Human Detection in Moving Fisheye Camera using an Improved YOLOv3 Framework

Olfa Haggui, Hamza Bayd, Baptiste Magnier, Arezki Aberkane

Human Detection in Moving Fisheye Camera using an Improved YOLOv3 Framework

Olfa Haggui, Hamza Bayd and Baptiste Magnier
EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales,
Alès, France
{Olfa.Haggui, Baptiste.Magnier}@mines-ales.fr

Arezki Aberkane
Technical Innovation Team,
Audensiel Technologies,
Boulogne-Billancourt, France

***Abstract*—Pedestrian detection has large relevance to the understanding of static and moving scenes of video sequences. The increasing demand for safety and security of people has resulted in more research on intelligent visual surveillance in a wide range of applications, such as moving human detection. With the great success of deep learning methods, researchers decided to switch from traditional methods based hand-crafted feature extractors to recent deep learning-based techniques in order to detect and track people. In this work, the topic of person detection with a Top-view moving fisheye camera is addressed. Although the fisheye camera is a useful tool for video monitoring, most of object detection techniques, with (or without) deep learning, concern classical perspective cameras. However, due to the distortions of fisheye images, we are expected to have higher requirements and challenges on the pedestrian detection using this device. In this paper, we propose an end-to-end learning people detection method based on YOLOv3 detector that detects people using oriented bounding boxes. The proposed model customizes the traditional YOLOv3 for the detection of oriented bounding boxes, by regressing the angle of each bounding box using a periodic loss function. With rotation bounding box prediction, our approach is efficient, reaching 98,1% of true detection. The proposed method is evaluated on a new available dataset where rotated bounding boxes represent annotations from several fisheye videos: https://partage.imt.fr/index.php/s/nytmFqiq8jaztkX**

keywords *Human detection, Fisheye camera, YOLOv3.*

I. Introduction

During the last few years, significant progress has been made in computer vision for people detection and tracking challenges, notably with the advancement of network technology. Within this context, typical cameras used in visual surveillance include perspective and fisheye cameras. Most of the existing researches are using perspective cameras, as they generate views similar to human vision, with small image distortions. However, its main disadvantages concerns the limited field of view. Therefore, the direct use of algorithms for classical cameras is not directly applicable on fisheye images.

People detection via video frames captured by fisheye cameras has received massive attention due to a certain number of advantages in visual surveillance application such as the large field of views. Yet, the major challenge is to take into consideration the radical distortions obtained in the image. Furthermore, the pedestrians in a fisheye image appear in different shapes, sizes and at various orientations, such as upright, upside-down, horizontal or diagonal. Unfortunately, most of the existing people-detection algorithms are designed for standard camera images where people appear upright. This paper focuses on the problem of people detection from video sequences recorded by Top-view moving fisheye cameras, as represented in Fig. 1, right. Over the past decade, a significant improvement has been witnessed with the help of traditional handcrafted features and models based on end-to-end learning. Among traditional people-detection algorithms, the most popular ones for pedestrian detection, HOG (Histogram of Oriented Gradients) and ACF (Aggregate Channel Features) have been used with overhead fisheye images. In [1], the popular human detection algorithm based on the Histogram of HOG features and SVM (Support Vector Machines) classifier are combined after rotating each search window on a radial line to the vertical reference line. Another method in [2] is based on HOG and LBP (Local Binary Patterns) features and SVM classifier to model people as upright cylinders and derived a series of elliptic detection masks whose size diminishes with the distance from the image center. In [3], ACF are trained on side-view, standard-lens images for pedestrian detection without unwrapping a fisheye image into a panoramic image.

Recently, with the advent of Deep Learning, numerous benchmarks and datasets have been created in order to train and evaluate people detection algorithms with high accuracy in real-time. Some algorithms based on classification worked in two stages. First, the Regions of Interest (ROIs) are detected. This step represents a preprocessing, consisting of an image division into several regions using basic segmentations based on the colors or contours. Then, those regions are classified using Convolutional Neural Networks (CNN) or SVM. This process is very slow because every selected region must be predicted. In this context, the most popular algorithms are the Region-based convolutional neural network (RCNN) and the versions Fast-RCNN [4] and Faster-RCNN [5].

Instead of a selection of ROIs from the image, classes and bounding boxes (BBoxes) are predicted for the image in one run of the algorithm based on regression as in YOLO (You Only Look Once) [6] and SSD (Single-Shot multibox Detection) mobilenet [7] algorithms. Much research has addressed the topic of Top-view person detection with a static fisheye camera, mostly YOLO-based. In [8], a rotation invariant training method is applied, using randomly rotated standard images, without any additional annotation to simulate various poses and orientations of people in fisheye images. Another

YOLO-based people detection method adapts YOLOv3 trained on standard images for people counting [9]. Each image is rotated in 15° steps and YOLO is applied to the top–center part of the image followed by post-processing to remove multiple detections. Recently, the algorithm proposed in [10] provides much faster and more accurate results than previous algorithms aiming people detection in fisheye images, without any pre-processing. Its goal is to predict BBoxes of people, with certain center and size, but also the angle of each BBox.

Compared with the existing works, the technique presented in this paper can operate the detection of people in a complex scene recorded by Top-view moving fisheye cameras. No constraints on the peoples' movements is established, i.e., people can stand, sit, walk, kneel down, push objects and occlude each other for long periods of time. Moreover, this method does not require any camera calibration. To achieve this work, a new Top-view people detection dataset is introduced.

II. Oriented People Detection via Fisheye Cameras

A. Fisheye Camera Description

Usually, omnidirectional and fisheye cameras offer panoramic views of 2π radian angles [11]. Specific mirrors equip catadioptric cameras, whereas only lens concerns fisheye devices; then its angle of view can attain 2π radian angle or more. Therefore, objectives with wide-angle lens capture images typically warped, creating the effect of a fisheye. Fisheye cameras represent a major asset for several applications. In this way, these cameras are popular in many fields of computer vision, robotics and photogrammetric tasks such as navigation, localization, tracking, mapping and so on.

A fisheye camera is a camera fixed to a front lens group which appears as a single "big" lens, as shown in Fig. 1, left. This device enables a far greater negative refraction power than usual lenses, allowing greatly increasing the back focal distance and embracing wider fields of view [12]. In the context of people detection, the wide field of vision provided by these cameras makes people look inclined and distorted. Consequently, standard detection and tracking techniques are not reliable on warped images, especially with a cluttered and moving background [13]. Moreover, specific detectors for unconventional cameras are hard to design because they need a calibration stage which could be difficult to design [14]. Even though many algorithms already exist for standard images, people detection and tracking regarding top-view images acquired by fisheye cameras are not a very documented topic and demand very specific involvement to work consistently.

B. Top View People Detection via Fisheye Cameras

1) Overview: The focus here deals with real time people detection using a moving fisheye camera. There exist many methods for people detection and tracking using conventional camera, as referenced in [15]. However, people detection using fisheye cameras has been barely studied, due to the complexity of the device caused by the distortion effects. Additionally, in our investigation, we give extra focus on a people detection in a moving scenes in real-time. In recent methods, pedestrian detectors are trained using fisheye images, even though the manual labeling remains a hard task, which consumes times. Therefore, a new strategy for learning fisheye pedestrian detectors using images from a selected pedestrian dataset is proposed in this work. Another important development constraint, this detector should be equally applicable on a visual moving sensor that is either fixed in the environment or mounted on a mobile platform (like an aerial drone). To accommodate these challenges, a CNN model based on YOLOv3 detector is employed for person detection using top view video fisheye frames. This model is illustrated in Fig. 2. Its goal is to predict BBoxes of people, with certain center point position and size (width and height), but also the angle of each BBox. The angles of the BBoxes represent an important clue for the training or the detection. Indeed, rectangular BBoxes meet difficulties for object localization with different orientation angles, as produced by fisheye lens.

Actually, the proposed detector in this paper is a fully CNN with an architecture based on YOLOv3, and is configured to detect only one class, i.e., a person. In that respect, the network is structured in three parts. The first one represents the backbone network, known as Darknet-53, trained on the ImageNet database [16]. Its main goal is to extract features at different spatial resolutions; it takes an input image and outputs a list of features from different parts of the network. Darknet-53 mainly consists of two blocks: residual and convolutional blocks. Each uses successive 1×1 and 3×3 convolution with doubly increasing filter channels, as well as shortcut connection between input and convolutional output, as summarized in Fig. 2. The second part concerns the Features Pyramid Network (FPN) [17]. This network takes as input the multi-resolution features computed by our Draknet53 backbone in order to extract features related to person detection. In fact, FPN contains information about small and large objects. We expect D_1^{FPN}, the output of the FPN, to contain information about small objects and D_3^{FPN}, the output about large objects. The construction of this pyramid involves a bottom-up pathway, a top-down pathway, and lateral connections as shown in FPN block of the Fig. 2. The bottom-up pathway is the feed-forward computation of the backbone ConvNet, which computes a feature hierarchy consisting of feature maps at several scales with a down-scaling step of 2. The output of the last layer of each stage will be used as the reference set

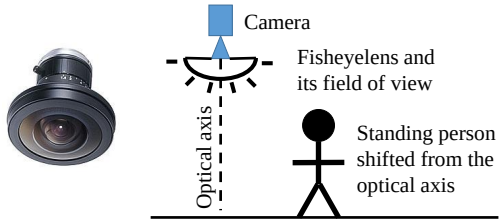


Fig. 1. Fisheye camera. On the left: fisheye lens used in our experiments: 2/3" Format C-Mount fisheye lens 1.8mm FL, with a horizontal field of view, 1/2" sensor for 185°. On the right: diagram of the experimental protocol.

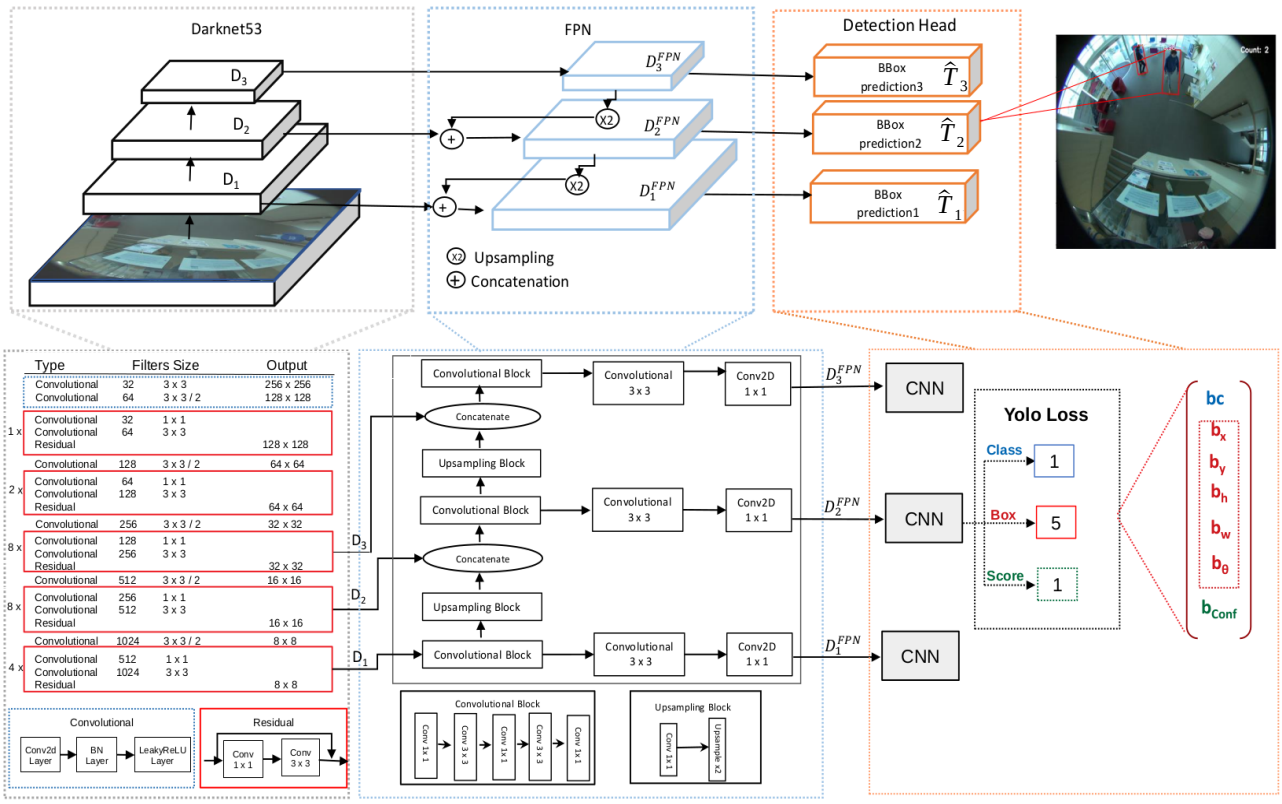


Fig. 2. Architecture of the proposed network. input: fisheye image, backbone (Darknet53), Features Pyramid Network (FPN), and detection head (BBox regression network), Oriented BBoxes as outputs. For the YOLO loss function, only one class, 5 parameters for each BBox and a confidence score.

of feature maps for enriching the top-down pathway by lateral connection. For the top-down pathway, the higher resolution features are up-sampled by a factor of 2 spatially coarser, but semantically stronger, from higher feature maps of pyramid levels. These features are then merged with features from the bottom-up pathway via lateral connections. Finally, the third part is the head detection, allowing building a tensor $\hat{T}_{1,2,3}$, containing information on the BBox position, including its angle of rotation. The implemented model uses a loss function combining Binary Cross Entropy (BCE, see Eqs. 2 and 3), as described in YOLOv3 [6] [18], and, a periodic loss function that regresses the angle of each BBox, accounting for angle periodicities [19] [10] [9]. Therefore, the detection of oriented objects is an extension of a general horizontal object detection.

2) Oriented Bounding Box Detection: In fisheye images, since most targets have an orientation, neither vertical, nor horizontal, rotated object detection is essential for overhead people detection (an example is available in Fig. 4). In our case, outputs of an improved YOLOv3 [6] network are used with both horizontal location boxes and angle information, rendering YOLOv3 module more sensitive to the angle. By introducing the oriented BBoxes, for each video frame, the predicted results of the proposed framework return six BBox parameters: the position coordinates (b_x, b_y), the BBox size (b_w, b_h) and the angle of all individuals b_θ. They are represented by a six-dimensional vector $(b_x, b_y, b_w, b_h, b_\theta, b_{Conf})$, where b_{Conf} is the predicted confidence score; it quantifies how confident the algorithm sounds that the target represents a human being. In addition, there is a confidence threshold, determined by the user, but usually fixed to 0.5, and the algorithm only returns the BBoxes whose confidence score is higher than this threshold. The Fig. 3 shows the transform from the anchor to the BBox where the coordinates center (b_x, b_y) of BBox is calculated by applying a sigmoid to predicted values and adding the corner points of the corresponding grid cell. Meanwhile, the dimensions b_w and b_h of the BBox are calculated by applying a log-space transform to the predicted output dimensions and then multiplying with an anchor dimensions (p_w, p_h). The network establishes a multitask and crucial loss function (Eq. 1) inspired by that used in YOLOv3, with an additional BBox rotation-angle loss to optimize the target detection. It is computed by the ground truth and the predicted result of the network:

$$Loss = Loss_{Box} + Loss_{Conf} + Loss_{Angle}, \qquad (1)$$

where the Box regression loss ($Loss_{Box}$ in Eq. 2) is calculated only when the prediction box contains detected people. Confidence loss ($Loss_{Conf}$ in Eq. 3) determines whether there are persons in the prediction frame. BBox rotation-angle loss ($Loss_{Angle}$ in Eq. 4) determines the prediction orientation of a person. Note, that the category-classification loss is not used since only one class (i.e., persons) is used here. Here, $\hat{T} = (\hat{t_x}, \hat{t_y}, \hat{t_w}, \hat{t_h}, \hat{t_\theta}, \hat{t_{conf}})$ represents the transformed

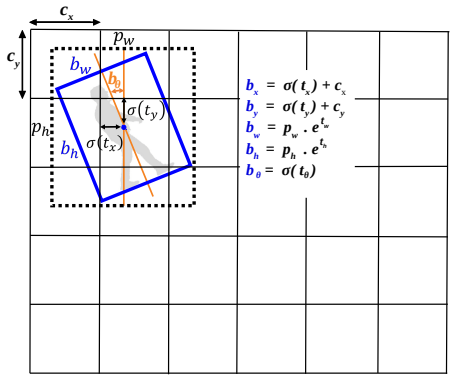


Fig. 3. Oriented Bounding Box (BBox) with its tied parameters.

version of BBox predictions for each stride s_k, from which a BBox prediction $\hat{b} = (\hat{b_x}, \hat{b_y}, \hat{b_w}, \hat{b_h}, \hat{b_\theta}, \hat{b}_{conf})$ is computed as represented in Fig. 3, where (t_x, t_y, t_h, t_w) is calculated from the ground truth $b = (b_x, b_y, b_w, b_h, b_\theta, b_{conf})$. These loss functions are given by the 3 following formulas:

$$\begin{aligned} Loss_{Box} = &\sum_{\hat{t} \in \hat{T}^+} BCE\left(\mathcal{S}(\hat{t}_x), t_x\right) + BCE\left(\mathcal{S}(\hat{t}_y), t_y\right) \\ &+ \sum_{\hat{t} \in \hat{T}^+} \left(\mathcal{S}(\hat{t}_w) - t_w\right)^2 + \left(\mathcal{S}(\hat{t}_h) - t_h\right)^2, \end{aligned} \tag{2}$$

$$Loss_{Conf} = \sum_{\hat{t} \in \hat{T}^+} BCE\left(\mathcal{S}(\hat{t}_{conf}), 1\right) + \sum_{\hat{t} \in \hat{T}^-} BCE\left(\mathcal{S}(\hat{t}_{conf}), 0\right), \tag{3}$$

with $\mathcal{S}$ the logistic sigmoid activation function, $(\hat{T}^+, \hat{T}^-)$ positive and negative samples from the predictions respectively.

$$Loss_{Angle} = \sum_{\hat{t} \in \hat{T}^+} L_{Angle}\left(\hat{b}_\theta, b_\theta\right), \tag{4}$$

such that for a given range (α, β), $\hat{b}_\theta = \alpha \cdot \mathcal{S}(\hat{t}_\theta) - \beta$, which is the predict of b_θ and the function L_{Angle} is defined by

$$L_{Angle}\left(\hat{b}_\theta, b_\theta\right) = R\left(\text{mod}\left[\hat{b}_\theta - b_\theta - \frac{\pi}{2}, \pi\right] - \frac{\pi}{2}\right),$$

with "mod" representing the modulo operation and R a symmetric regression function.

III. Implementation Details

A. Dataset Description

Numerous benchmarks and datasets have been created in order to train and evaluate people detection algorithms regarding fisheye images. Most of the existing public fisheye datasets are annotated by an aligned BBox. In this work, a dataset of overhead fisheye images with oriented BBoxes is needed for each person aligned with its orientation in the image. However, different challenges are reported with the dealing of fisheye images, mainly spatial and temporal illumination variations, occlusions, and various body poses for example. Additionally, when people are walking straight under the camera and at the periphery of the fisheye image, the appearance is different and the image resolution is small near the borders, disturbing regularly the detection. To overcome these challenging scenarios of different videos captured from a moving fisheye camera, a new dataset has been collected and annotated. It is called Oriented Bounding Boxes from Moving Fisheye cameras (OBBMF) and is composed of 6 videos (see Tab. I for details). Clearly, the new dataset contains many more frames and human objects, and also includes challenging scenarios, which do not exist in the other datasets. Furthermore, experiments are performed using three public datasets, MW-18Mar[1], HABBOF[2], CEPDEOF[3], and our datasets in order to fit and evaluate the effectiveness of our proposed method.

Video	#person(s)	#frames	fps	Description/Challenges
Stairs	2	500	48	Person go up and down the stairs with rotational movement of camera
Parking	2	536	48	Person walking, body camouflage with the scene
Window	2	534	48	Person in a top-view position and non uniform illumination
Workshop	5	530	48	More than 4 walking and sitting in a large space
Entrance1	2	543	48	Person walking and sitting in center and boundary of image
Entrance2	1	567	48	Walking activity in reception room with Top-view challenge

TABLE I
DESCRIPTIONS OF VIDEO SEQUENCES AND THEIR TIED CHALLENGES

B. Data Acquisition

In our case, the data are collected in the CERIS laboratory of the IMT Alès research center. The videos were collected with fisheye camera (Basler ace acA1300-200uc) facing down at 48 fps, where one or several persons perform under the camera various poses such as walking and sitting. Also severe body occlusions are present, and people go up and down the stairs with rotational movement. Then, a number of frames are generated from these video clips which contains new scenes with some challenging scenarios such as illumination, camera rotation and motion in the center (Top-view), which are generally unavailable in the common literature

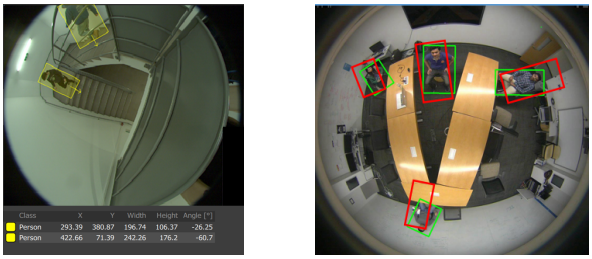
C. Data Annotation and File Conversion

Data labelling is an essential step in a supervised machine learning task requiring a lot of manual work. To annotate our new dataset, person regions are manually annotated with rotated BBoxes. In order to do so, the MVTec Deep Learning Tool [4] is used. It is a very useful target detection and labeling deep learning tool in a short time, but it generates a *hdict* file which cannot be directly used in other deep learning tools. So, it is converted into *txt* file in the first step using a small application with C# and HalconDotNet-V. 19.11.0, and finally converted again to the data format of YOLOv3 as a *json* file. Technically, first, the main axis of the rectangle is drawn to define the orientation of the person in the scene. Then the width and height of the person in pixels are defined. Consequently, each BBox is represented by five parameters:

- (x, y): coordinates of the BBox center,
- w and h: width and height of the BBox respectively,
- θ: clock-wise rotation angle from the vertical axis.

Furthermore, Figs. 4(a) and (b) represent these parameters in a rectangle BBox. The whole of our data set is represented

[1] http://www2.icat.vt.edu/mirrorworlds/
[2] http://vip.bu.edu/projects/vsns/cossy/datasets/habbof/
[3] http://vip.bu.edu/projects/vsns/cossy/datasets/cepdof/
[4] https://www.mvtec.com/products/deep-learning-tool



(a) Frame annotation with BBox angles (b) Frame and BBoxes of MW-R dataset

Fig. 4. Examples of annotated frames with BBox rotation angle. In (b), a frame and its BBoxes from MW-R dataset (the red BBoxes are the initial tied to the MW-R dataset whereas the green correspond to the corrected).

by more than 12.000 OBBMF. However, some annotations in the MW database contain errors in the angles, as shown in Fig. 4(b). Consequently, these annotations have been manually modified to improve the relevance of this database. It allows bringing a very efficient model for human being detection.

D. Training Datasets

Concerning training, a pre-trained Darknet53 model was used as a starting point, which is initialized with ImageNet pre-trained weights for faster training. Also, in order to train the proposed detector, we used one of the largest dataset MS COCO [20] that is commonly utilized as a general object detection benchmark, because it contains various appearances of people. Our network was trained end to end by optimizing the cross entropy loss function by updating weights using "Stochastic Gradient Descent" (SGD) [21] with a momentum of 0.9 for more than 50,000 iterations (one iteration contains 128 images). The SGD represents an iterative optimization technique [22], and is most widely used in the field of deep learning to minimize the loss function to search hyper-parameters. This algorithm calculates the gradient and makes the update of the network parameters by the mean of the training set'subset, which is called mini-batch. Each gradient evaluation using the mini-batch is defined as an iteration. At each iteration, the algorithm takes one step to minimize the loss function. The complete progress of the training algorithm over the total training set using mini-batches is called an epoch. During experiments, the initial learning rate is set to 0.001 and the weight decays to 0.0005. The mini-batch size is set to 16, and the network is trained for 500 epochs. We have set the learning rate factor to 0.0001 with the same SGD parameters and batch size to fine-tune parameters of the network. All the process was trained on multiple cross fisheye datasets for more than 8000 iterations from weights pre-trained in ImageNet using COCO. For these two networks, the images were resized into 608×608 pixels and fed into the network until the loss has been saturated. Our model was conducted on an NVIDIA Quadro P5000 GPU accelerator (Pascal architecture). It includes 2560 CUDA cores with 16 GB GDDR5 memory. The host is an Intel® Xeon® CPU E5-1620 V4 processors with 4 cores.

IV. Experimental Results and Evaluations

A. Evaluation Metrics

The detection system returns a list of detected BBoxes in an image. The match of a detected BBox and the ground truth is rated by asserting an overlap area of more than 50%. To quantitatively evaluate the performance of the proposed network, the statistical analysis of $Precision$, $Recall$, F-score and Average Precision (AP) are performed as the evaluation metrics. For TP, FP and FN denoting le number of true positives, false positives and false negatives in a video, $Precision$ means the percentage of the correctly detected persons (TP) over all the detected persons ($TP + FP$): $Precision = \frac{TP}{TP+FP}$. Meanwhile, $Recall$ is the ability of a model to find all the objects. It associates with the correct predictions among all the positive cases, which means the percentage of the correctly detected: $Recall = \frac{TP}{TP+FN}$. Hense, $Precision$ and $Recall$ are considered to be common evaluation metrics, and the F-score combines the two:

$$F = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}. \quad (5)$$

Finally, Average Precision (AP) is the area under the Precision-Recall curve: $AP = \int_0^1 F(x)\,\mathrm{d}x$. Therefore, the closer the evaluation scores of both F and AP are to 1, the more the detection is qualified as suitable. On the contrary, a score close to 0 corresponds to a poor detection of persons.

B. Benchmark Results

Various experiments are presented to analyze the performance of the proposed method. We fine-tuned the algorithm trained on COCO with various cross datasets from MW-R, CEPDEOF, HABBOF and OBBMF. Hence, we cross-validate on these datasets, i.e., two datasets are used for training, then they are tested on another dataset. For example, a cross dataset trained on MW-R + HABBOF is tested on OBBMF, and inversely. Tab. II shows the detection performance of our method on each video in the OBBMF dataset obtained by fine-tuning with cross-validation1 (index $cross1$) and cross-validation2 (index $cross2$), respectively. Our model with 608×608 resolution achieves impressive performance, despite of using videos captured from moving fisheye camera. Thus, the proposed method performs outstandingly with an acceptable convergence behaviours in several experiments carried out with various cross validation datasets.

The overall performance metrics obtained with our model and with our OBBMF dataset are evaluated using AP,

	Performance metric					
	AP_{50}	AP_{75}	AP_{90}	Precision	Recall	F-measure
$stairs_{cross1}$	0.857	0.478	0.467	0.810	0.808	0.852
$Workshop_{cross1}$	0.785	0.306	0.396	0.818	0.649	0.756
$Window_{cross1}$	0.891	0.511	0.502	0.901	0.687	0.765
$Parking_{cross1}$	0.864	0.501	0.490	0.903	0.762	0.894
$Entrance1_{cross1}$	**0.970**	0.576	0.564	0.972	0.936	0.963
$Entrance2_{cross1}$	0.686	0.432	0.556	0.791	0.692	0.637
$stairs_{cross2}$	0.911	0.432	0.544	0.901	0.911	0.839
$Workshop_{cross2}$	0.929	0.402	0.661	0.849	0.912	0.918
$Window_{cross2}$	0.916	0.642	0.706	0.791	0.892	0.883
$Parking_{cross2}$	**0.971**	0.557	0.691	0.913	0.992	0.926
$Entrance1_{cross2}$	0.896	0.530	0.656	0.891	0.892	0.837
$Entrance2_{cross2}$	0.686	0.432	0.556	0.893	0.992	0.902

TABLE II

Performance comparison of our method with cross validation $cross1$ and $cross2$ for each video in OBBMF dataset.

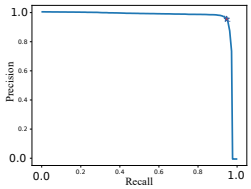


Fig. 5. Precision-recall curve of the best model on the test set. The precision remains close to 100 % for recall values as high as 85%. The optimal point (the closer to the upper-right corner) is at 0.981.

	Performance metric					
	AP_{50}	AP_{75}	AP_{90}	Precision	Recall	F-measure
Lab2	**0.980**	0.863	0.673	0.977	0.875	0.883
stairs	0.936	0.717	0.598	0.960	0.925	0.873
Lunch2	**0.971**	0.254	0.446	0.976	0.969	0.973
Meeting2	**0.978**	0.720	0.594	0.977	0.957	0.967
Workshop	0.942	0.817	0.655	0.942	0.925	0.892
MW-R18	0.941	0.690	0.514	0.955	0.899	0.894

TABLE III
PERFORMANCE TUNING EVALUATION OF OUR METHOD WITH MULTI CROSS VALIDATION FOR EACH VIDEO IN EACH DATASET.

$Precision$, $Recall$ and F-measure. As shown in the Tab. II, the new algorithm performs efficiently on ordinary videos with a score more than 0.95 for AP. However, more complex scenes (moving camera, low light, strong shadows, etc.) remain challenging. The Fig. 6 shows sample results applied to the four datasets where detections are nearly perfect in a range of scenarios, such as various body poses, orientations, and diverse background scenes. For this reason, we expand our training datasets by cross validating various samples from all the used datasets, in order to improve the resulting performance model and prevent it from over-fitting. The resulting cross dataset was split: 70% used for the training stage and 30% from data are used for testing. On the one hand, the Tab. III shows clearly the improved performance with large scales datasets, exceeding 90% of the AP. On the other hand, the $P-R$ ($Precision-Recall$) curve of the best model on the test set is plotted in the Fig. 5; it shows the trade-off between the $Precision$ and the $Recall$ as the threshold score of the model changes. $Recall$ should increase to guarantee that all the persons are detected. However, as $Recall$ increases, it is common for some scenarios, such as distortion, camera movements, low light, and strong shadows, decreasing $Precision$. Ideally, the upper-right corner of the curve should reflect 100% of $Recall$ and $Precision$, often impossible to obtain in real scenarios. Eventually, $P-R$ curve illustrates why the AP of our model is high. Indeed, $Precision$ remains close to 100% for $Recall$ values as high as 85%, with a 0.98% for the optimal point.

V. CONCLUSION

An approach is proposed in this paper to detect people in Top-view fisheye images, using moving camera. It is based on a pre-trained deep CNN architecture, extended from YOLOv3 detector. Experimental results confirm that people are extracted in an indoor environment using videos streaming from a fisheye moving camera with a high level of AP. Our approach eliminates the need for pre-processing and/or data augmentation, by considering oriented bounding boxes. Finally, a new dataset of videos has been created regarding human detection with a top view fisheye moving camera; the great interest is that the ground truths are also available online.

For future works, we plan to expand the proposed model for more large and complex datasets at different image resolutions. We believe both that our method and dataset will be beneficial for various real-world applications, especially for human detection and tracking using overhead fisheye videos captured and treated automatically from an aerial drone.

REFERENCES

[1] A.-T. Chiang and Y. Wang, "Human detection in fish-eye images using hog-based detectors over rotated windows," in *ICME Workshops*. IEEE, 2014, pp. 1–6.

[2] T. Wang, C.-W. Chang, and Y.-S. Wu, "Template-based people detection using a single downward-viewing fisheye camera," in *ISPACS*, 2017, pp. 719–723.

[3] M. Demirkus, L. Wang, M. Eschey, H. Kaestle, and F. Galasso, "People detection in fish-eye top-views," in *VISIGRAPP*, 2017, pp. 141–148.

[4] H. Lin, Z. Kong, W. Wang, K. Liang, and J. Chen, "Pedestrian detection in fish-eye images using deep learning: Combine faster r-cnn with an effective cutting method," in *SPML '18*, 2018.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[7] J. Yu, R. Seidel, and G. Hirtz, "Omnipd: One-step person detection in top-view omnidirectional indoor scenes," *Current Directions in Biomedical Engineering*, vol. 5, pp. 239 – 244, 2019.

[8] M. Tamura, S. Horiguchi, and T. Murakami, "Omnidirectional pedestrian detection by rotation invariant training," *WACV*, pp. 1989–1998, 2019.

[9] S. Li, M. Tezcan, P. Ishwar, and J. Konrad, "Supervised people counting using an overhead fisheye camera," in *IEEE AVSS*, 2019, pp. 1–8.

[10] Z. Duan, O. Tezcan, H. Nakamura, P. Ishwar, and J. Konrad, "Rapid: rotation-aware people detection in overhead fisheye images," in *IEEE/CVF CVPR Workshops*, 2020, pp. 636–637.

[11] D. Scaramuzza and K. Ikeuchi, "Omnidirectional camera," 2014.

[12] J. J. Kumler and M. L. Bauer, "Fish-eye lens designs and their relative performance," in *Current developments in lens design and optical systems engineering*, vol. 4093. International Society for Optics and Photonics, 2000, pp. 360–369.

[13] O. Haggui, M. Agninoube Tchalim, and B. Magnier, "A comparison of opencv algorithms for human tracking with a moving perspective camera," *IEEE EUVIP -to appear-*, 2021.

[14] M. Boui, H. Hadj-Abdelkader, F.-E. Ababsa, and E. H. Bouyakhf, "New approach for human detection in spherical images," in *IEEE ICIP*, 2016, pp. 604–608.

[15] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE TPAMI*, vol. 34, no. 4, pp. 743–761, 2011.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE CVPR*, 2009, pp. 248–255.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE CVPR*, 2017, pp. 2117–2125.

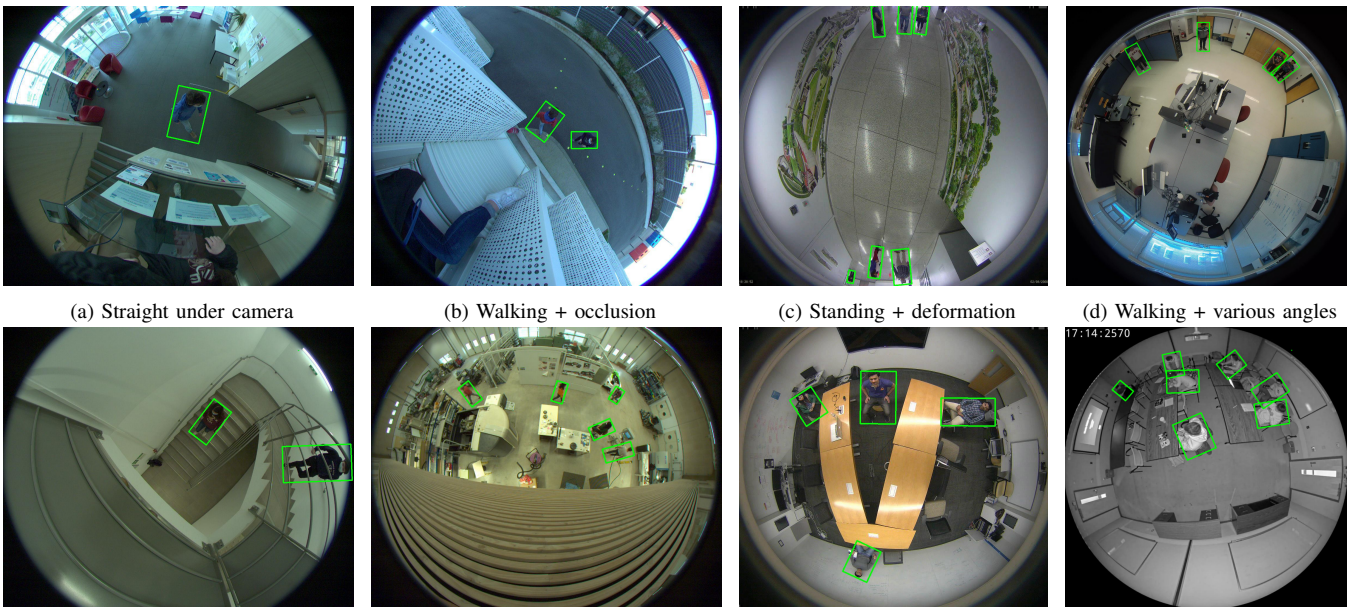
[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE CVPR*, 2016, pp. 779–788.

[19] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *IEEE/CVF CVPR*, 2019, pp. 2849–2858.

[20] 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 *ECCV*. Springer, 2014, pp. 740–755.

[21] A. Ramezani-Kebrya, A. Khisti, and B. Liang, "On the generalization of stochastic gradient descent with momentum," *arXiv preprint arXiv:2102.13653*, 2021.

[22] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*. Springer, 2010, pp. 177–186.



(a) Straight under camera (b) Walking + occlusion (c) Standing + deformation (d) Walking + various angles

(e) Walking + rotation (f) Various angles and poses + far (g) Different poses (h) Low-light scenario

Fig. 6. Detection results of our benchmark on sample frames in different scenarios and challenge, including various poses, orientations and background scenes. Green boxes are predicted BBox (true positives, i.e., matching of a detected BBox and the groundtruth with an overlap area of more than 50%).