Supervised People Counting Using An Overhead Fisheye Camera

Shengye Li, M. Ozan Tezcan, Prakash Ishwar, Janusz Konrad*
[shengye, mtezcan, pi, jkonrad]@bu.edu
Boston University
Department of Electrical and Computer Engineering

Abstract

We propose two supervised methods for people counting using an overhead fisheye camera. As opposed to standard cameras, fisheye cameras offer a large field of view and, when mounted overhead, reduce occlusions. However, methods developed for standard cameras perform poorly on fisheye images since they do not account for the radial image geometry. Furthermore, no large-scale fisheye-image datasets with radially-aligned bounding box annotations are available for training. We adapt YOLOv3 trained on standard images for people counting in fisheye images. In one method, YOLOv3 is applied to 24 rotated, overlapping windows and the results are post-processed to produce a people count. In another method, YOLOv3 is applied to windows of interest extracted by background subtraction. For evaluation, we collected and annotated an indoor fisheye-image dataset that we make public. Experiments on this dataset show that our methods reduce the people counting MAE of two natural benchmarks by over 60%.

1. Introduction

Indoor people counting (IPC) is an important task in applications such as smart HVAC control, space utilization and management, crowd statistics, building safety and emergency evacuation, and surveillance. At a high level, IPC methods are either passive or active in character. Active methods require occupants to always carry an electronic device (a beacon), which communicates with a data-collection unit. This approach, however, is intrusive, inconvenient, unreliable (occupants may lose the device) and costly to implement. Passive methods typically use cameras, CO_2 sensors (as a proxy for people count) or RF signal monitoring. In this paper, we focus on developing and validating passive IPC methods using fisheye cameras (also known as panoramic or 360° cameras) mounted overhead. This approach is very advantageous for IPC since a single camera can cover a large field of view with minimal occlusions. However, images from overhead-mounted fisheye cameras may contain human bodies at various orientations, such as upright, upside-down, horizontal or diagonal. Unfortunately, most of the existing people-detection and counting algorithms are designed for standard camera images where people appear upright. Another difficulty are severe geometric distortions at the periphery of a fisheye image.

In this paper, we leverage a state-of-the-art people detection algorithm designed for upright-oriented people to develop IPC algorithms for fisheye camera images. While many people detection algorithms have been designed for standard side-mounted cameras, the best performance to date has been achieved by deep-learning object detection algorithms. In particular, YOLO [14] achieves a very competitive performance in real time (using a desktop GPU). However, YOLO is trained on standard images where people usually appear upright. Moreover, there are no large-scale fisheye-image datasets with bounding boxes aligned with radially-oriented bodies for training. Consequently, application of YOLO to fisheye images is not straightforward. Here we design two methods to leverage YOLO for IPC in fisheye images. In one approach, we apply YOLO only to a window extracted from the upper central part of a fisheye image where the orientation of people should be close to upright. To cover the whole image, we create 24 rotations of the image and apply YOLO to the same window after each rotation. Then, we rotate the results back to the original angles and apply post-processing to prune multiple detections of the same person (the results from neighboring rotations may overlap). In an alternative approach, we first identify regions of interest (ROIs), where activity takes place, then we rotate each ROI to the upper central part of the image and apply YOLO. To identify areas of activity we apply our own variant of classical background subtraction.

In order to evaluate our methods, we collected a dataset of indoor images captured by overhead fisheye cameras where bounding boxes are aligned with human bodies (i.e., radially), unlike in other datasets where they are aligned with image axes. Experimental results on this dataset

*This work was supported in part by ARPA-E under agreement DE-AR0000944 and by the donation of Titan GPUs from NVIDIA Corp.

demonstrate that the proposed approaches reduce the people counting MAE by over 60% compared to two benchmarks.

The main contributions of our paper are:

- The development of two new people-counting algorithms for overhead-mounted fisheye cameras[1] that leverage a supervised object detection trained on standard side-mounted cameras without the need for any additional training.
- The collection of a new people-detection dataset composed of indoor videos captured by overhead-mounted fisheye cameras with body-aligned annotation bounding boxes. We believe this is the first such dataset; bounding boxes in existing datasets are aligned with image axes regardless of human body orientation.

2. Related Work

People detection using side-mounted, standard-lens cameras: Pedestrian detection using HOG features with Support Vector Machine classifier [4] is a well-known method in which an image is divided into square blocks, and a feature vector representing the weighted histogram of gradient orientations is computed for each block to train a linear SVM classifier. Dollár *et al.* proposed a fast pedestrian detector based on aggregate channel features (ACF) [7]. It uses HOG information and LUV luminance/color information to construct a feature pyramid to train an AdaBoost classifier. Among classical methods for people detection, ACF is considered to be state of the art.

Recently, object detection algorithms based on deep convolutional neural networks have demonstrated outstanding performance. While R-CNN [10] and Faster R-CNN [9] are the current top-performing detection methods, YOLO [14] and its newer version YOLOv3 [15] achieve very competitive performance in real time. YOLO has inspired our work on people counting from overhead fisheye cameras.

People detection using overhead, fisheye-lens cameras: Compared to side-mounted, standard-lens cameras, people detection using overhead, fisheye cameras has been studied much less, with most papers appearing in the last few years.

In a very recent method [20], Wang *et al.* proposed to model people as upright cylinders and derived a series of elliptic detection masks whose size diminishes with the distance from the image center. They applied four SVM classifiers to features derived from each detection mask: HOG and LBP features from full-size and half-size masks. The final result is a linear combination of scores from two pairs of SVMs for HOG and LBP features. In another method [3], Chiang and Wang proposed to rotate a fisheye image in small increments, and extract windows of various sizes from a narrow but tall region in the upper-center of the image. Then, they computed HOG features from each detection window and applied an SVM classifier to detect a person. This method inspired us to adopt a similar strategy of extracting rotated windows. Saito *et al.* proposed a Bayesian MAP approach for people detection [16]. They used principle component analysis and kernel ridge regression to first build a template model for the silhouettes of an upright human body and the top-views of the head and shoulders. Then, they used the model to compute MAP estimates. In contrast to training new classifiers on top-view fisheye images, Krams and Kiryati proposed a method that uses ACF trained on side-view, standard-lens images [11]. A novel aspect of their work is that they do not dewarp a fisheye image into a panoramic image. Instead, they dewarp features extracted from the fisheye image.

Recently, CNN-based people detection methods have been proposed for overhead, fisheye images. Nguyen *et al.* adapted YOLO using only the grayscale channel, but augmented it with background subtraction results obtained using an adaptive Gaussian mixture model [18] for people detection. They also simplified the network structure to reach speeds needed for real-time people detection on embedded devices. In [19], Tamura *et al.* applied rotation-invariant training, a data augmentation method to train YOLO on rotated perspective images. They used randomly-rotated standard images to simulate various poses and orientaions of people in fisheye images. They also proposed a clustering-based method to refine bounding boxes (detection results) as an alternative to non-maximum suppression. Another YOLO-based people detection method extracts highly-overlapping windows in order to avoid misses, and dewarps those windows by means of an omnidirectional-to-perspective image mapping [17]. The dewarped windows are then fed into standard YOLO trained on perspective images. The authors also proposed several variants of non-maximum suppression as a post-processing step. In contrast to using networks designed for perspective images, Nosaka *et al.* proposed a novel neural network architecture specialized for bounding box regression [13]. They used orientation-aware convolutional layers [21] followed by orientation-aware regression.

In summary, among methods that use YOLO, some use data augmentation to fine-tune YOLO while others use YOLO as is but combine it with dewarping and post-processing. Our proposed methods require neither dewarping nor data-augmented fine-tuning. Both our proposed methods combine clever geometric pre- and post-processing with an important YOLO-based re-verification step to realize significant performance gains.

3. Activity-Blind Application of YOLOv3

Our first approach leverages the observation that the appearance of people in the top-center region of a fisheye im-

[1] Source code available at: vip.bu.edu/projects/vsns/cossy/fisheye

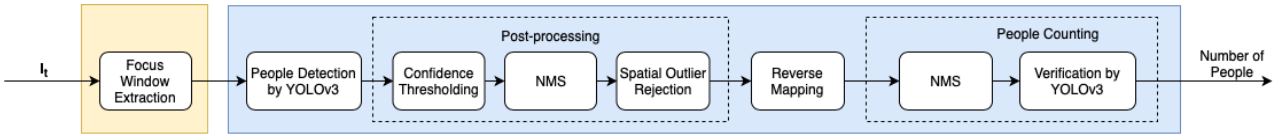
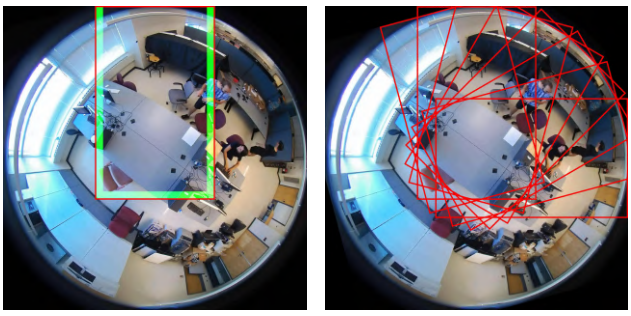


Figure 1: Block diagram of the proposed activity-blind (AB) people counting method.

age is similar to that in standard, side-mounted cameras. In this approach, we first extract a rectangular window, which we shall call a *focus window*, at the top-center of an overhead fisheye image. In the next step, we rotate the image by a small angle, and extract the same window with new data. The rotation and window extraction steps are repeated until focus windows are extracted from all parts of the image. Then, we use YOLOv3 as a person detector on each of the focus windows that we extracted, and perform a series of post-processing steps to generate reliable people detection results within each focus window. Subsequently, all detections are mapped from each focus window onto the complete fisheye image. Finally, multiple detections from neighboring focus windows are merged and verified to produce the final people count. Since this approach does not utilize activity information, we call it the *activity-blind* (AB) method. Its block diagram is shown in Fig. 1 and algorithm's details are provided next.

3.1. Focus window and image rotation

Since YOLOv3 is designed for full-size images, we can use a focus window that is large enough to capture human bodies in an upright or almost upright position in the upper half of the image. We use a focus window whose height and width equal 65% and 40% of the height and width of the full image, respectively, and the window's top is aligned with the upper boundary of the image. We applied image rotation in 15° increments as a trade-off between the overall complexity and precision. Fig. 2 illustrates the selected window size and rotation angle.



(a) (b)

Figure 2: (a) Placement of the 1,300×800-pixel focus window. The faint green area is the margin area defined in Section 3.2. (b) Focus window after reverse rotations.

3.2. People detection and post-processing

The YOLOv3 detector that we apply to every focus window is a *Fully Convolutional Network* (FCN) trained on the COCO dataset with 80 object classes. Since our focus is on people counting, we only retain those bounding boxes produced by YOLOv3 for which the confidence of the "people class" is high. Specifically, we only retain detections with an "objectness" score above a threshold of 0.3. Then, out of the retained object detections only those whose person-class score is the highest are kept. YOLOv3 may detect a person with several bounding boxes that significantly overlap each other within a single focus window. In order to avoid over-counting people, only one representative box among the overlapping detections should be retained. To this end, we apply Non-Maximum Suppression (NMS) to the detections [2] using an Intersection Over Union (IOU) threshold of 0.4. Finally, people who are situated very close to the left, right, or bottom boundaries of the focus window may not be fully visible and might create bounding boxes that do not cover the whole body of the person. This could deteriorate the performance of people counting when we merge results across different focus windows as described in Section 3.4. Therefore, we apply Spatial Outlier Rejection (SOR) to remove bounding boxes that intersect a Δ-wide margin inside the focus window along the right, left, and bottom boundaries of each focus window (Fig. 2(a)). We set Δ to 6.25% of the focus window width.

3.3. Reverse mapping of detections

People detection results (bounding boxes) need to be mapped from the relative position within each extracted focus window to the absolute position in the full fisheye image. A naïve approach is to rotate the bounding box by reverse angle used in image rotation. However, the bounding boxes generated by YOLOv3 in any focus window are aligned to the focus window axes and need not be radially oriented with respect to the center of the fisheye image. We therefore reverse-rotate only the center of a bounding box and then form new box of the same size as the original with a radial alignment. This process is illustrated in Fig. 3.

3.4. NMS, verification, and people counting

Despite the bounding box suppression within each focus window during post-processing, typically there will be multiple overlapping bounding boxes after reverse mapping. This is because neighboring focus windows have a large

(a) (b)

Figure 3: Comparison of the two reverse mapping approaches: (a) naïve approach, and (b) improved approach.

overlap with each other and a person can be detected within several focus windows without intersecting their boundaries. In order to assure accurate people counts, duplicate person detections need to be eliminated so that one person is associated with only one bounding box. We therefore implement NMS with an IOU threshold of 0.4 on the detections after reverse mapping.

In order to reduce false positives resulting fom erroneous detections by YOLOv3 that have not been eliminated by the preceding steps, we implement a final person detection verification step. Around each remaining bounding box, we extract a rectangular window that encompasses the box with a 30 pixel border. This new window is first rotated to the upright position as described in Section 3.3, but in the opposite direction. To account for potential angular misalignment we apply additional rotations by $\pm 15^{\circ}$ to extract three windows that are passed to YOLOv3. The detection results then undergo the confidence thresholding and NMS post processing steps as detailed in Section 3.2. If no less than 2 results confirm this is a person, then the original bounding box is accepted; otherwise it is rejected. The final people count is the number of bounding boxes that remain after this step.

4. Activity-Aware Application of YOLOv3

While the activity-blind approach achieves good performance in most cases (see Section 5.4), it is computationally intense since all the steps described in Section 3 need to be applied to each of the 24 focus windows, even if there is no person present. Therefore, we develop an *activity-aware method* (AA) to reduce the computational complexity. The main idea is to identify regions of interest (ROI) where people are likely to be present, and apply a people detector to only windows containing these regions. We extract ROIs *via* background subtraction, focus window selection and background model update steps. The block diagram of our AA algorithm is shown in Fig. 4 where the steps in the right blue box are exactly the same as those in our AB algorithm.

4.1. Background subtraction

Let $\boldsymbol{I}_t(x,y)=[I_t^R, I_t^G, I_t^B]$ and $\boldsymbol{B}_t(x,y)=[B_t^R, B_t^G, B_t^B]$ denote the RGB color values of the observed image and a reference background image, respectively, at time t and pixel spatial coordinates (x,y). Section 4.2 will describe how the reference background $\boldsymbol{B}_t$ is obtained. In the first step, the following thresholding is applied to produce an initial mask of changes $S_t(x,y)$:

$$S_t(x,y) = \begin{cases} 1, & \text{if } \sum_{A\in\{R,G,B\}} |I_t^A(x,y) - B_{t-1}^A(x,y)| > \theta \\ 0, & \text{otherwise} \end{cases}$$

where θ is a threshold. An example of a reference background and an input frame is shown in Fig. 5(a–b). Subsequently, two morphological operations are applied to the mask $S_t(x,y)$. First, opening with a 3×3 rectangular structuring element is applied to remove tiny patches that are likely to arise due to noise. Then, dilation operation with a 25×25 elliptical structuring element is applied to expand the remaining areas of the detected changes. A connected-component analysis is then performed and small-area components (having fewer than 3,600 pixels) are removed. This leads to the final ROI mask. Figure 5 (d) shows an example.

4.2. Background model

A variety of background models have been studied in the literature. A simple static background model is usually a fixed "empty" frame which cannot reflect changes in the background, such as those due to illumination variations, and may lead to unnecessarily large ROIs. Dynamic background models utilize recent frames to update model parameters, but they may produce false negatives if moving objects become nearly stationary for longer than the time period at which model parameters get updated. We use the following simple dynamic background model which leverages people detection results from previous time instants:

$$\boldsymbol{B}_t(x,y) = \gamma_t(x,y)\cdot\boldsymbol{B}_{t-1}(x,y) + (1-\gamma_t(x,y))\cdot\boldsymbol{I}_t(x,y),$$

where $\gamma_t(x,y)$ equals 1 if (x,y) belongs to a bounding box associated with a detected person and is zero otherwise. We note that at time t, people detection is performed first (so that indicator γ_t can be computed) and then the background is updated. Our background update mechanism uses the indicator γ_t to decide at each location (x,y) whether to use the current image value at time t as the new background (pixel belongs to the background) or to use the background value from previous time (pixel belongs to a bounding box). This reduces background contamination which affects many dynamic background models. Since background locations outside of a person's bounding box get immediately updated by the current image value, the model is robust to illumination changes that are challenging for static background models. The proposed update mechanism

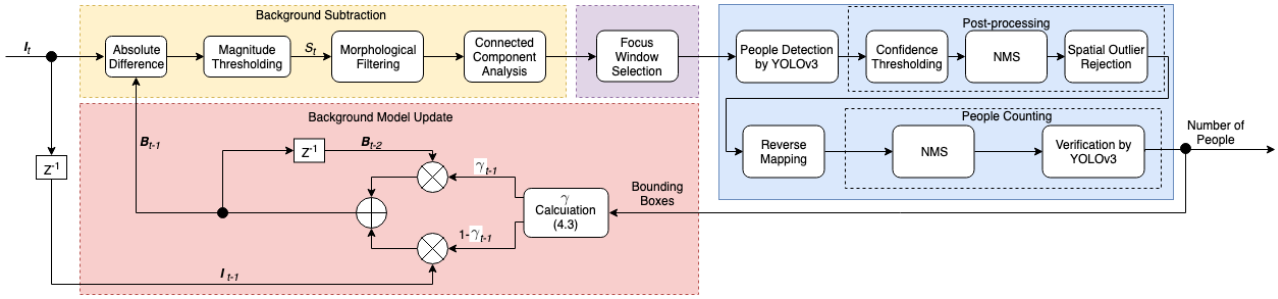


Figure 4: Block diagram of the proposed AA algorithm. Blocks within the blue-shaded rectangle are common to both AA and AB methods, while the other parts are unique to AA algorithm.

therefore offers benefits of both static and dynamic background models. We set the initial background to zero, i.e., $\boldsymbol{B}_0(x, y) = 0,\ \forall(x, y)$.

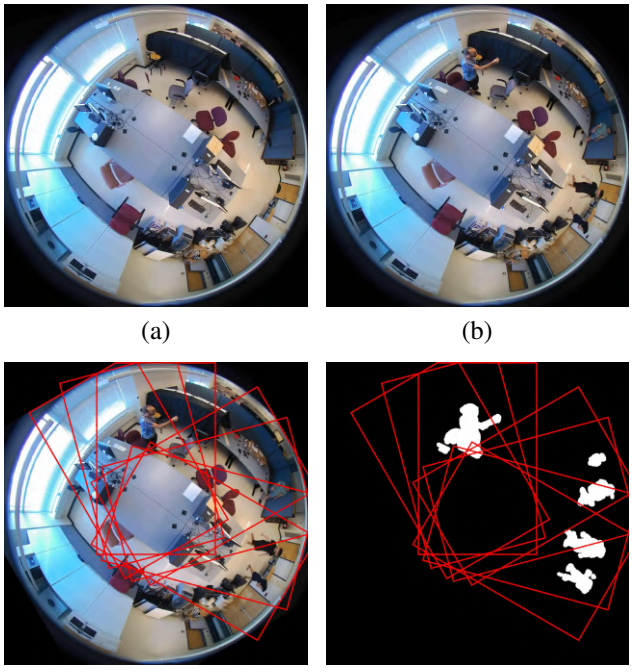
4.3. Focus window selection

We use a subset of the 24 rotated windows used in the AB method, and apply the same methodology to each window selected. In order to ensure full coverage of a connected component by a rotated window of width W and height H, the centroid C of a connected component is calculated first. Let O denote the center of camera's FOV and $R_i, i = 1, ...24$ the center of each of the 24 rotated windows. Window number $k = \arg\min_i \angle(\vec{OC}, \vec{OR}_i)$ is selected as the focus window for this connected component. If the connected component exceeds the left boundary of the central part of the focus window (a rectangle of width $W_c < W$ and height H), the neighboring window in counterclockwise direction is added to the final selection. Then, the same check is performed for the newly-added window. The process is repeated until the connected component does not exceed the left boundary of the central part of the last newly-added window. A similar procedure is applied to the right boundary and neighboring windows in the clockwise direction. An example illustrating focus window selection is shown in Fig. 5(d). The above steps are repeated for all remaining connected components. Then, people detection by YOLOv3, post-processing, reverse mapping and people counting (as detailed in Section 3) are applied to all focus windows selected earlier.

5. Experimental Results

5.1. Dataset

In order to evaluate the performance of our algorithms, we need a dataset of overhead fisheye images with bounding boxes for each person aligned with that person's orientation. Although there exist public people-detection datasets for fisheye images, they are annotated either by point location of a person's head [5] or by a bounding box aligned with image boundaries [6, 1]. However, our algorithms aim to produce radially-aligned bounding boxes since in an



(a) (b) (c) (d)

Figure 5: Examples of: (a) reference background; (b) current frame; (c) focus windows for the current frame overlaid on the RGB image; and (d) overlaid on the connected components of the final ROI.

overhead view people standing in a room appear as radially aligned in a fisheye image. In order to address this limitation, we collected and annotated a new dataset[2] called Human-Aligned Bounding Boxes from Overhead Fisheye cameras (HABBOF) composed of 4 videos (see Table 1 for details). The spatial resolution of all videos is 2,048×2,048 pixels. "Meeting1" and "Meeting2" videos were recorded by an AXIS M3057-PLVE camera whereas "Lab1" and "Lab2" videos – by a Geovision GV-FER12203 camera. Videos in the dataset capture some challenging scenarios, such as spatial and temporal illumination variations, occlusions, and motion in the center and at the periphery of the fisheye field of view. Example frames from all videos to-

[2] vip.bu.edu/habbof

Table 1: Properties of videos in the HABBOF dataset. "# of people" shows the maximum number of people in a single frame.

Video	Scenario	# of people	# of frames	FPS	Challenges
Meeting1	Conference room	3	1,200	30	Walking activity at image periphery
Meeting2	Conference room	3	1,200	30	Walking activity in image center and at periphery
Lab1	Computer lab	4	1,800	30	Strong occlusions Complex poses and walking at image periphery Spatially-nonuniform illumination
Lab2	Computer lab	4	1,800	12	Close proximity between people Time-varying global illumination Spatially-nonuniform illumination

gether with human-aligned bounding box annotations are shown in the first row of Fig. 6.

5.2. Baseline Algorithms

Although some recent methods leverage YOLO for people detection in overhead fisheye images [17, 19], we cannot compare the performance of our methods against them since neither dataset annotations nor source code used in those methods are publicly available. Moreover, the ground truth bounding boxes in some of those methods are not aligned with orientations of people. Thus, we compare our algorithms against two natural baseline methods that we designed ourselves.

In our first baseline algorithm $\text{YOLO}_{\text{fish}}$, we apply YOLO directly to fisheye images in order to detect people bounding boxes (using the same post-YOLO processing steps as in our proposed algorithms) and then rotate them to align them radially. This baseline is expected to be successful in the upper part of the image, where people are upright, but should be sub-par in the lower part. In the second baseline algorithm YOLO_{dew}, we first de-warp the fisheye image and then apply YOLO. The post-YOLO processing steps are the same as in our proposed methods. The coordinates of the final bounding boxes are then re-warped back to the fisheye domain. Since dewarping creates severe distortion in image center and at its periphery, the method should perform well only in the middle region. In both baseline algorithms we apply the same confidence thresholding and NMS that we used in our AB and AA algorithms (baselines have no focus windows to apply spatial outlier rejection).

5.3. Evaluation Metrics

As our primary evaluation metric we use the mean absolute error (MAE) between the true and predicted people counts in each frame: MAE_{fr}, and also the mean absolute error per person: MAE_p, which is equal to sum of all MAE_{fr} divided by the total number of people in all of the frames.

Although our main goal is people counting, for completeness we also report object-detection metrics (limited to the "person" object) that are used in object-detection challenges, such as PASCAL, VOC [8] and COCO [12]. The estimated people bounding boxes are examined one-by-one in the order of decreasing "objectness" score produced by YOLO. Each box is matched to an unclaimed ground truth box (if one is available) with which it has the maximum IOU. If there is no unclaimed ground-truth box or if the matching IOU is less than 0.5, then the estimated box is marked as a false positive (FP). Otherwise, it is marked as a true positive (TP) and its matching ground-truth box is marked as "claimed". Any unclaimed ground-truth box left at the end is marked as a false negative (FN). In addition to TP, FP, and FN for individual videos we also report the overall TP, FP, FN, and the F-score across all videos. Since object detection is not the focus of our work and in our problem we only have a single class, we do not vary the IOU or objectness thresholds to calculate an area under the ROC or precision-recall curves.

5.4. Results and Discussion

The AA, AB and the baseline algorithms are compared in terms of people-counting MAE in Table 2 and in terms of various people-detection performance metrics in Table 3.

In terms of people-counting performance, the AB method has lower MAE metrics (on average across all videos) than both baseline algorithms by over 60%. The AA method's MAE is lower by additional 6–11%. These results clearly demonstrate that the series of pre- and post-processing steps, that we proposed, enable YOLO to be effective in people counting from overhead fisheye images despite the fact that YOLO is not trained on fisheye images. In a video-wise comparison, both AA and AB methods outperform the two baseline methods (with AA outperforming AB) on "Meeting1", "Meeting2", and "Lab1" videos. However, $\text{YOLO}_{\text{fish}}$ has the lowest MAE for "Lab2" video, and performance of the AA method is worse than that of the AB method. We traced this anomalous result to two sources. First, we found that people in "Lab2" video appear mainly in the upper half of the field of view. Secondly, people in this video remain still for a long period of time, which is a challenge for the AA method. If a person goes undetected in a certain frame, he/she becomes part of the background

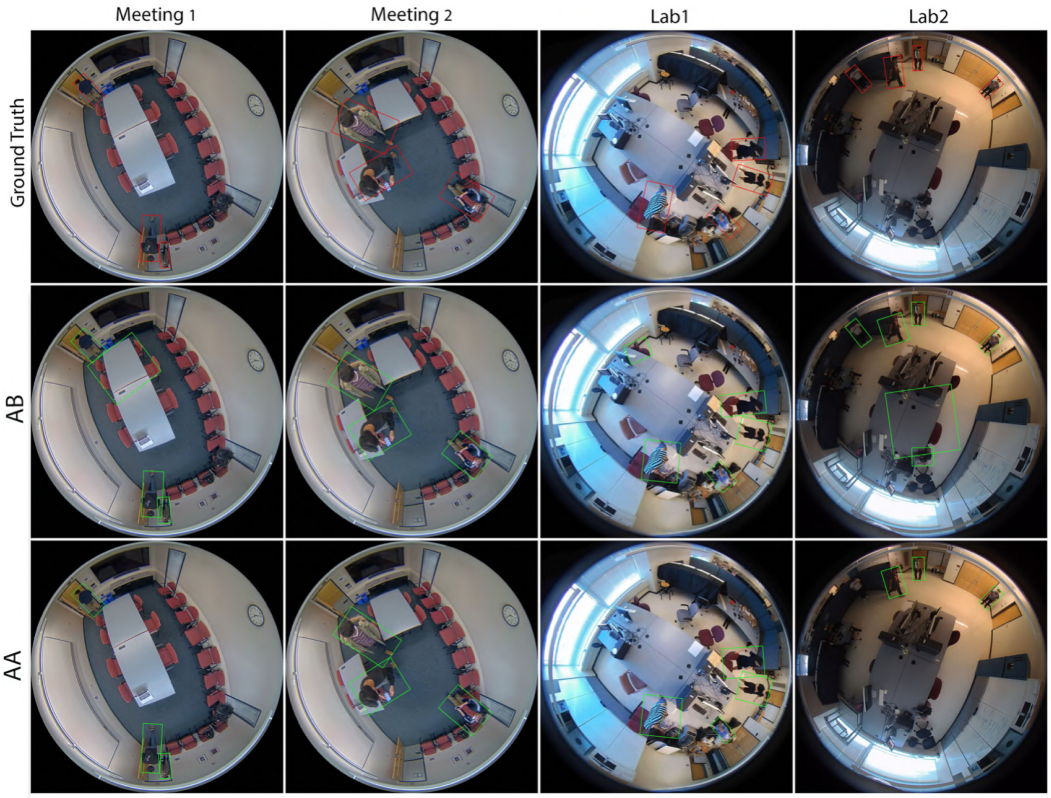


Figure 6: Qualitative comparison of people detection results on sample frames from each video (one per column) in the HABBOF dataset. Rows show the ground truth annotations as well as the AB and AA people detection results.

Table 2: Comparison of people-counting MAE per frame (MAE_{fr}) and per person (MAE_p)

Video	Meeting1		Meeting2		Lab1		Lab2		Overall	
Metric	MAE_{fr}	MAE_p	MAE_{fr}	MAE_p	MAE_{fr}	MAE_p	MAE_{fr}	MAE_p	MAE_{fr}	MAE_p
$YOLO_{fish}$	1.87	0.65	1.75	0.62	2.04	0.53	**0.62**	**0.16**	1.57	0.49
$YOLO_{dew}$	0.77	0.27	0.80	0.28	1.89	0.49	2.71	0.68	1.54	0.43
AB	0.44	0.15	0.33	0.12	0.78	0.20	0.84	0.21	0.60	0.17
AA	**0.28**	**0.10**	**0.22**	**0.08**	**0.54**	**0.14**	1.19	0.30	**0.56**	**0.15**

Table 3: Comparison of people-detection performance in terms of TP, FP, FN, and overall F-score

Video	Meeting1			Meeting2			Lab1			Lab2			Overall			
Metric	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN	F-sr.
$YOLO_{fish}$	766	342	2430	675	509	2466	1,742	1,735	5,713	4,728	1,428	2,495	7,911	4,014	14,103	0.43
$YOLO_{dew}$	2257	149	939	2,122	248	1,019	3,243	**336**	3674	2,249	**68**	4,964	9,817	**801**	10,519	0.67
AB	**2,896**	493	**300**	2,853	385	288	**5,970**	1,694	**947**	**6,464**	1,503	**749**	**18,183**	4,075	**2,284**	0.86
AA	2,813	**118**	383	**2,864**	**147**	**277**	5,674	530	1,243	5,041	159	2,172	16,392	954	4,095	**0.88**

until there is a significant change, thus causing miscounts.

In terms of people-detection performance, both AB and AA methods have larger TP and lower FN values than both baseline methods in each video and overall. Both proposed methods also achieve a higher overall F-score by over 0.2 compared to the baseline algorithms. Although the AB method detects more people (higher TP value) than the AA method, it has also a higher FP value. This is caused by the fact that most people appear within a single focus window in the AA method (focus window is constructed around ROI). However, even if a person is in the window's center, he/she may not be exactly upright and the detection by YOLOv3 may fail. In contrast, in the AB method a person appears in several neighboring focus windows, in each at a slightly different angle and in some with a fully-visible body, so that there are more chances for detection. Therefore, the activity-blind method outperforms the activity-aware method in terms of TP and FN and, consequently, has a higher *Recall*. On the other hand, the AB method has more FPs than the AA method, since the AA method implements

YOLO only on selected windows containing ROIs, and this reduces the chance of making an erroneous detection.

The proposed methods have difficulty in detecting people close to the center of the FOV where head and shoulders are the only visible body parts. This is because the YOLOv3 model we use is trained on the COCO dataset of regular-view images which does not have such examples.

6. Summary and Conclusions

We proposed two novel people-counting algorithms for overhead fisheye images by leveraging a state-of-the-art object-detection algorithm (YOLOv3) trained on standard side-captured images. In our activity-blind approach, we rotated the image 24 times to cover the entire field of view and produced detection results for the whole image. In the activity-aware approach, we used background subtraction as a pre-processing step to find the windows of interest in the frame and applied YOLO only to these windows. Since there are no annotated fisheye-image datasets with bounding boxes aligned with people's orientations, we collected and published a new, high-resolution HABBOF dataset, and used it for evaluation. Experimental results on this dataset show that both proposed algorithms outperform two baseline algorithms in terms of both people-counting and people-detection performance. Furthermore, the activity-aware method performs better than the activity-blind method on almost all HABBOF videos, which demonstrates the potential for using temporal information in people detection and counting in fisheye videos.

The new HABBOF dataset enables a fair, quantitative comparison of people-detection and people-counting algorithms for overhead fisheye videos, and may foster the development of new network architectures specifically designed for these two problems.

References

[1] Mirrow world challenge. https://icat.vt.edu/mirrorworlds/challenge/index.html.

[2] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-NMS improving object detection with one line of code. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[3] A.-T. Chiang and Y. Wang. Human detection in fisheye images using HOG-based detectors over rotated windows. In *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, July 2014.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, 2005.

[5] C. del Blanco and P. Carballeira. The PIROPO database. http://sites.google.com/site/piropodatabase, 2016.

[6] B. E. Demiroz, . Ari, O. Erolu, A. A. Salah, and L. Akarun. Feature-based tracking on a multi-omnidirectional camera dataset. In *International Symposium on Communications, Control and Signal Processing*, May 2012.

[7] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8), 2014.

[8] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1), Jan. 2015.

[9] R. Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(1), 2015.

[11] O. Krams and N. Kiryati. People detection in top-view fisheye imaging. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2017.

[12] 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 *European Conference on Computer Vision*. Springer, 2014.

[13] R. Nosaka, H. Ujiie, and T. Kurokawa. Orientation-aware regression for oriented bounding box estimation. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[15] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[16] M. Saito, K. Kitaguchi, G. Kimura, and M. Hashimoto. People detection and tracking from fisheye image based on probabilistic appearance model. In *IEEE International Symposium on Communications, Control and Signal Processing (ISCCSP)*, 2010.

[17] R. Seidel, A. Apitzsch, and G. Hirtz. Improved person detection on omnidirectional images with non-maxima supression. *arXiv:1805.08503*, 2018.

[18] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, June 1999.

[19] M. Tamura, S. Horiguchi, and T. Murakami. Omnidirectional pedestrian detection by rotation invariant training. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[20] T. Wang, C. Chang, and Y. Wu. Template-based people detection using a single downward-viewing fisheye camera. In *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Nov. 2017.

[21] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao. Oriented response networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.