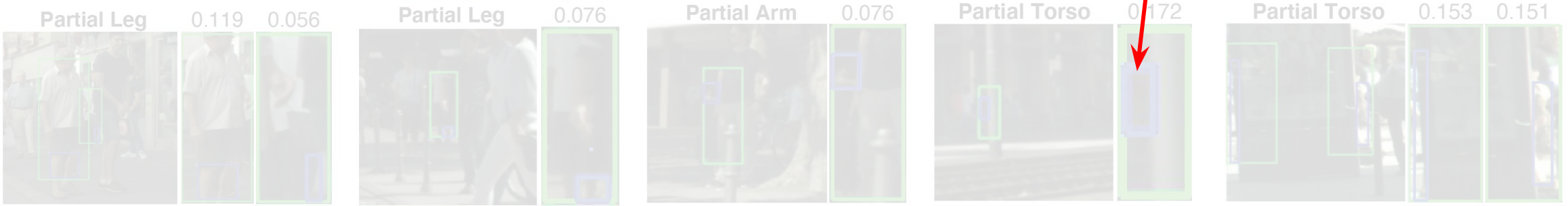
: detection : ground truth

Breakdowns in Heavy Subset

We breakdown heavy subset into HO and EO, i.e., Heavy = HO ∪ EO.







Cross-dataset Generalization

CityPersons → Caltech

Method	R (o)	R+HO (o)	R (n)	Time
ALFNet [18]	25.0	35.0	19.0	39.2
CSP [19]	**20.0**	[**27.8**]	**11.7**	61.3
PRNet (ours)	[**18.3**]	**28.4**	[**10.7**]	42.1

CityPersons → ETH

Method	R+HO	Time
FasterRCNN [39]	35.6	-
FasterRCNN+ATT [40]	33.8	-
CSP [19]	37.2	61.3
ALFNet [18]	**31.1**	39.2
PRNet (ours)	[**27.0**]	42.1

PRNet Prediction

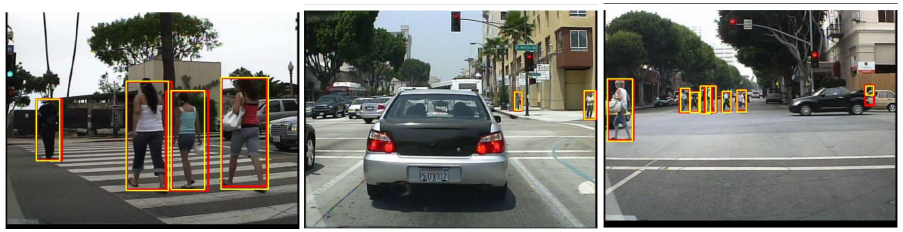
: Visible part

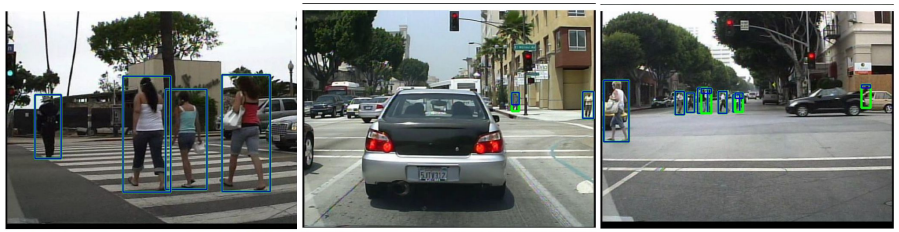
: Full body

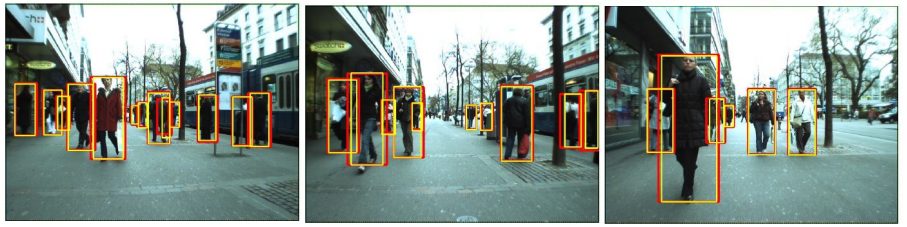
Ground Truth

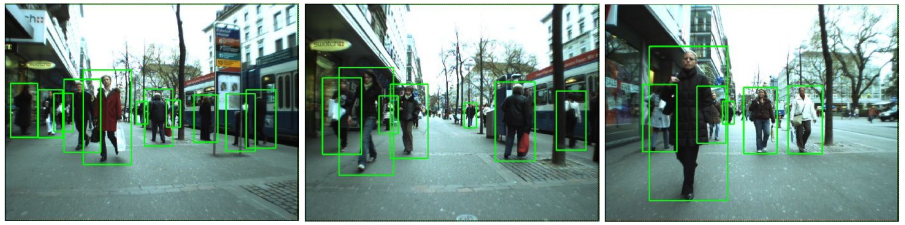
: Visible part

: Full body





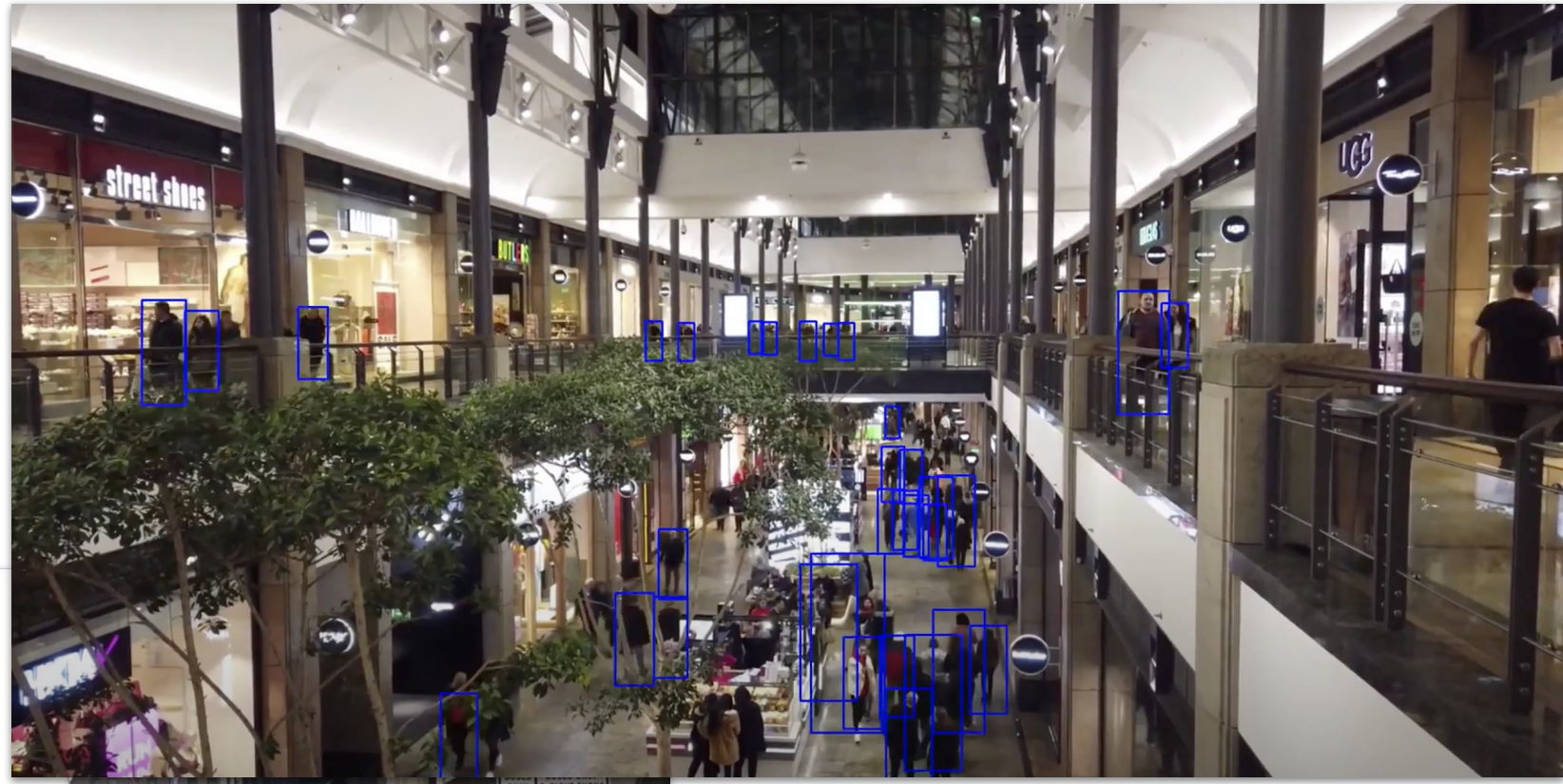




Rationale for PRNet's gain:

- Mimics every step in human's established principles, and thus fits the problem more naturally.
- Propagates detection from visible parts, and liberates occlusion patterns exhausted in full-body detectors.

PRNet Generalizes Well!



Code and model are available online!
https://github.com/sxlpris/PRNet