BEV-Net: Assessing Social Distancing Compliance by Joint People Localization and Geometric Reasoning

Supplemental Material

Zhirui Dai[1], Yuepeng Jiang[1], Yi Li[1], Bo Liu[2], Antoni B. Chan[3], and Nuno Vasconcelos[1]

[1]Department of Electrical and Computer Engineering, UC San Diego
[2]Wormpex AI Research
[3]Department of Computer Science, City University of Hong Kong

{zhdai,yuj009,yil898,boliu}@eng.ucsd.edu, abchan@cityu.edu.hk, nvasconcelos@ucsd.edu

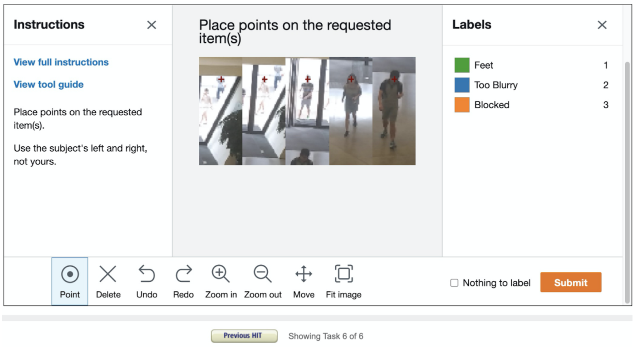
A. Dataset Annotation

Annotation procedure. The original CityUHK-X dataset [4] contained the head annotations of all people in the scene, as well as extrinsic camera parameters in the form of height h and pitch angle θ relative to ground plane. The intrinsic parameters were assumed available at training and test time. As the height of each individual is unknown, head locations are not sufficient to recover pedestrians' locations in the world coordinates. Therefore, we used Amazon Mechanical Turk to annotate feet locations of each person, with one-to-one correspondence to the head locations.

As the number of people in each scene varies greatly (minimum 1 to maximum 121), the scene images are preprocessed into rectangular crops around each head location. The size of rectangles are selected adaptively to make sure that each crop contains the whole person selected in the original image. Given each crop with marked head location, workers are required to locate the midpoint between both feet that correspond to the same person (figure 1); In crowded areas where one or both feet are occluded by objects or other pedestrians, workers are expected to provide their best estimate of feet location, or indicate that too little information is available to do so.

Each of the crops is assigned to three workers. The annotated coordinates from each worker are averaged after the exclusion of outliers. If at least two workers think they could see the feet clearly of the given person in the crop, then the crop is marked 'valid' (clearly visible). Otherwise, the feet of the given person are marked to be occluded.

Annotation outcome. 87,746 feet locations were annotated using the procedure described above. Among them, 63,669 (72.56%) were clearly visible and 24,077 (27.44%) occluded. Figure 2 shows the percentage of estimated annotations due to occluded body parts as functions of camera height and angle. The statistics reveal that occlusion occurs more frequently with low camera height and small pitch angles, making social distancing detection particularly challenging in these scenarios.



Instructions

Place points on the requested item(s)

Place points on the requested item(s).

Use the subject's left and right, not yours.

Labels

🟩	Feet	1
🟦	Too Blurry	2
🟧	Blocked	3

□ Nothing to label

Showing Task 6 of 6

Figure 1: Annotation interface on MTurk.

B. Homography Derivation

The setting of the camera is shown as figure 2 in the main text. The origin of world coordinate is set to be the camera's perpendicular projection on the ground plane, and the yaw angle of camera is set to be 0 by aligning it with the x-axis of world coordinates. We further assume that the camera has zero roll angle, *i.e.* its view is straightened to the horizon. This is a reasonable setting for most surveillance systems. Given the camera's height h and pitch angle θ, the transformation from the world frame to the optical frame,

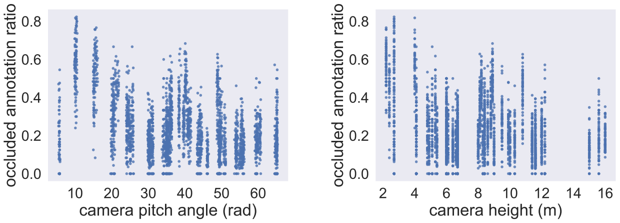


Figure 2: Percentage of estimated annotations from occluded body parts. More occlusion is found at smaller pitch angles and lower camera heights.

${}_O\mathbf{T}_W$, is given by

$$
\begin{aligned}
{}_O\mathbf{T}_W &= {}_O\mathbf{T}_C \; {}_W\mathbf{T}_C{}^{-1} \\
&= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & h \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} 0 & -1 & 0 & 0 \\ -\sin\theta & 0 & -\cos\theta & h\cos\theta \\ \cos\theta & 0 & -\sin\theta & h\sin\theta \\ 0 & 0 & 0 & 1 \end{bmatrix},
\end{aligned} \tag{1}
$$

where ${}_O\mathbf{T}_C$ is the transformation from the camera frame to the optical frame, and ${}_W\mathbf{T}_C$ is from the camera frame to the world frame.

In CityUHK-X-BEV dataset, the camera focal lengths (f_u, f_v) are given and for generality, we suppose there is no optical skew nor image center displacement. Hence, the intrinsic matrix is

$$
\mathbf{K} = \begin{bmatrix} f_u & 0 & u_c^I \\ 0 & f_v & v_c^I \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}
$$

Denoting with $\mathbf{P}$ the canonical projection matrix, transformation from point (x, y, z) in the world frame to coordinates (u, v) in the image frame is given by

$$
{}_I\mathbf{T}_W = \mathbf{K}\mathbf{P} \; {}_O\mathbf{T}_W \tag{3}
$$

$$
\begin{bmatrix} u & v & 1 \end{bmatrix}^\top = {}_I\mathbf{T}_W \begin{bmatrix} x & y & z & 1 \end{bmatrix}^\top. \tag{4}
$$

For a plane at $z = h_0$, we can easily get the projection of points in the plane by using the camera's relative height $h' = h - h_0$. So, $z = 0$ and equation 4 becomes

$$
\begin{bmatrix} u & v & 1 \end{bmatrix}^\top = {}_I\mathbf{H}_W \begin{bmatrix} x & y & 1 \end{bmatrix}^\top, \tag{5}
$$

where

$$
{}_I\mathbf{H}_W = \begin{bmatrix} u_c^I \alpha & -f_u & u_c^I h'\beta \\ v_c^I \alpha - f_v \beta & 0 & h'(f_v \alpha + v_c^I \beta) \\ \alpha & 0 & h'\beta \end{bmatrix}, \tag{6}
$$

$\alpha = \cos\theta$, $\beta = \sin\theta$, (f_u, f_v) are the horizontal and vertical focal length of the camera respectively, and (u_c^I, v_c^I) the image center.

Since the BEV map is under a certain scale as equation 2 in the main text, the transformation between BEV map coordinates and the world frame is

$$
{}_W\mathbf{T}_B = \begin{bmatrix} 0 & -s & x_c + sH/2 \\ -s & 0 & y_c + sW/2 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7}
$$

where H, W are the height and width of the BEV map, and (x_c, y_c) is the world coordinate of the image center on the ground plane. Matrices of equation 6 and equation 7 are combined in equation 5 of main text to build the transform from image frame to BEV.

C. Network Architecture

Figure 3 summarizes the architecture for each branch of BEV-Net. Image-view (IV) branches estimate head or feet locations from input image using an encoder-decoder structure. The IV encoders followed the same design as the first 4 convolutional blocks of VGG-16 [10] with batch normalization [3]. Head and feet feature maps are then processed by a fully-convolutional decoder network into the IV heatmap. Pose branch uses fully connected layers stacked on top of a ResNet-101 [2] feature extractor to regress camera height and pitch angle. The head and feet feature maps are projected into bird's eye view (BEV) using the BEV-Transform module (section 4.2 of main text), then fed into the BEV decoder which predicts the final BEV heatmap.

D. More Ablation Study

Performance on split scene setting. As shown in figure 4, camera poses varies even within the same scenes of the CityUHK-X-BEV dataset. In the paper, we use the setting of PoseNet [5], which trains and tests on the same scenes. We believe that this is the most suited for a public health setting, where there is usually some planing of the locations to monitor and data can be collected at those locations. In this setting, parameter variation is mostly due to camera motion (e.g. pan-zoom cameras), wind effects, etc. and usually less severe than even in figure 4. A more drastic generalization to completely unseen scenes is a much more challenging task. We also test BEV-Net with some scenes unseen during training. The chamfer distance increases to 2.41/80.33%, IoU of local risk drops to 54.86%, and the global risk MSE is 50.14×10^{-4}. We can see that BEV-Net still outperforms most baselines.

Encoder shared across branches. A BEV-Net with encoder shared across feet, head and pose branches has chamfer distance 1.25, local risk IoU 71.01%, and global risk

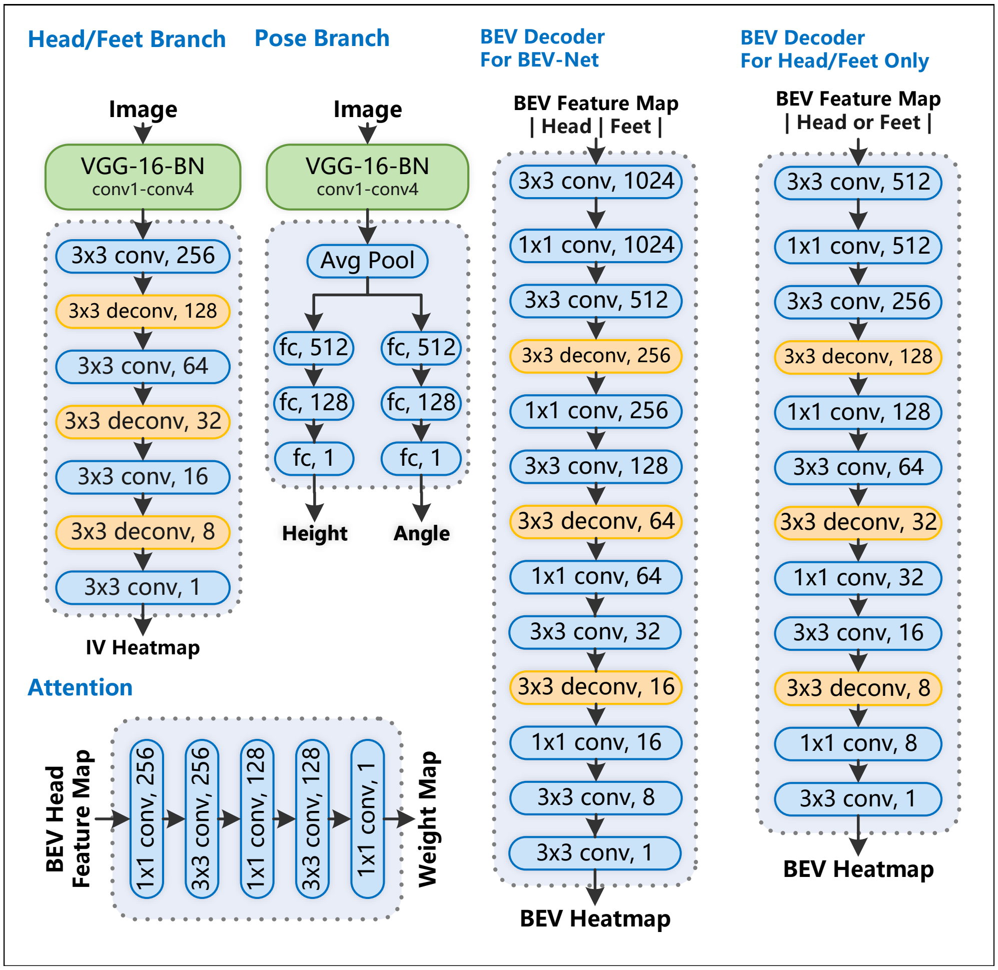
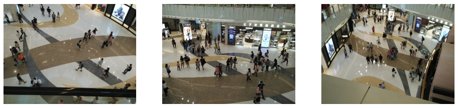


Figure 3: Network architectures. From left to right: IV (head/feet) branch, pose branch, BEV branch. The left bottom is the attention module. All `conv` layers have stride $s = 1$; `deconv` layers use stride $s = 2$. Nonlinearity, dropout [11] and batch normalization [3] omitted between some layers for simplicity.



$55.7°, 12.0$m $30.0°, 8.0$m $40.4°, 12.0$m

Figure 4: Variation in camera poses in the same scene of CityUHK-X-BEV.

MSE 5.88×10^{-4}. i.e. a little weaker than original implementation.

E. Qualitative Examples

Figure 5 and 6 contain qualitative comparison of localization and risk predictions from the proposed BEV-Net and baseline approaches using detection [1, 8] and crowd counting [6, 7] backbones. The results confirm the observations in main paper that detection methods have low recall for pedestrians far away, while counting methods fail to produce accurate localization in ground plane. In contrast, BEV-Net captures more people in crowded scenes, especially in areas far from the camera where occlusion is common, as well as those with extreme (close to 90°) camera angles. This advantage translates to better localization and risk estimate performance, both in the visualizations and in quantitative results (table 1 of main text).

References

[1] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3, 4, 5

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

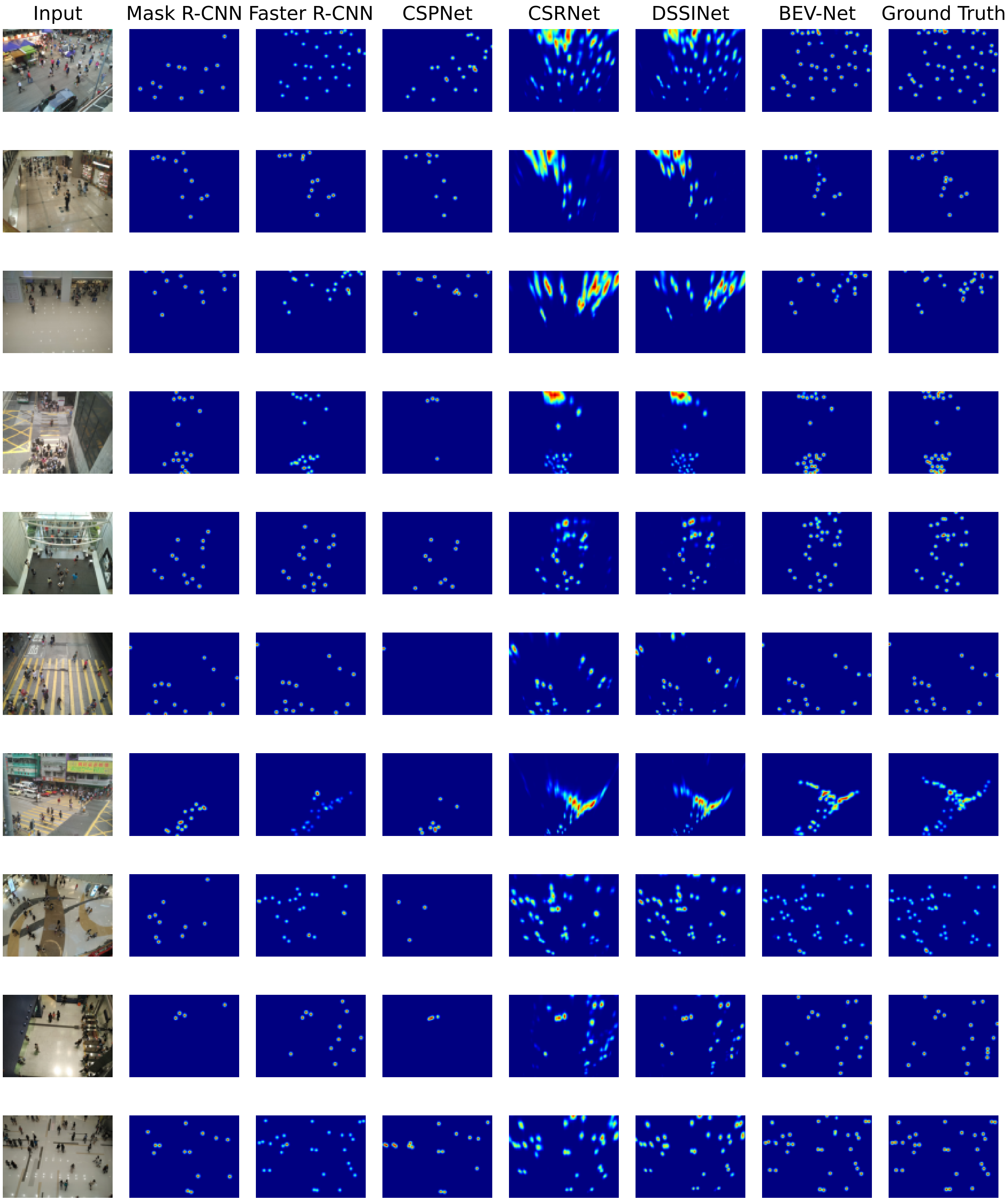


Figure 5: Qualitative comparison of **BEV heatmaps** between Mask R-CNN [1], Faster R-CNN [9], CSP [8], CSRNet [6], DSSINet [7] baselines and BEV-Net (ours). BEV-Net misses fewer people than detection methods [1, 8] and produces more accurate localization than crowd counting approaches [6, 7].

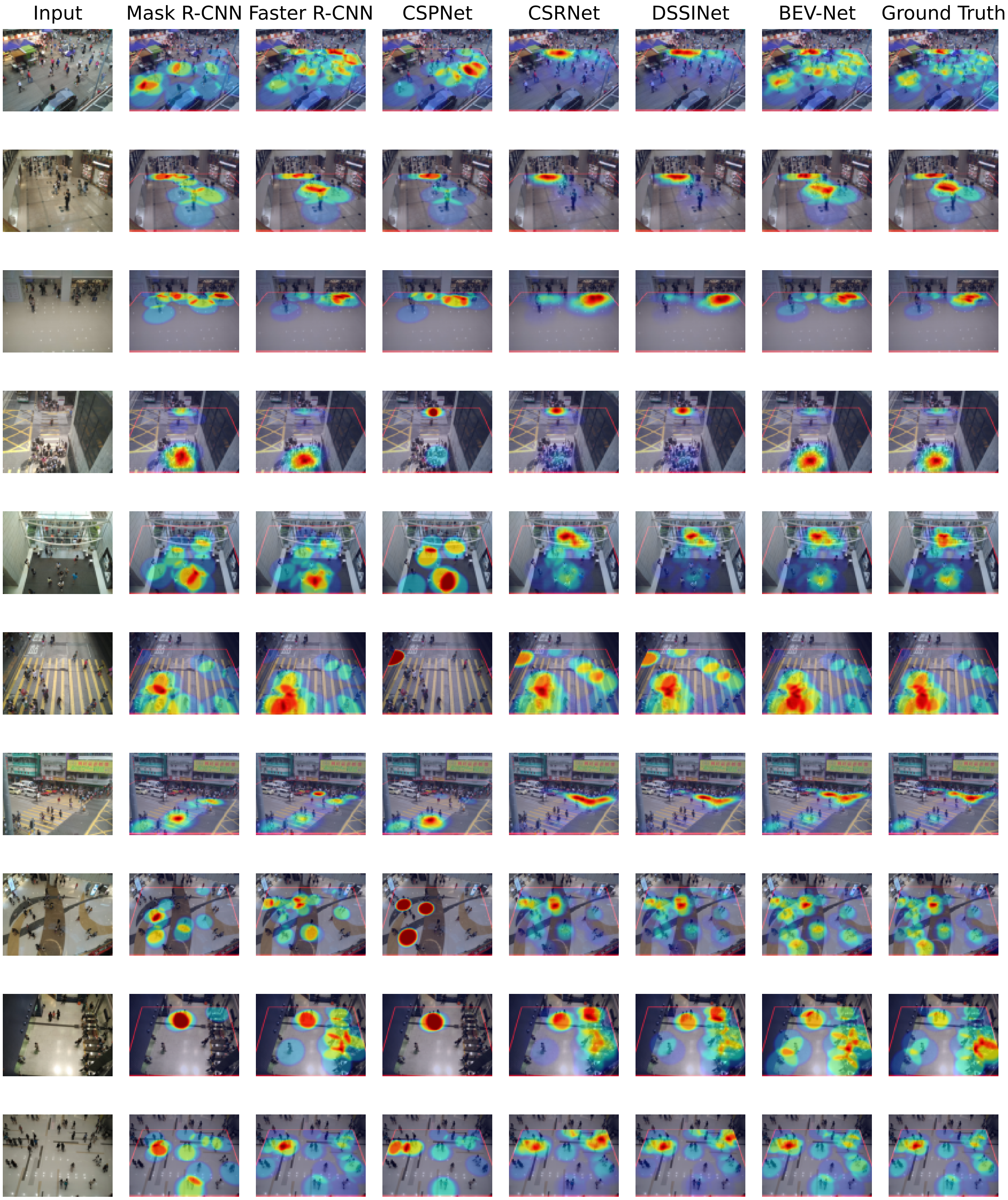


Figure 6: Qualitative comparison of **risk heatmaps** between Mask R-CNN [1], Faster R-CNN [9], CSP [8], CSRNet [6], DSSINet [7] baselines and BEV-Net (ours). Risk maps predicted by BEV-Net are closest to ground-truth.

ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 2

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2, 3

[4] Di Kang, Debarun Dhar, and Antoni Chan. Incorporating side information by adaptive convolution. In *Advances in Neural Information Processing Systems*, pages 3867–3877, 2017. 1

[5] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. 2015. 2

[6] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1091–1100, 2018. 3, 4, 5

[7] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. Crowd counting with deep structured scale integration network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1774–1783, 2019. 3, 4, 5

[8] Wei Liu, Shengcai Liao, Weiqiang Ren, Weidong Hu, and Yinan Yu. High-level semantic feature detection: A new perspective for pedestrian detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 4, 5

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 4, 5

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 3