Fisheye-DETRAC: A New Fisheye Video Benchmark for Multi-Object Detection and Tracking

[1]*Ping-Yang Chen* (陳平揚), [2]**Jun-Wei Hsieh* (謝君偉), [3]*Ming-Ching Chang* (張明清),
[4]*Chung-I Huang* (黃仲誼), [4]*Wei-Yu Chen* (陳威宇),
[5]*Munkhjargal Gochoo (Moyo)*, and [4]*Fang-Pan Lin* (林芳邦)

[1]Department of Computer Science,
National Yang Ming Chiao Tung University, Hsinchu, Taiwan

[2]College of AI and Green Energy,
National Yang Ming Chiao Tung University, Hsinchu, Taiwan

[3]Department of Computer Science,
University at Albany, State University of New York, USA

[4]National Center for High-Performance Computing, Hsinchu, Taiwan

[5]Department of Computer Science and Software Engineering
United Arab Emirates University, United Arab Emirates

E-mail: jwhsieh@nycu.edu.tw

ABSTRACT

Fisheye lenses inherently offer a wider, omnidirectional coverage area compared to traditional cameras, which can reduce the number of cameras required for intersection monitoring. In our study, we introduce a new large-scale **Fisheye DE**tection and **TRAC**king (Fisheye-DETRAC) dataset. This dataset is designed for the training and assessment of 2D road object detection and multiple object tracking from fisheye cameras, containing a total of 470K bounding boxes spanning five classes: Pedestrian, Bike, Car, Bus, and Truck. The dataset includes 20,000 images, 157,000 bounding boxes, and 313,204 identities captured in 27 videos. These videos were recorded using 22 fisheye cameras deployed for traffic monitoring in Hsinchu, Taiwan, with resolutions of 1080×1080, 1920×1920, and 1280×1280. These images exhibit significant distortion and often feature numerous road users, particularly people on scooters. This paper further focuses on vehicle tracking and proposes a novel Hybrid Data Association (HDA) method for tracking vehicles directly from a fisheye camera. The benchmark is available at https://dakors.com, providing annotation formats compatible with PASCAL VOC, MS COCO, YOLO, and MOT. The Fisheye-DETRAC dataset promises to be a substantial contribution to the field of fisheye video analytics and smart city applications.

Index Terms— Fisheye Benchmark, Fisheye Camera, Multiple Object Tracking (MOT), Object Detection

1. INTRODUCTION

Traffic flow estimation is a key task for monitoring and managing traffic streams in an intelligent transportation system. To estimate traffic flows from a whole multi-lane intersection, a fisheye camera will be more suitable compared to an CCTV camera due to its wider Field of View (FOV).

In recent years, fisheye camera applications attracts growing attentions, as 360^o omni-directional wide coverage can be easily obtained when compared with the narrow FOV of traditional cameras. Employing fisheye cameras for traffic monitoring systems reduces the required number of cameras for monitoring areas such as street intersections.

In the last decade, the amount of road traffic object detection datasets in the literature has increased greatly, as traffic monitoring is an important research topics in computer vision; see Table 1 for an overview. However, to the best of our knowledge, there is no open competition website constructed from fisheye traffic surveillance cameras for road object detection and multiple object tracking tasks. The only exception is the fisheye based road dataset [11] captured by a car dash camera for self-driving vehicle usage.

In this study, we introduce a new large-scale **Fisheye DE**tection and **TRAC**king (Fisheye-DETRAC) dataset that is specifically designed for the training and assessment of fisheye road object detection and multiple object tracking tasks. It contains a total of 470K bounding boxes spanning five

Table 1. **Summary of existing road traffic datasets.** The **Frame** column indicates the number of images containing at least one object on them ($1K = 10^3$). The **Boxes** column indicates the unique object bounding boxes. In the remaining columns, '+' indicates the availability of a supported feature, 'D' indicates the target is a detection task, '3D' indicates a three-dimensional detection task, 'T' indicates a tracking task, and 'Seg' indicates a segmentation task.

Dataset	Frame	Boxes	Task	Vehicles	Pedestrian	Weather	Occlusion	Altitude	View	Classes	Location	Type
MIT-Car 2000[1]	1.1K	1.1K	D	+						-	Surveillance	2D
KITTI-D 2014[2]	15K	80.3K	D	+	+		+			3	Car	2D
UA-DETRAC 2015[3]	140K	1210K	D,T	+		+	+			4	Surveillance	2D
Detection in LLC 2017[4]	7.5K	15K	D	+		+				12	Car	2D
CARPK 2017[5]	1.5K	90K	D	+						-	Drone	2D
UAVDT 2017[6]	80K	841.5K	D,T	+		+	+	+	+	-	Drone	2D
NEXET 2017[7]	50K	-	D	+		+				5	Car	2D
BDD100k 2018[8]	5.7K	-	D,T	+	+	+				10	Car	2D
AAU RainSnow 2018[9]	2.2K	13297	D,Seg	+		+					Surveillance	RGB&Thermal
MIO-TCD CCTV 2018[10]	113K	200K	D	+		+				5	Surveillance	2D
BDD100k Adas 2018[8]	100K	250K	D,Seg	+		+				10	Car	2D
Woodscape 2018/2019[11]	10K	-	D,3D,T	+		+				7	Car	Fisheye
CityFlow2D 2021[12]	-	313.9K	D,T	+							Surveillance	2D
Fisheye-DETRAC [**our**]	20K	470.0K	D,T	+	+				+	5	Surveillance	Fisheye

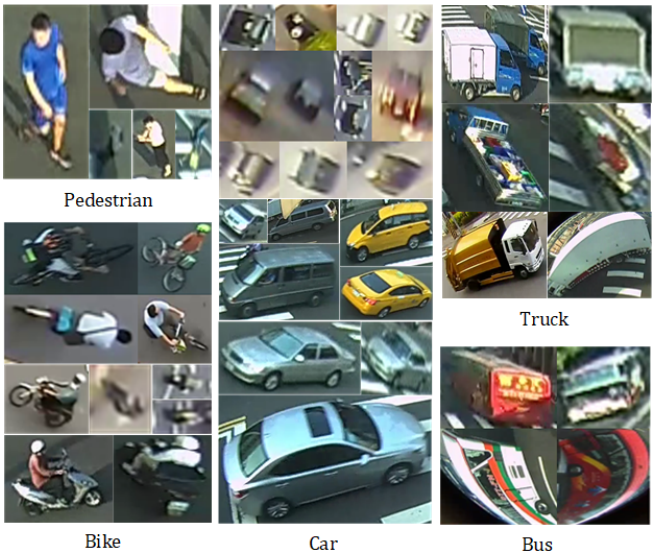

Fig. 1. **Samples of the 5 classes in our Fisheye-DETRAC benchmark dataset: Pedestrian** (people on the streets), **Bike** (people riding bicycles, motorcycles, or scooters), **Car** (light vehicles such as sedans, SUVs, Vans, *etc.*), **Bus**, and **Truck** (dump-truck, semi-trailers, *etc.*) Observe the large FOV and distortions introduced by the fisheye lens, which provides great opportunities and challenges.

classes: Pedestrian, Bike, Car, Bus, and Truck; see Figure 1. The dataset includes 20,000 images, 157,000 bounding boxes, and 313,204 identities captured in 27 videos. These videos were recorded using 22 fisheye cameras deployed for traffic monitoring, with resolutions of 1080×1080, 1920×1920, and 1280×1280. These surveillance cameras, owned by Hsinchu City's police department in Taiwan, provided our data free from licensing or consent agreement issues. We fine-selected 22 short videos, ranging from 8 to 20 minutes, from long hours of footage collected via 35 fisheye cameras. We have also undertaken the necessary precautions to anonymize visible faces and license plates within the video frames.

The Fisheye-DETRAC dataset encompasses diverse traffic scenarios and conditions, including urban highways, intersections, varying light conditions, camera angles, and varying scales of five road object classes. We exercised diligence in labeling objects, including all visible and identifiable objects, irrespective of their distance from the camera.

However, a fisheye camera always causes hemispherical distortions to the flat ground, thus vehicles at different positions are quite distorted. There are two key issues to building an accurate traffic flow estimation system from a fisheye camera. The first issue is to build a robust **vehicle detector** that can detect various vehicles in real time from surveillance videos under various conditions, such as small sizes, occlusions, and perspective distortions. The second component is a robust tracker to track each vehicle for avoiding double counting or missing.

We provide two critical contributions in this work toward a practical use of fisheye cameras for MOT traffic analysis: (1) effective *fisheye object detection and re-identification* that can run directly on the highly distorted fisheye views, without the need of view dewarping, (2) effective *fisheye tracker* that can overcome the nonlinear physical modeling in target/tracklet association on the distorted fisheye views. Our method is superior over most existing MOT algorithms on fisheye cameras, as they rely on the constant velocity assumption of Kalman filtering and thus require fisheye dewarping, which is less effective and prune to error.

Notable tracking algorithms encompass the Kalman filter [14], particle filters [15], and SORT [16]. The Kalman filter employs a linear quadratic estimation model to predict targets' positions over time. Particle filters use a set of particles to depict the object's movement's posterior distribution, while SORT merges the Kalman filter and Hungarian algo-

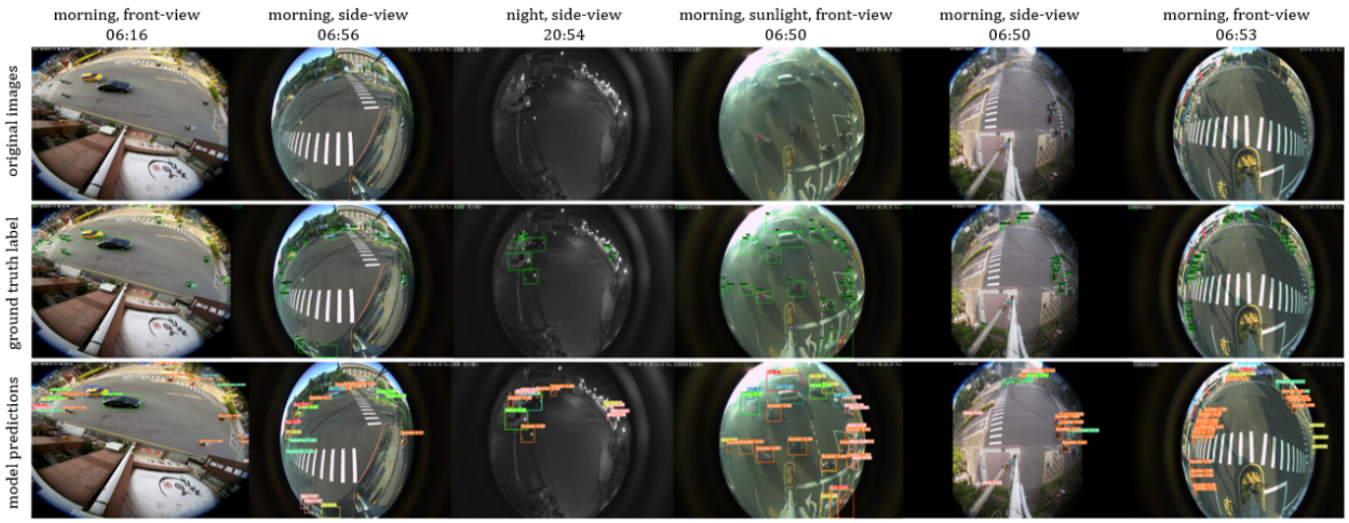

Fig. 2. **Sample images from the Fisheye-DETRAC benchmark:** (Top) the original unlabelled images, (Middle) the labeled ground truths, (Bottom) the YOLOv5x6 [13] detected objects. The columns illustrate several viewing angles, time of day, various intersections and road participants in the dataset.

rithm for real-time multiple object tracking. These State-of-The-Art (SoTA) methods excel at tracking object movement through standard cameras but falter when handling hemispherical distortions in images from fisheye cameras. One solution is to "de-warp" the distorted fisheye image for vehicle detection and tracking, although this increases both the image size and computational demands.

This paper proposes a novel Hybrid Data Association (HDA) method for accurate traffic flow estimation from a fisheye camera. The HDA method can search a vehicle's next position not only from the distorted fisheye image but also perspective one.

We believe that the proposed Fisheye-DETRAC represents a new benchmark for fisheye video analytics. It enables large-scale deployment of fisheye cameras, which takes advantage of the wide-angle fisheye views to improve surveillance, traffic monitoring, and smart city applications.

2. RELATED WORK

2.1. Datasets

Road datasets. High-resolution, diverse, and large-scale road datasets play a critical role in advancing and enhancing traffic monitoring systems. In the last decade, the number of open road datasets [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] for 2D and 3D road object detection, single and multiple object tracking, object segmentation tasks have significantly increased. Table 1 provides a summary of popular road datasets that are used in both model development as well as for benchmarking and public contests. In terms of camera locations, the following datasets are captured using fixed surveillance cameras: MIT-Car [1], UA-DETRAC [3], AAU RainSnow [9], MIO-TCD [10], and AI-City [12] datasets. The CARPK [5] and UAVDT [6] dataasets are captured using drones. The KITTI [2], Detection in LLC [4], NEXET [7], BDD100K [8], and Woodscape [11] datasets are captured using in-dash cameras mounted on a car. In terms of FoV, all the datasets were constructed using standard perspective cameras, with the drawback of narrow FoV. The only exception is the WoodScape dataset [11] that are captured using an in-dash 180° fisheye camera. To our knowledge, the proposed FishEye8K dataset is the first of the kind among the open datasets, that are designed and constructed specifically for the development and evaluation of road object detection using fisheye traffic surveillance cameras.

Fixed perspective traffic camera-based datasets. Table 1 shows that most datasets are captured using fixed, perspective cameras, which are limited by the narrow FoV. All the datasets have annotations for 2D road object detection task; on top of it, a few datasets [6, 12] have multiple objects tracking annotation, and one [9] has segmentation mask annotation. In 2000, MIT-Car dataset [1] was publicly offered as a flagship dataset pioneering the road automation research field. The dataset has 1.1K frames, including 1.1K bounding boxes for the vehicle detection task. In 2016, UA-DETRAC [3] dataset was offered with 140K frames, including rich annotations of illumination, vehicle type, occlusion, and 1210K bounding boxes. The dataset has four classes (car, van, bus, and others) for detection and multiple object detection tasks. In the same year, similarly, MIO-TCD CCTV [10] dataset is offered with 113K frames, including 200K bounding boxes for the detection task. In 2018, the AAU RainSnow [9] dataset was offered as a benchmark for evaluating the SoTA rain removal algorithms. The dataset has 22 five-minute real-world camera video sequences collected from 7 urban intersections covering various weather conditions, i.e., snow, rain, haze, and fog. They have extracted 100 frames from each five-minute video

to construct 2200 frames, including 13297 bounding boxes. Recently, in 2021, AI-City Challenge [12] was held, including vehicle detection and re-identification on CityFlowV2-ReID dataset and multi-target multi-camera vehicle tracking challenge on CityFlow2D dataset. CityFlow2D dataset has 313.9K bounding boxes for 880 distinct vehicles.

Drone based datasets. Lately, drone road datasets have been publicly offered in the literature, namely CARPK [5] and UAVDT [6]. Both datasets were captured from a high altitude with a viewing angle of the top by narrow FOV cameras for the drone-based road monitoring systems. Thus they are not suitable for fixed surveillance camera-based traffic monitoring.

2.2. Algorithm

Object Detection in MOT. Object detection has been a very active field in computer vision since the blooming of deep learning, and it is the basis of multi-object tracking. The extensive amount of literature can be organized into two categories based on their network architectures: *two-stage* proposal-driven [17, 18] and *one-stage* (single-shot) approach [19, 20, 21], [22] improve the tracking performance based on these given detection results. The association ability of these methods can be fairly compared. However, the above methods are unsuitable for multi-class tracking tasks because they are evaluated on a single-class MOT (multiple objects tracking) benchmark.

Tracking by Detection. Tracking by detection approaches form trajectories by correlating a given set of detections over time. RetinaTrack [23] proposes a conceptually efficient and straightforward joint model of detection and tracking, which modifies the famous single-stage RetinaNet [24] approach to be amenable to instance-level embedding training. The FPN [25] series detectors [26] are popularly used for JDE [27, 28] for their excellent balance of accuracy and speed. The CenterNet [29] is anchor-free and becomes the most popular detector cited by most latter methods [28, 30] for its simplicity and efficiency. Most of these methods rely on the detection boxes on a single image for tracking. However, the number of missing and very high hemispherical distortions on bounding boxes begin to increase when the vehicle is close to the edges of video sequence.

Detection and Tracking from Fisheye Cameras. Several research works use a single top-view camera for object detection [31, 32] and object tracking [33]. Thanks to the boom in deep learning, CFPN [34] is the first automatic traffic flow estimation system to detect smaller objects even with significant distortions from fisheye cameras on a real-time embedded system. However, the above SoTA method on Fisheye video did not consider the effects of distortion from the tracking procedure; instead, they focused on detection.

3. FISHEYE-DETRAC BENCHMARK

We provide detailed information on the new detection split of the Fisheye-DETRAC dataset. Figure 2 shows sample images of the wide-angle fisheye views, which provide new opportunities for large coverage, but also new challenges of large distortions of the road objects.

To avoid bias, the train, val, and test sets do not share frames from the same camera. Annotations are provided in several standard formats, including Pascal-VOC[35], MS COCO [36], MOT [37], and YOLO [38].

3.1. Video Acquisition

We have acquired a total of 35 fisheye videos captured using 20 traffic surveillance cameras at 60 FPS in Hsinchu City, Taiwan. Among them, the first set of 30 videos (**Set 1**) was recorded by the cameras mounted at Nanching Hwy Road on July 17, 2018, with 1920×1080 resolution, and each video lasts about 50-60 minutes. The second set of 5 videos (**Set 2**) was recorded at 1920×1920 resolution, and each video lasts about 20 minutes.

All cameras are the property of the local police department, so there is no issue of user consent or license issues. All images in the dataset will be made available to the public for academic and R&D use.

3.2. Dataset Preparation and Characteristics

Sampling. We chose 18 videos from the recorded footage, with 15 videos coming from Set 1. These were cropped into shorter videos, each lasting approximately 8 to 10 minutes, except for one that lasted 16 minutes. Using a sampling method of one frame per 50 and 200 frames for Set 1 and Set 2 videos, respectively, we extracted over 20,000 frames. The resulting images were then resized to 1080×1080 and 1280×1280 for Set 1 and Set 2, respectively.

To incorporate a wide range of perspectives on road conditions, we carefully selected videos for our dataset that feature diverse camera angles, including side-view and front-view shots, as well as varying video quality. The dataset also includes images from different intersection types, such as T-junctions, Y-junctions, cross-intersections, midblocks, pedestrian crossings, and non-conventional intersections. The videos were captured under various lighting conditions, including morning, afternoon, evening, and night, and diverse traffic congestion levels ranging from free-flowing to steady and busy. Figure 2 illustrates some of the wide-ranging road conditions with ground truth annotations of road objects and detection results obtained from YOLOv5x6 [13].

Object classes: We annotate 5 major classes for road objects, namely, **Pedestrian** (all visible people on the streets), **Bike** (riders on bicycles, motorcycles, or scooters), **Car** (light vehicles such as sedans, SUVs, vans, *etc.*), **Bus**, and **Truck** (dump-truck, semi-trailers, *etc.*).

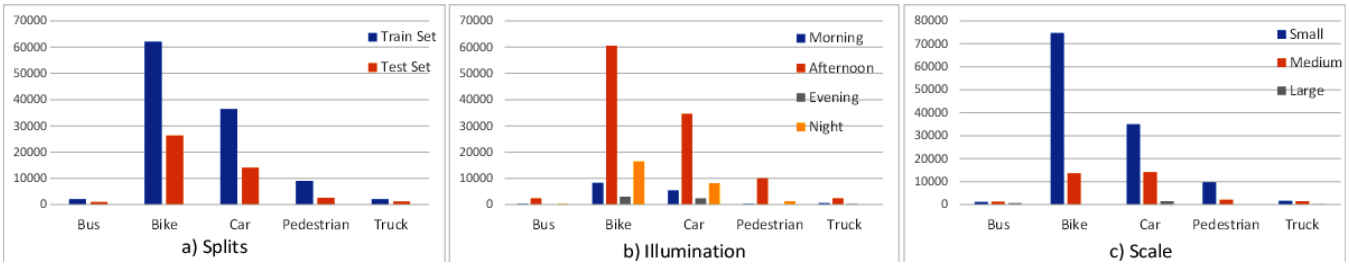

Fig. 3. **The object class distributions in the detection split of the Fisheye-DETRAC dataset**, categorized according to (a) splits, (b) illumination, and (c) scale.

Distant objects: The wide fisheye lens creates a wide FoV but also results in a panoramic hemispherical image that is notably distorted with a barrel effect. Additionally, the camera has a tendency to produce blurred images of objects located around the edges of the lens. As a consequence, distant objects can appear minuscule and indistinct. Annotating these distant objects can be an arduous or even impossible task due to their lack of clarity.

Illumination: Four categories of illumination conditions were identified, namely morning (sunrise), afternoon (sunny), evening (sunset), and night. The distribution of video sequences based on their respective illumination attributes is illustrated in Figure 3(b), with the majority of bounding boxes falling under the afternoon category. Night-time sequences follow in second place, with morning and evening categories trailing behind respectively. Notably, the distribution of classes across all times of day is remarkably similar

Object scale: We define the scale of the bounding boxes of road participants based on their size (length and width) in pixels. The MS COCO evaluator is employed for small and medium, and large scaled objects. However, as the size of the image grows toward 1080×1080 or 1280×1280, respectively for Sets 1 and 2, we doubled the size of standard scales, i.e., *small* (pixels $\leq 64 \times 64$), *medium* ($64 \times 64 <$ pixels $\leq 192 \times 192$), and *large* (pixels $> 192 \times 192$). The distribution of road participants in the dataset in terms of scale is presented in Figure 3 (c), where small and medium-scaled objects make the most of the dataset. Bus and Truck classes have a similar number of small and medium scaled objects. On the contrary, other classes have a comparatively high number of small-scaled objects than medium and large-scale objects.

3.3. Annotation

Annotation rules. The road participants were annotated based on their clarity and recognizability to the annotators, regardless of their location. In some cases, distant objects were also annotated based on this criterion.

Notably, the night video captured by Camera 3 has the highest number of objects. In this dataset, the dominant classes are Bike (88,373) and Car (50,597), which can be attributed to the semi-tropical location of the country where the videos were recorded. On the other hand, the classes of Truck (3,317) and Bus (2,982) have the lowest number of objects, rendering the dataset highly imbalanced. Figure 1 displays a selection of samples from all classes, showcasing various scales. Furthermore, the distributions of classes are depicted as bar graphs in Figure 3.

3.4. Validation

Given the complexity and effort required for the labeling task, human errors were inevitable, and it was necessary to correct them to avoid inaccurate results. Therefore, in order to minimize human error, we employed two semi-automatic approaches to validate all bounding boxes.

In the case of mislabeled objects, we followed a two-step approach. Firstly, we cropped and copied the objects based on their respective bounding boxes into the corresponding directories. Secondly, our annotators manually verified if the objects were correctly placed in their designated directories through simple inspection, which is highly accurate and requires less time and effort. However, this approach is blind to objects that were not labeled in the first place, which is known as a missing label error. To address this issue, we inspected the False Positives generated by the YOLOv7 model [39] trained on FishEye8K, which helped identify numerous missing label errors. This approach was especially effective in identifying errors in distant areas and regions with high traffic density of vehicles and bikes.

3.5. Data Anonymization

The identification of road participants such as people's faces and vehicle license plates from the dataset images was found to be unfeasible due for various reasons. The cameras used for capturing the images were installed at a higher ground level, making it difficult to capture clear facial features or license plates, especially when they are far away. Additionally, the pedestrians are not looking at the cameras, and license plates appear too small when viewed from a distance. However, to maintain ethical compliance and protect the privacy of the road participants, we blurred the areas of the images containing the faces of pedestrians and the license plates of vehicles, whenever they were visible.

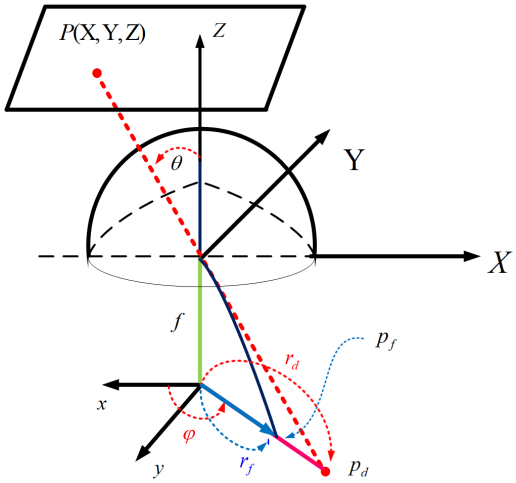

Fig. 4. **The fisheye camera model.**

4. FISHEYE CAMERA MODEL

Refer to Fig. 4. Denote the projection of a 3D point $P = (X, Y, Z)^t$ on the 2D undistorted perspective image as $p_d = (x_d, y_d)^t$ and such a 2D projection point on the fisheye image as $p_f = (x_f, y_f)^t$, respectively. Denote the angle between the light ray and the Z-axis as θ, focal length as f, and the distance between p_d and the Z-axis as r_d. The perspective relation between r_d and θ is $r_d = f \cdot \tan(\theta)$. For small θ, we assume that the length r_f between p_f and the Z-axis is approximately $f \cdot \theta$. Denote the center and radius of a circle on the fisheye image as (c_x, c_y) and R, respectively. Then, r_f is calculated as $r_f = \sqrt{(x_f - c_x)^2 + (y_f - c_y)^2}$. Next, r_d is calculated as: $r_d = R \cdot \tan\left(\frac{r_f}{R}\right)$, where $f = R$. Denote the angle between p_f and the x axis as φ. We have:

$$\varphi = \arctan\left(\frac{y_f - c_y}{x_f - c_x}\right). \tag{1}$$

Next, x_d and y_d can be obtained by $x_d = c_x + r_d \cdot \cos\varphi$ and $y_d = c_y + r_d \cdot \sin\varphi$. Finally, x_d and y_d is calculated as:

$$\begin{aligned} x_d &= c_x + R \cdot \tan\left(\frac{r_f}{f}\right) \cos\left(\arctan\left(\frac{y_f - c_y}{x_f - c_x}\right)\right), \\ y_d &= c_y + R \cdot \tan\left(\frac{r_f}{f}\right) \sin\left(\arctan\left(\frac{y_f - c_y}{x_f - c_x}\right)\right). \end{aligned} \tag{2}$$

5. VEHICLE TRACKING

In fisheye images, straight lines from the original perspective become curved. We propose to first detect these curved trajectories and then self-calibrate the fisheye camera to determine its parameters. This enables efficient correction of vehicle positions with significant hemispherical distortions, without the need to de-warping the original image.

For real-time applications, we have adapted the StrongSORT algorithm [16] to track vehicles using a fisheye camera. As detailed in Sec.5.1, our proposed Hybrid Data Association (HDA) predicts vehicle movements and calculates the Twin Intersection over Union (Twin-IoU) similarity scores as outlined in Sec.5.2. We found that the use of fisheye and distorted perspective images together can improve vehicle tracking.

5.1. Hybrid Data Association (HDA)

We propose the Hybrid Data Association (HDA) to enable effective learning for performing accurate and robust target tracking directly from the distorted fisheye views. Let B_f denote the predicted box for a target (vehicle) V with a fish-eye camera. Let (x_f^L, y_f^T) and (x_f^R, y_f^B) denote the positions of the upper-left and bottom-right box of V, B_f=$(x_f^L, y_f^T, x_f^R, y_f^B)$. B_f can be converted to its new position B_d on the distorted perspective image using Eq.(2). In HDA, a hybrid bounded box B is created to represent V both on the fisheye image and the distorted perspective image, $B = (B_f, B_d)$. The movement state S of target V is modeled as: Explain what the symbols with dot mean.

$$S = (B_f, B_d, \dot{B}_f, \dot{B}_d). \tag{3}$$

Kalman filtering (KF) [14] is then adopted on S to solve the trajectory prediction problem for target tracking directly on fisheye views. This way, the inter-frame displacement of each target can be effectively predicted via KF. In the case when there is no detection to associate with a target, its positions on the fish eye image and the distorted perspective one are simply predicted using linear velocity terms $\dot{B}_f$ and $\dot{B}_d$.

5.2. Twin Intersection over Union

We introduce the concept of Twin Intersection over Union (Twin-IoU), which accounts for the bounding box B, encompassing both B_f from the fisheye image and B_d from the distorted perspective image.

Let's consider a vehicle V_{t-1} at the $(t-1)^{th}$ frame with corresponding bounding boxes denoted as $\overline{B}$. Subsequently, the vehicle detected in the t^{th} frame, represented as V_t, is assigned a predicted bounding box B through our earlier work, the PRB-Net detector [20]. The similarity between V_{t-1} and V_t is measured by their Twin-IoU:

$$TwinIoU(V_{t-1}, V_t) = \frac{|\overline{B}_f \cap B_f|}{|\overline{B}_f \cup B_f|} + \frac{|\overline{B}_d \cap B_d|}{|\overline{B}_d \cup B_d|}, \tag{4}$$

Different from the IoU score used in the SORT algorithm [16], Eq.(4) considers the IoU score not only from the fisheye camera but also the perspective image. Then, the next position of V^{t-1} at the t^{th} frame is tracked by solving the following equation:

$$V_t = \underset{V_t^i}{\arg\max}\ TwinIoU(V_{t-1}, V_t^i). \tag{5}$$

If $IoU(V_{t-1}, V_t) <$ a threshold, the its position of V_{t-1}^i at the t^{th} frame is simply predicted with $\overline{B}_f^{Vt-1}$.

Table 2. **Evaluation of SoTA detection models** trained on the Fisheye-DETRAC benchmark. The table consists of two groups of various versions of YOLO object detection models for input sizes of 1280×1280 and 640×640.

Model	Version	Input Size	*Precision*	*Recall*	$mAP_{0.5}$	$mAP_{.5:.95}$	*F1-score*	AP_S	AP_M	AP_L	Inference [ms]
YOLOv5 [13]	YOLOv5l6	1280×1280	0.7929	0.4076	0.6139	0.4098	0.535	0.1299	0.434	0.6665	22.7
	YOLOv5x6	1280×1280	**0.8224**	0.4313	0.6387	0.4268	0.5588	0.133	0.452	0.6925	43.9
YOLOR [40]	YOLOR-W6	1280×1280	0.7871	0.4718	0.6466	**0.4442**	0.5899	0.1325	0.4707	0.6901	16.4
	YOLOR-P6	1280×1280	0.8019	0.4937	**0.6632**	0.4406	0.6111	0.1419	0.4805	**0.7216**	**13.4**
YOLOv7 [39]	YOLOv7-D6	1280×1280	0.7803	0.4111	0.3977	0.2633	0.5197	0.1261	0.4462	0.6777	26.4
	YOLOv7-E6E	1280×1280	0.8005	**0.5252**	0.5081	0.3265	**0.6294**	**0.1684**	**0.5019**	0.6927	29.8
YOLOv7 [39]	YOLOv7	640×640	0.7917	0.4373	0.4235	0.2473	0.5453	0.1108	0.4438	0.6804	**4.3**
	YOLOv7-X	640×640	0.7402	**0.4888**	0.4674	0.2919	**0.5794**	**0.1332**	**0.4605**	**0.7212**	6.7
YOLOv8	YOLOv8l	640×640	0.7835	0.3877	0.612	0.4012	0.5187	0.1038	0.4043	0.6577	8.5
	YOLOv8x	640×640	**0.8418**	0.3665	**0.6146**	**0.4029**	0.5106	0.0997	0.4147	0.7083	13.4

Table 3. **Evaluation of SoTA MOT models** trained on the Fisheye-DETRAC benchmark.

Method	HOTA↑	IDF↑	MOTA↑	AssA↑	DetA↑	IDs↓
SORT [16]	22.1	24.1	27.9	20.1	23.3	48,201
FairMOT [28]	37.2	45.8	46.2	32.7	38.7	32,597
ByteTrack [41]	40.8	49.2	50.4	39.6	40.1	25,691
DeepSORT [42]	38.1	47.5	48.8	37.9	40.0	26,984
StrongSORT [43]	40.3	49.8	49.8	40.5	40.3	25,999
BoT-SORT-R [44]	41.2	52.1	50.0	41.2	41.5	19,566

* Higher Order Tracking Accuracy (HOTA), ID F1 (IDF), Multiple Object Tracking Accuracy (MOTA), Association Accuracy (AssA), Detection Accuracy (DetA), ID switch (IDs).

6. BENCHMARK RESULTS

We evaluate the object detection performance of several YOLO models on Fisheye-DETRAC using a workstation with an 11^{th} Gen i7 CPU and an Nvidia RTX 3080 GPU.

6.1. Hyperparameter Settings and Evaluation Metrics

We utilized several frameworks and platforms, *i.e.*, Darknet [45], PyTorch [46], and PaddlePaddle [47] for the model training. platforms for detector and tracker

Hyperparameters. All YOLO variations were pre-trained on MS COCO [36] dataset. Among the models, we trained four models, namely YOLOv7 [39], YOLOv7-X [39], YOLOv8l, and YOLOv8x on input image size of 640×640. The rest six models, namely YOLOv5x6 [13], YOLOv5l6 [13], YOLOR-W6 [40], YOLOR-P6 [40], YOLOv7-D6 [39], YOLOv7-E6E [39], are trained with size 1280×1280. All models were trained with the same procedures for 250 epochs. Adam [48] optimizer were used with momentum of 0.937 except for YOLOv5, where the SGD optimizer was employed. The confidence and the IoU threshold for Non Max Suppression (NMS) were both set to 0.5; the learning rate is 0.01.

We use a confidence threshold 0.3 to determine the detection reliability balancing between false positives and negatives in performing tracking. We use a confidence threshold 0.4 for initializing new tracks, and we use a track buffer size 30 to determine lost tracks. These parameters we selected to handle occlusions properly. We use a matching threshold 0.7 to manage detection-track associations for controlling tracking accuracy. We use the aspect ratio threshold 1.6 and minimum box area of 10 pixels to consider only suitable detections. If enabled, the score and IoU fusion feature combines detection score and IoU to improve tracking.

Metrics. The evaluation metric employed for object detection tasks is the mean Average Precision (mAP), as defined in PASCAL VOC 2012 [35]. To calculate mAP, the Average Precision (AP) values for each class are averaged. AP for a specific class is derived from the Precision-Recall curve, which is generated by varying the detection confidence threshold. Precision (P) and recall (R) are defined as $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$, respectively, where True Positive (TP) represents the number of correctly detected objects of the class, False Positive (FP) denotes the number of incorrect detections, and False Negative (FN) indicates the number of undetected objects of the class. The AP is computed by calculating the area under the Precision-Recall curve using either the 11-point interpolation method or the integration of the interpolated curve. The final mAP score represents the mean AP across all object classes, providing an overall assessment of the object detection model's performance.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{6}$$

where N is the number of object classes, and AP_i is the average precision for the i^{th} class.

MOT Metrics [37]. We use the MOTA [37], IDF1 [37], and HOTA [37] to evaluate the MOT performance. These metrics reflect how well multiple object tracking is preformed and penalize identity switches.

6.2. Fisheye-DETRAC Benchmark Results

Object Detection. We quantitatively evaluate the Fisheye-DETRAC for the popular YOLO family of object detectors, namely, YOLOv5 [13], YOLOR [40], YOLOv7 [39], and the latest YOLOv8. Table 2 shows the outcome in terms of

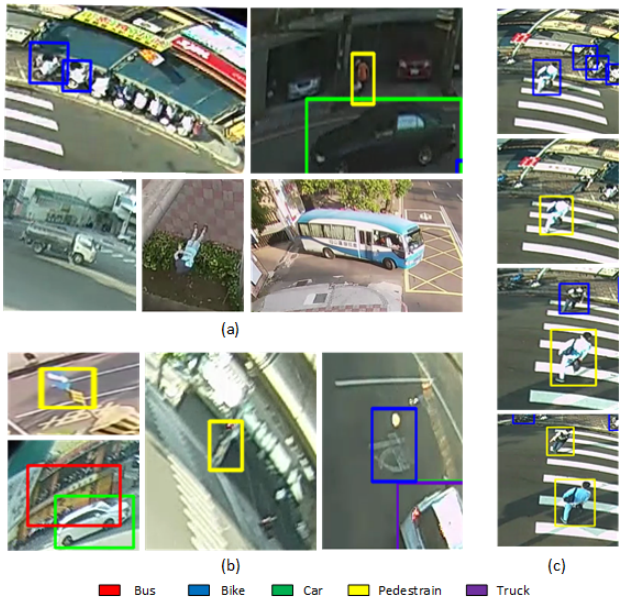

Fig. 5. **Samples of hard cases in Fisheye-DETRAC** for YOLOR-W6 detections on the input size of 1280 × 1280. (a) **False Negatives** (FN): instances where the labeled objects are not detected. These typically involve parked vehicles or moving road participants. (b) **False Positives** (FP): cases where the background is erroneously identified as an object class. (c) Detected objects that are misidentified as other classes, which frequently occur at road signs, buildings, and objects far away. For example, Pedestrians far from the camera could be incorrectly classified as Bikes.

precision-recall, mAP, and inference time. Results demonstrate that all models perform efficiently with only a few ms of inference time. Figure 5 presents challenging examples for the top-performing YOLOR-W6 [40] model.

MOT. We quantitatively evaluate the Fisheye-DETRAC for six SoTA trackers [16, 42, 41, 43, 44]. Table 3 shows the results using standard MOT evaluation metrics. The BoT-SORT-R [44] performs the best on the Fisheye-DETRAC benchmark.

6.3. Ablation study of HDA

Table 4 the results of an ablation study comparing StrongSORT [43] and BoT-SORT-R [44] with and without HDA. The incorporation of HDA significantly enhances performance, yielding superior scores across both methods compared to their counterparts without HDA.

7. CONCLUSION

We introduce the Fisheye-DETRAC benchmark dataset. We believe this benchmark dataset can filled a noticeable gap in fisheye camera surveillance applications regarding road object detection and multi-object tracking tasks. This anonymized dataset comprises 20,000 frames, 157K bounding boxes, and 313K identities spanning 5 different road participants, capturing a diverse range of road conditions. We also produce a new Hybrid Data Association (HDA) method as another contribution. The HDA can effectively improve vehicle tracking and velocity estimation directly on fisheye cameras, without the need to unwarp the underlie hemispherical distortions. Unlike existing state-of-the-art methods that primarily focus on detection, our HDA approach considers distortion effects while performing tracking and vehicle movement prediction. The proposed Twin-IOU can calculate the fisheye similarity scores, we found that the use of fisheye and distorted perspective images together can improve vehicle tracking. We expect the Fisheye-DETRAC benchmark will continue to impact future researches on fisheye video analytics and smart city applications.

Table 4. **Ablation study of HDA** on the Fisheye-DETRAC for two SORT based MOT methods [43, 44].

Method	HDA	HOTA
StrongSORT [43] as baseline		40.3
Ours	✓	**44.1**
BoT-SORT-R [44] as baseline		41.2
Ours	✓	**45.6**

8. ACKNOWLEDGEMENT

We thank to SciDM and National Center for High-performance Computing (NCHC) for providing computational and storage resources. This research is supported in part by the National Center for High-performance Computing (NCHC), SCIDM of NCHC, the Massachusetts Institute of Technology (MIT), United Arab Emirates University (UAEU), Hsinchu City Police Bureau, Hsinchu City Government, and the Central Taiwan Science Park Bureau (NSTC).

9. REFERENCES

[1] Constantine Papageorgiou and Tomaso Poggio, "A trainable system for object detection," *IJCV*, vol. 38, no. 1, pp. 15–33, 2000. 2, 3

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361. 2, 3

[3] L. Wen et al., "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Comput Vis Image Underst*, vol. 28, 2020. 2, 3

[4] Roman Kvyetnyy et al., "Object detection in images with low light condition," in *Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2017*, 2017, vol. 10445, p. 104450W. 2, 3

[5] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *ICCV*, 2017, pp. 4145–4153. 2, 3, 4

[6] Dawei Du et al., "The unmanned aerial vehicle benchmark: Object detection and tracking," in *ECCV*, 2018, pp. 370–386. 2, 3, 4

[7] Itay Klein, Nexar Blog, "NEXET - the largest and most diverse road dataset in the world," 2017. 2, 3

[8] Huazhe Xu et al., "End-to-end learning of driving models from large-scale video datasets," in *CVPR*, 2017, pp. 2174–2182. 2, 3

[9] Chris H Bahnsen and Thomas B Moeslund, "Rain removal in traffic surveillance: Does it matter?," *T-ITS*, vol. 20, no. 8, pp. 2802–2819, 2018. 2, 3

[10] Zhiming Luo et al., "MIO-TCD: A new benchmark dataset for vehicle classification and localization," *TIP*, vol. 27, no. 10, pp. 5129–5141, 2018. 2, 3

[11] Senthil Yogamani et al., "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *ICCV*, 2019, pp. 9308–9318. 1, 2, 3

[12] Milind Naphade, Shuo Wang, David C Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Yue Yao, Liang Zheng, Pranamesh Chakraborty, Christian E Lopez, et al., "The 5th ai city challenge," in *CVPR*, 2021, pp. 4263–4273. 2, 3, 4

[13] Glenn Jocher et al., "ultralytics/yolov5: v7.0 - yolov5 sota realtime instance segmentation," 2022. 3, 4, 7

[14] T. Basar, *A New Approach to Linear Filtering and Prediction Problems*, pp. 167–179, 2001. 2, 6

[15] F. Gustafsson et al., "Particle Filters for Positioning, Navigation, and Tracking," *IEEE Trans. Signal Process*, vol. 50, pp. 425–437, 2002. 2

[16] Alex Bewley et al., "Simple online and realtime tracking," *ICIP*, Sep 2016. 2, 6, 7, 8

[17] S. Ren et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE PAMI*, 2017. 4

[18] B. Singh and L.S. Davis, "An analysis of scale invariance in object detection SNIP," in *CVPR*, 2018. 4

[19] A. Bochkovskiy et al., "YOLOv4: Optimal speed and accuracy of object detection," in *arXiv*, 2020. 4

[20] P.Y. Chen et al., "Parallel residual bi-fusion feature pyramid network for accurate single-shot object detection," *IEEE TIP*, vol. 30, pp. 9099–9111, 2021. 4, 6

[21] M. Tan et al., "EfficientDet: Scalable and efficient object detection," in *CVPR*, 2020. 4

[22] J. Xu et al., "Spatial-temporal relation networks for multi-object tracking," in *ICCV*, 2019. 4

[23] Z. Lu et al., "Retinatrack: Online single stage joint detection and tracking," in *CVPR*, 2020. 4

[24] Tsung-Yi Lin et al., "Focal loss for dense object detection," in *ICCV*, 2017. 4

[25] T.Y. Lin et al., "Feature pyramid networks for object detection," in *CVPR*, 2017. 4

[26] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018. 4

[27] Zhongdao Wang et al., "Towards real-time multi-object tracking," in *ECCV*, 2020. 4

[28] Yifu Zhang et al., "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 3069–3087, 2021. 4, 7

[29] Kaiwen Duan et al., "CenterNet: Keypoint triplets for object detection," in *ICCV*, 2019. 4

[30] Jialian Wu et al., "Track to detect and segment: An online multi-object tracker," in *CVPR*, 2021. 4

[31] Han Wang et al., "Top-view people detection based on multiple subarea pose models for smart home system," in *ICCE*, 2016. 4

[32] Xue YUAN et al., "Pedestrian detection for counting applications using a top-view camera," *IEICE Trans Inf Syst*, vol. E94.D, no. 6, pp. 1269–1277, 2011. 4

[33] Mamoru Saito et al., "People detection and tracking from fish-eye image based on probabilistic appearance model," in *SICE Annual Conference*, 2011, pp. 435–440. 4

[34] P.Y. Chen et al., "Smaller object detection for real-time embedded traffic flow estimation using fish-eye cameras," in *ICIP*, 2019. 4

[35] M. Everingham et al., "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," . 4, 7

[36] Tsung-Yi Lin et al., "Microsoft COCO: Common objects in context," in *Computer Vision – ECCV 2014*, 2014. 4, 7

[37] Keni Bernardin and Rainer Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, 2018. 4, 7

[38] Joseph Redmon et al., "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. 4

[39] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022. 5, 7

[40] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao, "You only learn one representation: Unified network for multiple tasks," *CoRR*, vol. abs/2105.04206, 2021. 7, 8

[41] Yifu Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in *ECCV*, 2022. 7, 8

[42] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *ICIP*, 2017, pp. 3645–3649. 7, 8

[43] Yunhao Du et al., "StrongSORT: Make DeepSORT great again," *arXiv*, 2022. 7, 8

[44] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," *arXiv*, 2022. 7, 8

[45] Joseph Redmon, "Darknet: Open source neural networks in C," 2013–2016. 7

[46] Adam Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. 7

[47] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang, "PaddlePaddle: An open-source deep learning platform from industrial practice," *Frontiers of Data and Domputing*, vol. 1, no. 1, pp. 105, 2019. 7

[48] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2014. 7