

Pedestrian Detection in Top-view Fisheye Images for Drone Applications

Yassir Zardoua ^{*}, Joelle Jreis, and Joanna Faddoul 

Caplogy Innovation, Vélizy-Villacoublay, France

Email: y.zardoua@caplogy.com (Y.Z.); j.jreis@caplogy.com (J.J.); j.faddoul@caplogy.com (J.F.)

^{*}Corresponding author

Abstract—Fisheye cameras offer images covering a large Field of View (FOV), typically a hemisphere for a single camera, with the trade-off of significant radial distortions and object rotations. The literature indicates that a Frames-Per-Second (FPS) friendly approach creates multiple gnomonic projections, with the aim of covering the complete hemisphere, i.e., fisheye image. These projections are naturally similar to pinhole perspective cameras, thus allowing restoration of the correct object orientation and reduction of the distortions per-projection. Hence, we can make use of the abundant resources related to perspective cameras for fisheye imagery. However, the choice of projection parameters—number of views, their arrangement, field of view, longitude and latitude of the views—has received little systematic study. We show that a hypothesis-driven optimization of these parameters yields moderate to substantial Average Precision (AP) gains over the existing baseline, with AP improvements reaching up to $\sim +10\%$ across three public benchmarks. Additionally, the existing method backprojecting bounding boxes from the projected views to the original fisheye is altitude-dependent. To suit the drone imagery case, we propose an altitude-agnostic method, and use it for all projection settings we tested. To further confirm our optimized configuration, we conducted extensive tests isolating the effect of projection parameters, You Only Look Once (YOLO) backbones, and input resolutions.

Keywords—fisheye camera, gnomonic projection, object detection, Field of View (FOV), drone imagery

I. INTRODUCTION

Autonomous systems, particularly Unmanned Aerial Vehicles (UAVs), require full environmental perception in order to function safely and accomplish tasks. Fisheye cameras are appropriate for obstacle detection and surveillance applications in drones because they provide medium range FOV (Field-of-View) coverage of 180° from a single sensor [1]. Their wide coverage makes them a natural choice for crowd monitoring, indoor security surveillance [2], and robotics navigation, where a conventional narrow-FOV camera would require multiple units or active pan-tilt mechanisms to achieve equivalent scene coverage. Accurate person detection in these settings is the central practical problem this work addresses.

However, common object detection algorithms, including sophisticated methods like You Only Look Once (YOLO) [3], are severely challenged by the severe radial distortion caused by fisheye lenses.

Conventional techniques for fisheye image analysis can be divided into two groups [4, 5]: (a) pre/post-processing techniques, which depend on image transformations prior to detection and bounding box refinement following, and (b) architecture adaptation techniques, which alter and retrain neural networks to process the distorted inputs directly. For instance, fisheye images are projected as perspective view images, standard detectors are applied to the transformed image, and bounding box suppression algorithms (e.g., SoftNMS [6]) are used to back-project and filter the detections. Although extensive retraining with specialized datasets and modified architectures is necessary, architecture adaptation approaches can theoretically achieve optimal performance. By utilizing pre-trained networks, pre/post-processing techniques provide more flexibility and control over angular coverage and computational complexity.

Among pre/post-processing approaches, the multi-view gnomonic projection framework of Chiang *et al.* [5] showed that a standard detector can process the entire fisheye field in a single forward pass, achieving competitive accuracy at real-time speed, by stacking multiple perspective sub-views into a single composite image. Nevertheless, rather than being methodically optimized, the projection parameters used in that work—eight viewports arranged in a 2×4 grid with a narrow field of view—were inherited from previous designs. Although their combined effect has not been examined in the literature, the number of projections, the per-viewport FOV, the grid layout, and the viewing latitude all interact to determine detection quality.

In this paper, we revisit this design space and demonstrate that, without requiring retraining or architectural changes to the underlying detector, a carefully selected six-projection configuration consistently and significantly improves detection metrics over the eight-projection baseline. Our primary contributions are:

- (1) A 2×3 composite layout with six projections, 60° h-FOV, 85° v-FOV, and 45° latitude outperforms the

existing baseline across three public fisheye pedestrian benchmarks, according to a systematic projection parameter study that evaluated ten configurations spanning 4, 6, 8, and 9 projections with varying FOV and latitude.

- (2) Validation of backbone-agnosticism: detector-specific artifacts are ruled out by confirming the configuration advantage across six YOLO architectures of varying capacity levels (YOLOv8n/m, YOLOv9e, YOLOv10m, YOLO11m, and YOLO12m).
- (3) An altitude-agnostic geometric backprojection that does away with the pre-calibrated altitude-dependent lookup tables used in earlier work [5] by analytically mapping composite-space detections to radially-aligned rotated bounding boxes in fisheye space.
- (4) A resolution ablation quantifies the speed-accuracy trade-off from 640×640 to 320×320 composite resolution, demonstrating that the suggested layout outperforms the baseline on the largest benchmark even at 416×416 while providing a $1.4\times$ throughput increase.

The rest of this paper is organized as follows: the literature review is discussed and analyzed in Section II, with a summary and research motivation being presented in the subsequent Section III. The proposed method, along with its design rationale, is discussed in detail in Section IV. Before reporting experimental results in Section VI, we present the experimental setup in Section V, and we finally conclude in Section VII.

Related work on fisheye object detection is reviewed in Section II, and the literature summary with the research motivation is discussed in Section III. Our approach, which includes the altitude-agnostic backprojection, the multi-view detection fusion strategy, the gnomonic projection pipeline, and the design rationale, is presented in Section IV. The experimental setup is described in Section V, and the results are reported and analyzed in Section VI. The paper is concluded and future directions are outlined in Section VII.

II. LITERATURE REVIEW

The problem of object detection in fisheye images has attracted significant research attention due to the growing deployment of omnidirectional cameras in autonomous systems. Approaches can be broadly categorized into direct adaptation methods that modify neural network architectures, and indirect projection methods that transform images to standard perspective views.

Direct neural network adaptation approaches attempt to handle fisheye distortions within the model architecture. Wei *et al.* [7] deform convolutional kernels to align the receptive field with the radially variant warping nature of fisheye images. They rely on methods such as the Poincaré ball, Möbius operations, hyperbolic GCNs (Graph Convolutional Networks), and geodesic distances to capture multiple deformation criteria, including radial distance from the fisheye center, angular position, and local feature content. The last criterion makes this approach input-dependent, thus requiring

compute-intensive kernel deformation at each forward pass.

Yang *et al.* [3] proposed PGDS-YOLOv8s, an enhanced YOLOv8s baseline, with various architectural adaptations. They reported notable improvements in their ablation study while maintaining an excellent Frames-Per-Second (FPS) (250 vs. 286 for the baseline). However, we believe that improvements made in Ref. [3] are general-purpose rather than fisheye-specific, as all of their adaptations rely on integration of various plug-and-play technologies, namely Ghost modules [8], Squeeze-and-Excitation (SE) blocks [9], and Spatial Pyramid Pooling Fast (SPPF) [10]. We note that Yang *et al.* [3] process 360° equirectangular images rather than fisheye counterparts, which has a different type of distortion.

Unlike Ref. [3], earlier works tackling omnidirectional distortions, such as Ref. [11], criticized the direct usage of axis-aligned Bounding Boxes (BBoxes)—rectangles whose edges remain parallel to the image axes (x and y), as in the BBoxes produced by early YOLO models—on distorted images, crafting instead bending BBoxes that more tightly enclose detected objects. In top-view person detection, Duan *et al.* [12] made this explicit by extending a YOLOv3-style head to regress rotated BBoxes with an additional angle term $\hat{b}_\theta$ and a modified loss handling the periodic nature of rotated BBoxes. Haggui *et al.* [13] and Yuan *et al.* [14] later built on that recipe, with Haggui [13] proposing a lightweight centroid-based tracker to enhance current detections, and Yuan [14] leveraging that recipe through training on DOTA (Dataset for Object deTection in Aerial images) [15], which lacks fisheye images but includes valuable annotations of rotated BBoxes from aerial imagery, thus aiding the regression of rotated BBoxes. Yuan *et al.* [14] also adopted the DyHead (Dynamic Head) block from Dai *et al.* [16], inserted between the backbone and the neck of RTMDet [17]. While DyHead uses different types of attention, i.e., scale, size, and task attention, its principle seems similar to Ref. [3]: a general-purpose architectural enhancement not tailored to the specific characteristics of fisheye images.

Unlike the angle-regression adaptation in [12–14, 16], Yang *et al.* [18] argue that such regression is actually redundant and can even lead to unstable training, as the same object (person) can fit into multiple valid BBoxes with different orientations. Therefore, they introduced radius-aligned BBoxes, which not only standardizes BBox definition but also removes the need of angle regression, thus leading to simpler model and more stable training. Additionally, Yang *et al.* [18] proposed a dual-loss leveraging the rotation equivariance property of top-view fisheye images, which is grounded on the observation that rotation of an image (from the fisheye center) produces an image within the realm of what the model expects at test time, with the possibility of precisely computing GT (Ground Truth) BBoxes corresponding to rotated images. This property doesn't apply when the optical axis is horizontal (or near-horizontal), as we do not expect, for instance, to detect people upside down. Accordingly, the first and second term of the dual-loss correspond to predictions on two images, with one being the rotated

version of the other. Authors experimentally show that this is not a simple data augmentation, but rather a methodology leveraging rotation equivariance to encourage the model into producing rotation invariant features.

A recent arXiv paper [19] converted the fisheye image to an equirectangular version, and then proposed a Panoramic Distortion-Aware Tokenization (PDAT) method, which is adapted from the significance map tokenization proposed in SG-Former [20]. While the latter uses the significance map to increase the tokenization resolution at any feature map location, the PDAT adapts this process only within the latitude range of $\theta > 70^\circ$, i.e., toward the top of the panorama, which is justified by the observation that pedestrians' height linearly decreases after the 70° latitude threshold. Wakai *et al.* [19] leverage this linearity in their PDAT by establishing a non-uniform tiling strategy, thus producing smaller tiles in the uppermost regions to better fit pedestrian height. The tiles below the 70° latitude threshold are kept unchanged since there is no significant height change of BBoxes. While Wakai *et al.* [19] reports promising accuracy metrics, the absence of reported hardware and latency-related tests may hinder its direct integration within embedded drone systems.

Multi-view projection approach is a simpler way to adapt existing resources, namely existing detectors, with minimal labor. Chiang and Wang [21] pioneered an algorithm within this approach, fixing a narrow window (rectangle) while aligning its major axis with an arbitrary radial axis (usually the upward-pointing axis), then applying the ubiquitous Histogram of Oriented Gradients-Support Vector Machine (HOG-SVM) detector of that time. The fisheye image is then iteratively rotated while keeping the window at the same arbitrary position until the entire FOV is covered. Following this principle, Li *et al.* [22] increased the window width as they used a more advanced detector, applying YOLOv3 on a fixed window covering a specific fisheye part while rotating it with 15° increments. They aimed at producing radial-aligned BBoxes to accommodate the labels in their newly proposed HABBOF, thus projecting the detected BBoxes back onto the original fisheye image, and eventually forcing all BBox orientations to align with fisheye radius. An obvious limitation in Ref. [22] is the necessity of rotating the fisheye image 24 times, thus requiring YOLOv3 to process 24 windows per frame. The authors mitigate this limitation through an activity-aware mechanism, skipping empty windows and processing pedestrian regions only. However, the computational issue will persist in crowded scenes or in images where few pedestrians are scattered at a wide variety of angles, and the proposed activity-aware mechanism is effective in static indoor scenes only.

In parallel, Tamura *et al.* [23] focused on creating a rotation-aware training pipeline, thus avoiding the need of processing multiple windows per frame. Their baseline YOLOv2 keeps the same prediction head, while computing the BBox angle as a post-detection step through mathematical alignment with the radial axis. They

replaced Non-Maximum Suppression (NMS) of original YOLOv2 with their Bounding Box Refinement (BBR): rotated pedestrians lead to multiple BBoxes with different sizes and low Intersection over Union (IoU) of a given pedestrian, which NMS fails to merge. BBR avoids IoU and relies instead on clustering centers with mean-shift, eventually computing the final center and size (width and height) via a confidence-weighted average. This process may be delicate in crowded scenes with nearby centroids, potentially producing many False Negatives (FN). Nevertheless, BBR is shown experimentally to be more effective than the original NMS of YOLOv2.

While Tamura *et al.* [23] avoided the use of multiple windows for computational considerations, Chiang *et al.* [5] later showed the possibility of using 8 windows without affecting the baseline speed while achieving competitive or improved performance relative to Tamura *et al.* [23] on several benchmarks. The fixed window and fisheye rotation adopted by Tamura, although aligns BBoxes, crops the fisheye with its natural radial distortions, whereas Chiang *et al.* [5] produce the 8 windows through gnomonic projections with a fixed horizontal and vertical FOV, re-arranging them eventually into one single image, referred to as composite image, by stacking them into two rows and four columns, thus creating a 2×4 composite image input to the model. Back-projection of detected BBoxes on gnomonic views back onto fisheye doesn't produce rectangular BBoxes as in prior works, which is why a novel regression-based BBox mapping is introduced. This regression is conceptually similar to Ref. [19], where the linear decrease of persons' height on equirectangular panorama above 70° latitude is exploited; in contrast, Chiang *et al.* [5] pre-computed a large set of target BBoxes of different sizes that decrease systematically with the radial distance from the fisheye center, eventually selecting a fixed set of target BBoxes for each reference box detected on the composite image. The target BBoxes are selected based on IoU, and the final fisheye BBox is computed using an IoU-based weighting process.

III. LITERATURE SUMMARY AND RESEARCH MOTIVATION

In summary, the works we reviewed herein adopt various approaches and strategies, including architectural modifications accounting for fisheye or equirectangular distortions [7, 19] (e.g., modified tokenizers and convolutional kernels), extension of perspective detectors to regress rotated BBoxes with rotation-aware loss [12–14], seamless merging of various plug-and-play modules into YOLO variants [3], alignment of BBoxes with the fisheye radial axis [22, 23], rotation-aware training pipelines [18, 23], and the processing of sub-windows extracted through cropping or gnomonic projection [5, 21]. Table I summarizes the key characteristics of reviewed approaches alongside our method: all methods in that table either require fisheye-specific retraining, multiple forward passes per frame, or custom architectures. Our approach builds on the single-pass composite framework of Chiang *et al.* [5], adding systematic projection optimization and

detector-agnostic design while reducing the number of views from 8 to 6. Furthermore, the gnomonic back-projection geometry yields radially-aligned bounding boxes naturally and unambiguously, without any additional annotation convention or custom loss.

If we consider the detection robustness and computational constraints as key design factors for drone-based systems, we can safely conclude that a single

forward-pass of a single image stacking N projections, as adopted in Ref. [5], is a promising approach. Therefore, our research aims at performing further study within this framework. Specifically, we noticed that the literature lacks a study to optimize the projection configuration, and we ground our study on the hypothesis described in Section IV-A.

TABLE I. COMPARATIVE OVERVIEW OF FISHEYE PEDESTRIAN DETECTION APPROACHES. PASSES/FRAME: NUMBER OF DETECTOR FORWARD PASSES REQUIRED PER INPUT FRAME

Category	Method	Description	Retrain	Passes/frame	Evaluation
Direct architecture adaptation	Wei <i>et al.</i> [7]	Input-dependent deformable convolutions on the Poincaré ball; models radial and angular distortion within the network.	✓	1	Captures multiple deformation criteria directly; however, per-frame kernel recomputation adds latency and the architecture is tied to a specific backbone.
	Yang <i>et al.</i> [3]	Ghost modules, SE blocks, and SPPF integrated into YOLOv8s; applied to equirectangular images.	✓	1	Fast inference (250 FPS) with notable ablation gains; however, enhancements are general-purpose and applied to equirectangular rather than fisheye images.
	Wakai <i>et al.</i> [19]	Fisheye-to-equirectangular conversion with distortion-aware tokenisation (PDAT) above 70° latitude.	✓	1	Exploits the linear height decrease above 70° with non-uniform tiling; no reported latency or hardware analysis for embedded deployment.
Oriented / rotated BBox regression	Duan <i>et al.</i> [12]	YOLOv3-style head extended with angle regression b_θ and rotation-aware loss for top-view detection.	✓	1	Handles arbitrary orientations; angle regression can lead to ambiguous or unstable training [18].
	Hagui <i>et al.</i> [13]	Builds on [12] with a lightweight centroid-based tracker to temporally refine detections.	✓	1	Temporal refinement improves consistency; inherits angle-regression instability and assumes temporal continuity.
	Yuan <i>et al.</i> [14]	RTMDet with DyHead attention; pre-trained on DOTA aerial imagery for rotated BBox regression.	✓	1	DOTA pre-training provides rich rotated BBox supervision; DyHead adds general-purpose overhead and DOTA lacks fisheye imagery.
	Yang <i>et al.</i> [18]	Radius-aligned BBoxes eliminate angle regression; dual-loss exploits rotation equivariance of top-view fisheye.	✓	1	Simpler and more stable than free-angle regression; custom architecture and loss, rotation equivariance limited to top-view cameras.
Multi-view geometrical projection	Chiang <i>et al.</i> [21]	Fixed narrow rectangular crop; fisheye image rotated incrementally to sweep the full scene; HOG-SVM applied at each step.	✓	N	Pioneering multi-view approach; sequential processing scales linearly with the number of rotation steps.
	Li <i>et al.</i> [22]	24 gnomonic crops processed individually by YOLOv3; activity-aware mechanism skips empty views.	✓	≤ 24	Skip mechanism reduces cost in sparse static scenes; up to 24 forward passes in crowded or scattered scenarios.
	Tamura <i>et al.</i> [23]	Rotation-aware training on the full fisheye image; mean-shift BBR replaces NMS.	✓	1	Single-pass; BBR shown more effective than standard NMS, but mean-shift clustering fragile with nearby centroids in crowded scenes.
	Chiang <i>et al.</i> [5]	8 gnomonic views stacked into a 2×4 composite; single forward pass with off-the-shelf YOLOv3 weights.	-	1	Single-pass, retraining-free, and in principle detector-agnostic; projection layout manually chosen with no exploration of alternative detectors or configurations.
	Ours	6 optimised gnomonic views in a single composite; projection parameters selected via systematic search on detection metrics. Compatible with any YOLO variant without retraining.	-	1	Fewer projections (6 vs. 8) with optimised configuration; inherits the BBox back-projection complexity of gnomonic composite approaches [5].

IV. MATERIALS AND METHODS

Our methodology builds upon the equidistant (fisheye) projection model, which maps 3D directions to 2D image coordinates through the relationship:

$$r = f \cdot \theta \quad (1)$$

where r is the radial distance from the image center, f is the focal length, and θ is the incident angle of the 3D direction from the optical axis. This model underpins the geometric

transformations between fisheye and perspective coordinate systems described in the current section.

A. Design Rationale

A critical challenge in multi-projection fisheye processing is selecting appropriate projection parameters: the number of projections N , the horizontal and vertical field-of-view angles, and the viewing direction arrangement (longitudes $\{\lambda_i\}$ and latitude φ). Our design is grounded in the observation that a larger FOV per projection covers the fisheye image with fewer viewports, and fewer viewports means fewer boundaries at which

pedestrians can be truncated. Object truncation is a well-documented challenge in detection [24, 25], and minimising the number of viewport boundaries directly reduces its frequency. This reasoning motivates our choice of a 2×3 composite layout (6 projections) over the 2×4 layout (8 projections) of Chiang *et al.* [5].

A wider FOV is viable because the gnomonic projection is the ideal model of pinhole cameras [26, 27], which are the cameras used to capture the data on which YOLO detectors are trained. Widening the FOV beyond what is typical of commercial pinhole cameras introduces radial stretching (the farther from the viewport centre, the greater the stretching of depicted objects), but this effect is similar

in nature to the perspective and shear augmentations routinely applied during YOLO training. Furthermore, we conjecture that recent YOLO versions [28–30] exhibit improved tolerance to gnomonic distortion compared to the older models used in Ref. [5], enabling a safer increase of the FOV. Although the stretching across a gnomonic viewport is non-uniform, the intra-object stretching within the pixels depicting a single pedestrian remains nearly constant, since each person occupies only a small fraction of a wide-FOV viewport. We illustrate this in Fig. 1: the distortion produced by gnomonic projection (Fig. 1(b)) is visually comparable to standard YOLO augmentations such as shear (Fig. 1(c)) or stretch (Fig. 1(d)).



Fig. 1. Comparison of image distortions on a pedestrian (zoomed within the red box for better clarity): (a) Original image generated with an ideal FOV and object centering for an ideal non-stretched object (b) Distortion due to gnomonic projection. (c) Image augmented with a shear transformation. (d) Image augmented with a stretch transformation.

B. Selected Configuration

1) Number of projections

Our goal is to use fewer projections than the 8-view baseline of Ref. [5]. Since the composite image stacks projections into a rectangular grid, the number of projections must be even. The candidates below 8 are therefore 6, 4, and 2. Two projections are immediately impractical: to cover the full fisheye circumference with only two viewpoints, each would require a horizontal FOV close to 180° , producing extreme distortions. Four projections are feasible but require a minimum horizontal FOV of 90° for full coverage, which introduces significantly more distortion per viewport. Six projections represent the most attractive option: the horizontal FOV required for seamless coverage ($360^\circ/6 = 60^\circ$ per step) is moderate and well within the distortion tolerance established by the design rationale. We therefore focus our investigation on 6 projections, while also evaluating the 4-projection case for completeness. The 4-projection (2×2) layout consistently underperformed and is not retained.

2) Viewpoint arrangement

Each projection is defined by a viewing direction expressed as a longitude λ and a latitude φ in the fisheye spherical coordinate system. We fix a common latitude for all projections (e.g., $\varphi = 45^\circ$), positioning each viewport to look toward the mid-range of the scene. Longitudes are distributed uniformly: starting from an arbitrary origin $\lambda_0 = 0^\circ$, subsequent projections are placed at increments of $\Delta\lambda = 360^\circ/N$, where N is the number of projections. This

uniform arrangement ensures even azimuthal coverage of the fisheye image without any privileged direction.

3) Grid layout

For 6 projections we evaluated both 3×2 (3 columns, 2 rows) and 2×3 (2 columns, 3 rows) composite layouts. The 2×3 arrangement—3 projections per row, 2 rows—proved more suitable because it allocates more horizontal pixels per projection cell, better matching the portrait aspect ratio of a standing pedestrian. We also tested an asymmetric layout consisting of one central nadir projection (looking directly downward, $\varphi \approx 0^\circ$) combined with 5 uniformly distributed lateral projections. The central view covers the area directly beneath the camera, while the 5 lateral views handle the surrounding scene with a reduced vertical FOV since the nadir is already covered. Fig. 2 illustrates this layout: the composite image (Fig. 2(a)) shows the 6 sub-views assembled into the 2×3 grid, and the fisheye overlay (Fig. 2(b)) shows the corresponding projection boundaries on the original image. This layout did not yield a consistent improvement and was discarded in favor of the fully uniform $2 \times 3 =$ arrangement.

4) FOV and latitude tuning

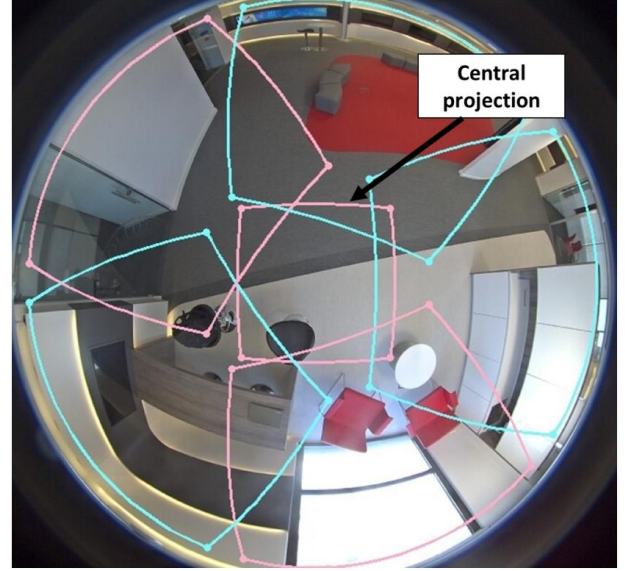
With the 2×3 layout and $\Delta\lambda = 60^\circ$ fixed, we tune the horizontal FOV, vertical FOV, and latitude jointly. The horizontal FOV must be at least 60° to avoid significant gaps between adjacent projections; increasing it beyond this minimum adds overlap at seams, reducing truncation at the cost of greater distortion near the viewport edges. The vertical FOV and latitude together determine how

much of the radial extent of the fisheye is covered: latitude sets the central elevation of each projection, and the vertical FOV extends coverage by $\pm vFOV/2$ around that centre (e.g., with latitude $\varphi = 45^\circ$ and $vFOV = 85^\circ$, the projection covers latitudes from $45^\circ - 42.5^\circ = 2.5^\circ$ to $45^\circ + 42.5^\circ = 87.5^\circ$, nearly spanning the full radial extent of the fisheye). An excessively large FOV in either direction

causes pedestrians near the viewport periphery to appear heavily tilted and distorted, degrading detection quality. We search over this parameter space on three datasets and report the results in Section V-C. The configuration retained for the main comparison against Chiang *et al.* [5] uses $hFOV = 60^\circ$, $vFOV = 85^\circ$, and $\varphi = 45^\circ$.



(a) Composite image in (2, 3) grid. The projection at the top-left corner corresponds to the central projection in (b)



(b) Projection boundaries on fisheye

Fig. 2. Non-uniform 2×3 layout: one central nadir projection combined with 5 uniformly distributed lateral projections. The composite image (a) shows the assembled sub-views; the fisheye overlay (b) shows the corresponding projection boundaries.

C. Gnomonic Projection and Composite Image Creation

Our approach generates perspective views from a fisheye image through virtual camera projections positioned at specified viewing directions. Each perspective view is defined by three parameters: viewing direction (longitude λ and latitude φ in spherical coordinates), Field-of-View (FOV) angle α , and output resolution $W_{out} \times H_{out}$ in pixels. The transformation from fisheye to perspective coordinates proceeds through three stages.

1) Rotation matrix computation

The virtual camera's viewing orientation is expressed as a rotation matrix $R = [R \mid U \mid F]$ constructed from orthonormal basis vectors:

$$F = (\sin\phi\cos\lambda, \sin\phi\sin\lambda, -\cos\phi) \quad (2)$$

$$R = \frac{F \times U_{world}}{\|F \times U_{world}\|}, U = \frac{R \times F}{\|R \times F\|} \quad (3)$$

where $U_{world} = (0, 0, 1)$ is the world up vector. The forward vector F defines the viewing direction, while the orthonormal basis ensures consistent coordinate frame orientation across all projections.

2) Perspective projection

For each output pixel (u, v) in the perspective view, local camera coordinates are computed as:

$$x_{local} = \frac{u - W_{out}/2}{f}, y_{local} = \frac{H_{out}/2 - v}{f}, z_{local} = 1 \quad (4)$$

where $f = (W_{out}/2) / \tan(\alpha/2)$ is the focal length derived from the desired field-of-view α . These local coordinates represent the direction from the virtual camera center through pixel (u, v) .

3) Inverse mapping to fisheye space

The local camera directions are rotated into world coordinates. Since the fisheye image is stored as a 2D pixel array, we employ a normalization step to convert 3D directions onto the unit sphere, establishing a bijection between 2D pixel coordinates and 3D surface locations. The direction vectors are computed as:

$$d_{world} = R^T \cdot \frac{(x_{local}, y_{local}, z_{local})}{\|(x_{local}, y_{local}, z_{local})\|} \quad (5)$$

Spherical angles are extracted from the unit direction vector as:

$$\theta' = \arccos(-d_{world,z}), \psi = \arctan2(d_{world,y}, d_{world,x}) \quad (6)$$

Following the equidistant fisheye projection model (Eq. (1)), these angles map to fisheye image coordinates:

$$r_{\text{px}} = \frac{\theta'}{\pi/2} \cdot R_{\text{fisheye}} \quad (7)$$

$$u_{\text{fish}} = u_c + r_{\text{px}} \cos \psi \quad (8)$$

$$v_{\text{fish}} = v_c + r_{\text{px}} \sin \psi \quad (9)$$

where R_{fisheye} is the fisheye image radius and (u_c, v_c) is the fisheye image center. These transformations are precomputed once during initialization as projection matrices m_x and m_y that establish point-to-point correspondence between perspective and fisheye domains. For each pixel coordinate (u, v) in the perspective view, these matrices directly provide the corresponding fisheye coordinates:

$$I_{\text{persp}}(u, v) = I_{\text{fisheye}}(m_x(u, v), m_y(u, v)) \quad (10)$$

This mapping enables efficient perspective view extraction with sub-pixel accuracy.

4) Composite image creation

Once all perspective projections are computed, they are assembled into a single composite image by arranging them into the defined $R \times C$ grid (2 rows, 3 columns in our case), concatenating projections row by row. Since the composite image will ultimately be fed to the YOLO detector at a fixed input resolution $W_{\text{YOLO}} \times H_{\text{YOLO}}$, each individual viewport should be generated at exactly $\lceil W_{\text{YOLO}}/C \rceil \times \lceil H_{\text{YOLO}}/R \rceil$ pixels. For example, with a YOLO input of 640×640 and a 2×3 grid, each viewport is generated at $\lceil 640/3 \rceil \times \lceil 640/2 \rceil = 213 \times 320$ pixels. Generating projections at a higher resolution and subsequently downscaling them to fit the composite would waste computation without any quality benefit. This resolution-aware generation strategy ensures that the pixel budget allocated by the YOLO input is used directly and efficiently, with no redundant resampling step.

D. Altitude-agnostic Geometric Backprojection

A fundamental limitation of the approach proposed in Ref. [5] is that back-projection relies on a pre-defined set of target bounding boxes computed offline at a fixed camera altitude. As illustrated in Fig. 3, these target boxes are distributed across the fisheye image at fixed sizes and positions, and the final fisheye bounding box is derived from this pre-computed set. If the camera is deployed at a different altitude, pedestrians appear at different scales in the fisheye image, rendering the pre-defined targets inadequate. Our back-projection, by contrast, is derived entirely from the gnomonic projection geometry: given any detection on the composite image, the corresponding fisheye box is computed analytically from the projection matrices, with no dependency on altitude or any pre-calibrated lookup. This makes our method directly applicable to varying deployment heights without any additional calibration or offline preparation.

1) Back-projection from single perspective views

For each bounding box detection represented by its axis-aligned bounds (x_1, y_1, x_2, y_2) in a perspective view, we perform geometric backprojection through three sequential operations: center mapping, dimension transformation, and radial alignment.

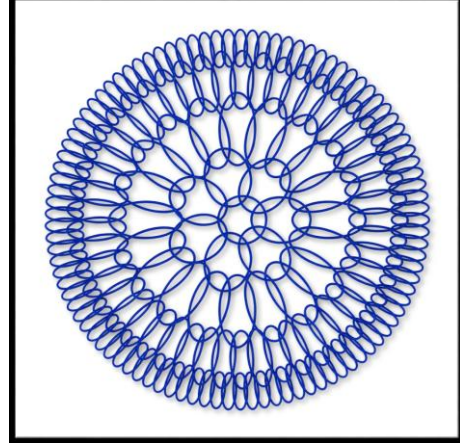


Fig. 3. Pre-defined target boxes used in Ref. [5], reproduced from the original paper. Target boxes are plotted as ellipses for easier visualization, with a low overlap ratio of 0.2 between adjacent boxes. The sizes and positions of these targets are fixed at a specific camera altitude.

a) Center mapping

The bounding box center (x_c, y_c) in perspective coordinates, where $x_c = (x_1 + x_2)/2$ and $y_c = (y_1 + y_2)/2$, is directly mapped to fisheye coordinates using the precomputed projection matrices:

$$(u_{\text{fish},c}, v_{\text{fish},c}) = (m_x(x_c, y_c), m_y(x_c, y_c)) \quad (11)$$

b) Dimension transformation

The gnomonic-to-fisheye transformation exhibits spatially-varying distortion: the relationship between perspective and fisheye distances depends on location within the image. Consequently, a bounding box with width w pixels on the composite does not correspond to a uniform fisheye width—the mapping scales differently at different positions. To preserve tight object fit when transforming from composite to fisheye, we employ a geometric distance measurement approach rather than direct pixel dimension transfer.

For width, we backproject the left and right edge locations at the bbox's vertical centerline, then measure the Euclidean distance between them in fisheye space. Similarly, for height, we backproject the top and bottom edge locations at the bbox's horizontal centerline and measure their fisheye-space separation. By sampling the transformation at the actual edge locations (rather than applying a global scale factor), we ensure that a bbox tightly enclosing a pedestrian on the composite image maintains that tight fit on the fisheye image.

Width transformation: We backproject the left and right edge midpoints at the bbox's vertical center:

$$(u_{\text{left}}, v_{\text{left}}) = (m_x(x_1, y_c), m_y(x_1, y_c))$$

$$(u_{\text{right}}, v_{\text{right}}) = (m_x(x_2, y_c), m_y(x_2, y_c)) \quad (12)$$

The fisheye-space width is the Euclidean distance between these backprojected edge points:

$$w_{\text{fish}} = \|(u_{\text{right}}, v_{\text{right}}) - (u_{\text{left}}, v_{\text{left}})\|_2 \quad (13)$$

Height transformation: Similarly, we backproject the top and bottom edge midpoints at the bbox's horizontal center:

$$(u_{\text{top}}, v_{\text{top}}) = (m_x(x_c, y_1), m_y(x_c, y_1))$$

$$(u_{\text{bottom}}, v_{\text{bottom}}) = (m_x(x_c, y_2), m_y(x_c, y_2)) \quad (14)$$

The fisheye-space height is:

$$h_{\text{fish}} = \|(u_{\text{bottom}}, v_{\text{bottom}}) - (u_{\text{top}}, v_{\text{top}})\|_2 \quad (15)$$

c) Radial alignment

Following the radially-aligned bounding box convention introduced by Yang *et al.* [18], we construct a rotated rectangle with dimensions $(w_{\text{fish}}, h_{\text{fish}})$ centered at $(u_{\text{fish},c}, v_{\text{fish},c})$ and oriented radially toward the fisheye center. The rotation angle θ_{radial} is computed as:

$$\theta_{\text{radial}} = \arctan2(v_{\text{fish},c} - v_{\text{center}}, u_{\text{fish},c} - u_{\text{center}}) \quad (16)$$

where $(u_{\text{center}}, v_{\text{center}})$ represents the fisheye image center. The four corners of the radially-aligned bounding box are constructed by rotating a rectangle of dimensions $(w_{\text{fish}}, h_{\text{fish}})$ around the backprojected center, with the height dimension oriented radially and the width dimension oriented tangentially. This backprojection strategy avoids the computational overhead of dense point sampling while accurately preserving the geometric relationship between perspective detections and their fisheye-space representations.

2) Back-projection from composite images

When YOLO runs on the composite image, each detection is reported in composite image coordinates. Before applying the back-projection described above, the detection must first be mapped back to the coordinate system of the individual perspective view it originated from. Since each perspective cell occupies a known rectangular region within the composite (determined by the grid layout and the per-cell resolution), this is a straightforward offset subtraction: the cell containing the detection centre is identified from its grid position (r, c) , and the detection coordinates are shifted by the cell's top-left corner to obtain view-local coordinates. The per-cell back-projection procedure then applies without modification, using the projection matrices of the corresponding view to recover fisheye coordinates.

E. Multi-view Detection Fusion

We adopt the two-stage NMS strategy of Ref. [5] without modification. A standard NMS pass is first applied

on the composite image to suppress redundant detections within each perspective view. After back-projection to fisheye space, a second pass applies Gaussian Soft-NMS on the resulting rotated boxes [6]. All NMS parameters are identical to those used in Ref. [5].

V. EXPERIMENTAL SETUP

A. Datasets

We evaluate on three public fisheye pedestrian benchmarks covering diverse environments, camera resolutions, and annotation conventions (Table II).

TABLE II. SUMMARY OF EVALUATION DATASETS

Dataset	Frames	Resolution	GT Annotation	Environment
BOMNI	245	640 × 480	Radial-aligned	Single room
PIROPO	3004	800 × 600	Radial-aligned	Two rooms
CEPDOF	3000*	up to 2048 × 2048	Free-orientation	Multi-sequence

* Spread-sampled from 25,358 total frames.

BOMNI [23] contains 245 annotated frames from four ceiling-mounted fisheye cameras (640 × 480 px) in a single indoor room. The limited frame count reflects the sparsely sampled annotations released by Tamura *et al.* [23], which cover only a subset of the original recording; additionally, we identified and discarded a number of frames carrying incorrect annotations during an integrity check. The limited size and repetitive scene content of BOMNI reduce statistical confidence; BOMNI results should be interpreted with caution.

PIROPO (People in Indoor ROoms with Perspective and Omnidirectional cameras) [31] contains recordings from two rooms across four omnidirectional cameras (800 × 600 px). The 3004 annotated frames used in this study result from the subset covered by Tamura *et al.*'s [23] annotations combined with our removal of frames bearing incorrect annotations during an integrity check. We found no issues in the 3000 spread-sampled CEPDOF frames checked.

CEPDOF (Challenging Events for Person Detection from Overhead Fisheye Images) [12] is the largest dataset used in this study, with 25,358 frames recorded by ceiling fisheye cameras at resolutions up to 2048 × 2048 px across eight activity sequences. Unlike BOMNI and PIROPO, CEPDOF ground-truth annotations are free-orientation: boxes are still rotated bounding boxes, but their angle reflects the actual physical orientation of each person rather than being constrained to point toward the fisheye centre. We spread-sample 3000 frames uniformly by dividing the full sequence into 3000 equal-length segments and taking one frame per segment. This is well-justified because CEPDOF sequences are videos and adjacent frames are nearly identical; dense evaluation would heavily over-represent static or slow-motion segments without meaningful gain in metric reliability. Evaluating all 25,358 frames per configuration across all YOLO backbones would also be prohibitively time-consuming.

Rotated bounding-box annotations for all three datasets were produced by Tamura *et al.* [23] and are publicly

available online. Ground-truth boxes in BOMNI and PIROPO are radially-aligned rotated boxes; those in CEPDOF are also rotated boxes but with free orientation, i.e. not constrained to radial alignment (see Section V-B).

B. Evaluation Metrics

All metrics are computed between rotated bounding boxes in fisheye image space. Detections undergo two NMS stages following Chiang *et al.* [5]: standard NMS on the composite image before backprojection, then Gaussian Soft-NMS on the backprojected rotated boxes in fisheye space.

The standard detection metric is $AP@[0.50:0.95]$ (Common Objects in Context, COCO-style), which averages AP over IoU thresholds from 0.50 to 0.95 and provides the most comprehensive single-number summary of detection quality across localization strictness levels. However, our pipeline constrains all backprojected predictions to be radially aligned—the predicted box height points toward the fisheye centre, as is standard for top-view fisheye detectors [18, 22]—whereas CEPDOF ground truth uses free-orientation annotations. At tight IoU thresholds (≥ 0.80), a perfectly localized prediction may receive a lower overlap score solely because its orientation convention differs slightly from the ground truth. We therefore also report $AP@[0.50:0.75]$ (mean AP over $\{0.50, 0.55, \dots, 0.75\}$, six thresholds at step 0.05), which caps the range below those tight thresholds and thus avoids penalizing radially-aligned predictions for this convention mismatch rather than genuine localization error. Both metrics are reported throughout; $AP@[0.50:0.95]$ serves as the COCO reference and $AP@[0.50:0.75]$ is more representative of genuine localization quality on CEPDOF. Configuration ranking is based on the full set of reported metrics rather than any single value. We additionally report $AP@0.50$ (Pascal VOC), $AP@0.75$ (single strict threshold), $Precision@0.50$, $Recall@0.50$, and $F1@0.50$. Throughput is reported as Frames Per Second (FPS), computed as the reciprocal of the mean end-to-end pipeline latency per frame (projection, YOLO inference, backprojection, and Soft-NMS).

C. Validation Strategy

We validate the proposed configuration in three stages. First, we conduct a direct comparison of our six-projection configuration against the Chiang *et al.* [5] eight-projection baseline using YOLOv9e on all three datasets (Section VI-A). Second, we repeat the comparison across five additional YOLO variants—YOLOv8n, YOLOv8m, YOLOv10m, YOLO11m, and YOLO12m—to verify that the observed gains are backbone-agnostic (Section VI-B). Third, we conduct a configuration parameter study comparing ten configurations spanning 4, 6, 8, and 9 projections with varying horizontal FOV and latitude, to characterize which parameters drive performance (Section VI-C).

D. Implementation Details

1) Hardware and software

All experiments were conducted on a desktop workstation equipped with an NVIDIA GeForce RTX 4060 GPU (8 GB VRAM), an Intel Core i5-12400F CPU (3.6 GHz), and 16 GB system memory, running Windows 11 Pro with CUDA 12.6 (driver 560.94). The detection pipeline is implemented in Python using the Ultralytics library for YOLO inference and OpenCV for projection computation and image processing.

2) YOLO backbones

We evaluate six YOLO variants spanning successive architecture generations: YOLOv8n and YOLOv8m [28], YOLOv9e [32], YOLOv10m [33], YOLO11m [29], and YOLO12m [30]. All models use publicly available pretrained weights without any fine-tuning on fisheye data. The nano (n), medium (m), and extended (e) size suffixes reflect increasing model capacity, allowing us to verify that the observed configuration gains hold across backbones of different strengths.

VI. EXPERIMENTAL RESULTS

We compare our proposed six-projection configuration (2×3 grid, h-FOV 60° , v-FOV 85° , latitude 45°) against the eight-projection baseline of Chiang *et al.* [5] (2×4 grid, h-FOV 48° , v-FOV 96° , latitude 36°). Section VI-A presents detailed per-dataset results using the strongest backbone (YOLOv9e). Section VI-B examines whether the gains hold across six YOLO variants to rule out backbone-specific effects. Section VI-C reports the full parameter study over ten projection configurations. Section VI-D studies the accuracy-speed trade-off when reducing the YOLO input resolution below the default 640 px.

A. Proposed Configuration vs. Chiang Baseline

Table III reports detection metrics on the three evaluation datasets using YOLOv9e, which achieves the strongest overall detection performance across all tested configuration-backbone pairs (see Section VI-B). Our proposed configuration outperforms the Chiang baseline on the primary AP metric across all three datasets: $AP@[0.50:0.95]$ gains of +6.5% on PIROPO and +4.6% on BOMNI, and an $AP@[0.50:0.75]$ gain of +5.1% on CEPDOF (the preferred metric for CEPDOF due to the free-orientation annotation convention, see Section V-B). These improvements are corroborated by consistent gains in $AP@0.75$, Recall, and F1.

On BOMNI and CEPDOF, the largest relative improvements occur at $AP@0.75$: +8.0% on BOMNI and +6.8% on CEPDOF. We attribute this to the wider horizontal FOV per projection (60° vs. 48°), which reduces the frequency of pedestrians intersecting the vertical boundaries between adjacent columns. Boundary intersections force the detector to operate on truncated silhouettes, producing imprecise bounding boxes that score well at loose IoU thresholds but fail at tighter ones. On BOMNI, the small recall gain (+1.3%) is further explained by the dataset's persistent false negatives: nearly stationary people in difficult poses that both configurations miss equally, regardless of tile layout.

TABLE III. DETECTION PERFORMANCE OF THE PROPOSED CONFIGURATION VS. CHIANG [5]

Metric	BOMNI ($n = 245$) [‡]			PIROPO ($n = 3000$)			CEPDOF ($n = 3000$)		
	Chiang	Ours	$\Delta\%$	Chiang	Ours	$\Delta\%$	Chiang	Ours	$\Delta\%$
AP@[0.50:0.75] [†]	0.353	0.420	+6.7	0.500	0.608	+10.8	0.458	0.509	+5.1
AP@[0.50:0.95]	0.214	0.260	+4.6	0.325	0.390	+6.5	0.284	0.327	+4.3
AP@0.50	0.550	0.553	+0.3	0.698	0.803	+10.5	0.687	0.755	+6.8
AP@0.75	0.052	0.132	+8.0	0.250	0.293	+4.3	0.145	0.213	+6.8
Precision@0.50	0.652	0.703	+5.1	0.704	0.831	+12.7	0.718	0.753	+3.5
Recall@0.50	0.659	0.672	+1.3	0.720	0.857	+13.7	0.763	0.805	+4.2
F1@0.50	0.656	0.687	+3.1	0.712	0.844	+13.2	0.740	0.778	+3.8
FPS	13.99	14.20	+0.21	14.93	14.91	-0.02	13.86	13.71	-0.15

Noite: [†] Mean AP over IoU $\in \{0.50, 0.55, \dots, 0.75\}$. Thresholds above 0.75 are excluded for CEPDOF due to annotation convention mismatch at high IoU. All results use a 640×640 YOLO input resolution. [‡]: Lower Statistical Confidence (Limited Dataset size, Single Room, Repetitive Frames). BOLD: Better value per metric per dataset. $\Delta\%$: (Ours-chiang) $\times 100$ for detection metrics; Absolute difference for FPS. Green: the difference favors ours method, red otherwise.

On PIROPO, the gain pattern is different: the largest improvements are in Recall@0.50 (+13.7%), Precision@0.50 (+12.7%), and AP@[0.50:0.75] (+10.8%), while AP@0.75 improves more modestly (+4.3%). The recall gain is explained by a subtler boundary effect: when a pedestrian falls equally across two adjacent tiles, neither half provides sufficient visual context for the detector to fire at AP@0.50, even though the person is technically visible in both. Fig. 4 illustrates this on a

representative PIROPO frame: under Chiang's 2×4 layout, the only pedestrian is split equally between the third and fourth subviews of the second row, and goes undetected; under our 2×3 layout, the same pedestrian falls within a single wider subview and is successfully detected (confidence 0.62). Since our layout has one fewer column, such equal-split boundary cases are less frequent, directly reducing false negatives and raising recall.

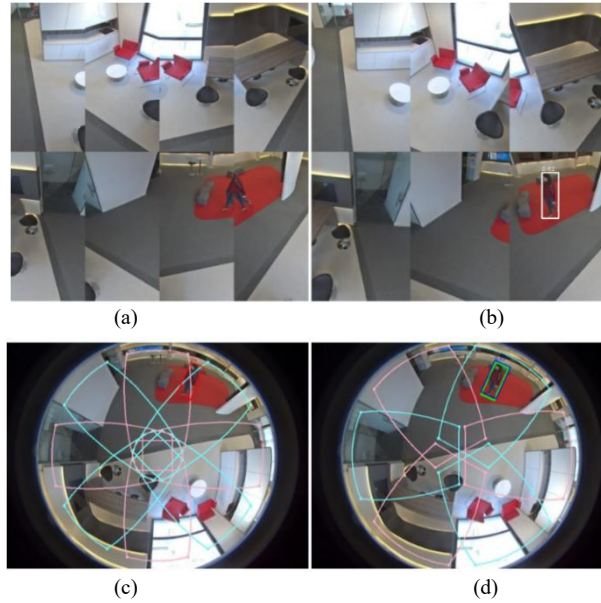


Fig. 4. Boundary truncation causing a false negative under Chiang's configuration (PIROPO dataset, frame 150—the only pedestrian in the scene). (a) In Chiang's 2×4 composite, the pedestrian is split equally between the 3rd and 4th subviews of the bottom row; no detection is produced (no white box). (b) In our 2×3 composite, the same pedestrian falls within a single wider subview and is detected (white box, conf. 0.62). (c) Chiang's fisheye result: the ground-truth box (green) is marked as a false negative (red box). (d) Our fisheye result: the pedestrian is correctly detected (blue box) with good overlap with the ground truth (green). Pink and cyan outlines show the backprojected boundaries of alternating subview rows.

The BOMNI AP@0.50 gain is negligible (+0.3%), consistent with the earlier observation that both configurations miss the same hard-pose pedestrians equally—the detection rate at loose IoU is therefore nearly identical. The gain concentrates instead in localization quality, as reflected by the primary metric AP@[0.50:0.95] (+4.6%). We caution that BOMNI results carry lower statistical confidence than PIROPO and

CEPDOF given the dataset's limited size and homogeneous content.

Regarding throughput, FPS is virtually unchanged between configurations (mean 14.27 vs. 14.26 fps), despite the difference in the number of viewports. This is expected for two reasons. First, as described in Section IV-C4, each viewport is generated at exactly the pixel resolution it occupies within the composite, so the total number of

pixels projected across all viewports is always the same, regardless of how many viewports the grid contains. Second, YOLO always receives a fixed $W_{\text{YOLO}} \times H_{\text{YOLO}}$ composite, so inference cost is equally unaffected by grid layout. Both the projection and inference steps therefore have practically the same computational cost across configurations.

B. Robustness across YOLO Backbones

Table IV reports the per-dataset AP gain of our configuration over the Chiang baseline across six YOLO backbones spanning a wide capacity range.

TABLE IV. GAIN OF THE PROPOSED CONFIGURATION (OURS) OVER CHIANG [5] ACROSS SIX YOLO BACKBONES IN DIFFERENCE PERCENTAGE Δ : $(\text{OURS} - \text{CHIANG}) \times 100$

Backbone	BOMNI ^{a,‡}	PIROPO ^a	CEPDOF ^b	Mean
YOLOv8n	+4.0	+4.7	+3.2	+4.0
YOLOv8m	-1.0	+7.6	+3.0	+3.2
YOLOv10m	+1.0	+6.1	+2.9	+3.3
YOLO11m	+1.6	+5.7	+1.3	+2.9
YOLO12m	+5.9	+4.9	+3.6	+4.8
YOLOv9e	+4.6	+6.5	+5.1	+5.4

^a AP@[0.50:0.95] (COCO-style). ^b AP@[0.50:0.75] (primary metric, free-orientation GT). [‡] Lower statistical confidence (245 frames, single room). All results use a 640×640 YOLO input resolution. MEAN: Equal-weight average. Green: Ours wins, Red Otherwise.

Crucially, both configurations use the same YOLO backbone in each row: the observed gains reflect the projection layout exclusively, not any advantage in detector capacity. Across six backbones and three datasets (18 combinations total), the proposed configuration outperforms the baseline in 17 of 18 cases (94%). The single exception occurs on BOMNI with YOLOv8m (-1.0%), the dataset with the lowest statistical confidence (245 frames, single room). On PIROPO and CEPDOF, which together represent 6000 diverse fisheye frames, gains are positive for all six backbones without exception.

All mean gains are positive, ranging from +2.9% (YOLO11m) to +5.4% (YOLOv9e), confirming that the configuration advantage holds across backbones of different capacities rather than being an artifact of a specific architecture.

C. Effect of Projection Configuration Parameters

Table V reports per-dataset AP for ten configurations spanning projection counts of 4, 6, 8, and 9, all evaluated with YOLOv9e (Best YOLO performer). The top four

positions are exclusively occupied by six-projection 2×3 layouts, differing only in h-FOV (60° or 70°), v-FOV, and latitude. No other family comes close: the fifth-ranked entry is the Chiang baseline [5], already 3.8% mean AP below the weakest member of the six-projection family.

Increasing the projection count to nine (3×3 grid) does not help—the best nine-projection result (mean AP = 0.422) still falls below the weakest six-projection variant (mean AP = 0.475). Four-projection configurations rank at the bottom: the 90° h-FOV variant suffers from insufficient panoramic coverage, while the 106° h-FOV (80° v-FOV) variant achieves broader coverage at the cost of excessive perspective distortion that degrades pedestrian appearance within each tile.

These results suggest that the six-projection 2×3 layout is the sweet spot that best exploits our tiling assumption within practical FOV ranges. The FOV values were explored within bounds chosen to ensure full fisheye coverage while avoiding excessively wide angles that would stretch pedestrian silhouettes and shrink their apparent size, degrading detection.

TABLE V. PER-DATASET AP FOR TEN PROJECTION CONFIGURATIONS EVALUATED WITH YOLOv9e, SORTED BY MEAN AP

Grid	Np	h-FOV	v-FOV	Lat.	BOMNI	PIROPO	CEPDOF	Mean
2×3	6	60°	90°	50°	0.422	0.654	0.501	0.526
$2 \times 3^\dagger$	6	60°	85°	45°	0.420	0.608	0.509	0.512
2×3	6	60°	90°	45°	0.369	0.639	0.479	0.496
2×3	6	70°	90°	45°	0.336	0.599	0.489	0.475
2×4 [5]	8	48°	96°	36°	0.353	0.500	0.458	0.437
3×3	9	60°	90°	45°	0.182	0.652	0.432	0.422
2×2	4	90°	90°	45°	0.344	0.433	0.448	0.408
3×3	9	50°	90°	45°	0.221	0.578	0.342	0.380
2×2	4	106°	80°	50°	0.274	0.401	0.418	0.364
3×3	9	40°	90°	45°	0.081	0.373	0.188	0.214

[†] Configuration used in the head-to-head comparison of Table II. BOMNI and PIROPO use AP@[0.50:0.95] (Radially-aligned GT); CEPDOF uses AP@[0.50:0.75] (Free-orientation GT). Mean: Equal-weight average. All results use 640×640 input.

Within those bounds, the secondary parameters (v-FOV, latitude) were varied experimentally, and the parameter study reveals that a variant with v-FOV = 90° and latitude = 50° achieves an even higher mean AP than the configuration tested against Chiang's baseline in Section VI-A [5].

D. Resolution Ablation

The 640×640 YOLO inference resolution used throughout the previous experiments is the default and highest setting tested. To assess the speed-accuracy trade-off, we evaluate the proposed configuration (2×3

grid, h-FOV 60°, v-FOV 85°, lat. 45°) at three lower resolutions: 512 × 512, 416 × 416, and 320 × 320. Since the composite image is fed directly to YOLO at the target size, the composite resolution and the YOLO input resolution are one and the same. The backbone is YOLOv9e throughout, the same model used in the head-to-head comparison of Section VI-A. All other parameters (datasets, NMS) remain unchanged. Table VI reports AP@[0.50:0.95] for BOMNI and PIROPO and AP@[0.50:0.75] for CEPDOF, alongside pipeline throughput (FPS).

Three observations are worth highlighting. First, 512 px yields no meaningful throughput gain over 640 px (14.4 vs. 14.3 FPS) while incurring a consistent accuracy drop across all datasets—there is therefore no practical justification for using 512 px; the full 640 px resolution is strictly preferable. Second, the 416 px variant surpasses Chiang on CEPDOF (AP@[0.50:0.75]: 0.365 vs. 0.325) with a 1.4× speedup, but falls marginally below BOMNI. Third, even at 320 px the proposed configuration still outperforms Chiang on PIROPO (AP@[0.50:0.95]: 0.333 vs. 0.325), while falling below on BOMNI and CEPDOF; the 2.2× speedup (31.5 vs. 14.3 FPS) makes this resolution worth considering for latency-critical deployments. Across all three observations, BOMNI results carry substantially lower statistical weight than PIROPO and CEPDOF (245 vs. 3000 frames each) and should not be given equal standing.

TABLE VI. RESOLUTION ABLATION ON THE PROPOSED CONFIGURATION (YOLOv9E, 2 × 3 COMPOSITE)

Resolution	BOMNI ^a	PIROPO ^a	CEPDOF ^b	FPS
640 px [1]	0.214	0.325	0.458	14.3
640 px (ours)	0.260	0.390	0.509	14.3
512 px (ours)	0.228	0.373	0.481	14.4
416 px (ours)	0.174	0.365	0.452	20.1
320 px (ours)	0.143	0.333	0.395	31.5

^a AP@[0.50:0.95] (COCO-style). ^b AP@[0.50:0.75] (primary metric, free-orientation GT). FPS is the mean end-to-end pipeline throughput across all three datasets. Per Column: Green = Best, Blue = Second, Red = Third.

VII. CONCLUSION

We showed that optimizing the number of viewpoints, FOV, and viewing latitude yields important gains. We found that a six-projection layout (h-FOV 60°, v-FOV 85°, lat. 45°) for pedestrian detection is significantly better than the existing baseline. We supported our findings through rigorous experimental study, including experiments—head-to-head comparison, configuration search, backbone isolation, and a resolution ablation, demonstrating that lower resolutions are still competitive with the high-resolution 640 px baseline while providing speedups of up to 2.2 times. We integrated in all of these experiments an altitude-agnostic bounding box inference, which allows to back-project the bounding boxes detected on the gnomonic projections onto the original fisheye

image without requiring any prior knowledge on cameras' altitude.

We point out that while our search for parameters was intensive, it could be more exhaustive. A constrained mathematical optimization that encodes known failure modes would be a more sensible strategy (too wide FOV distorts silhouettes; too many or too few projections increase boundary frequency or reduce coverage). The fact that NMS depends on IoU overlap and is unable to handle boundary-split detections—in which a truncated pedestrian creates two partial boxes with minimal spatial overlap that are neither suppressed nor merged—is another unresolved issue with the detection fusion step. We are currently working on resolving this failure by a customized fusion strategy that finds and combines boundary-adjacent partial detections.

CONFLICT OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

AUTHOR CONTRIBUTIONS

YZ conceptualized the research topic and overall approach, formulated the central hypothesis on gnomonic projection parameter optimization, proposed the methodological innovations, derived the mathematical framework, designed and implemented the end-to-end detection pipeline, designed the experimental validation strategy, conducted the projection parameter search and resolution ablation, produced the figures and tables, drafted the initial manuscript, and supervised the overall project.

JJ conducted the literature review, confirmed the research gap, contributed to the comparative analysis of existing approaches, collected and prepared the BOMNI, PIROPO, and CEPDOF benchmarks, performed annotation integrity checks and developed the sampling and cleaning pipelines, ran a subset of the ablation experiments, and contributed to writing.

JF designed and implemented the evaluation infrastructure for rotated bounding boxes, ran multiple experimental sessions, and contributed to the analysis of the numerical results.

The aforementioned author contributions are not exhaustive. All authors verified and approved the final version of the manuscript.

REFERENCES

- [1] H. Rashed, E. Mohamed, and G. Sistu *et al.*, “Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline,” in *Proc. the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 2271–2279.
- [2] C. Obasi and K. M. Tahir, “A real-time smart door access control system using haar cascade classifier and embedded vision,” *International Journal of Data Informatics and Intelligent Computing*, vol. 4, pp. 14–22, 2025.
- [3] D. Yang *et al.*, “PGDS-YOLOv8s: An Improved YOLOv8s Model for Object Detection in Fisheye Images,” *Applied Sciences*, vol. 14, 44, 2024.

- [4] R. Seidel *et al.*, “Improved person detection on omnidirectional images with non-maxima suppression,” in *Proc. the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP/VISAPP)*, 2019, pp. 474–481.
- [5] S.-H. Chiang *et al.*, “Efficient pedestrian detection in top-view fisheye images using compositions of perspective view patches,” *Image and Vision Computing*, vol. 105, 104069, 2021.
- [6] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS—improving object detection with one line of code,” in *Proc. the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5561–5569.
- [7] X. Wei *et al.*, “DCPB: Deformable convolution based on the Poincaré ball for top-view fisheye cameras,” in *Proc. the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 13262–13271.
- [8] K. Han *et al.*, “GhostNet: More features from cheap operations,” in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1580–1589.
- [9] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.
- [10] G. Jocher. (May 2020). YOLOv5 by Ultralytics version 7.0. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [11] W. Yang, Y. Qian, J.-K. Kämäräinen, F. Cricri, and L. Fan, “Object detection in equirectangular panorama,” in *Proc. 2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 2190–2195.
- [12] Z. Duan, M. O. Tezcan, H. Nakamura, P. Ishwar, and J. Konrad, “RAPiD: Rotation-aware people detection in overhead fisheye images,” in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.
- [13] O. Haggui, H. Bayd, and B. Magnier, “Centroid human tracking via oriented detection in overhead fisheye sequences,” *The Visual Computer*, vol. 40, pp. 407–425, 2024.
- [14] Y. Yuan, W. Xiang, and D. Tang, “Rotated top-view fisheye person detection based on dynamic attention,” in *Proc. 2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS)*, 2023, pp. 75–78.
- [15] G.-S. Xia *et al.*, “DOTA: A large-scale dataset for object detection in aerial images,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3974–3983.
- [16] X. Dai *et al.*, “Dynamic head: Unifying object detection heads with attentions,” in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7373–7382.
- [17] C. Lyu *et al.*, “RTMDet: An empirical study of designing real-time object detectors,” arXiv Preprint, arXiv:2212.07784, 2022.
- [18] L. Yang *et al.*, “Large-scale person detection and localization using overhead fisheye cameras,” in *Proc. the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 19961–19971.
- [19] N. Wakai, S. Sato, Y. Ishii, and T. Yamashita, “Panoramic distortion-aware tokenization for person detection and localization using transformers in overhead fisheye images,” arXiv Preprint, arXiv:2503.14228, 2025.
- [20] S. Ren, X. Yang, S. Liu, and X. Wang, “SG-Former: Self-guided transformer with evolving token reallocation,” in *Proc. the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 6003–6014.
- [21] A.-T. Chiang and Y. Wang, “Human detection in fish-eye images using HOG-based detectors over rotated windows,” in *Proc. 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2014, pp. 1–6.
- [22] S. Li, M. O. Tezcan, P. Ishwar, and J. Konrad, “Supervised people counting using an overhead fisheye camera,” in *Proc. 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2019, pp. 1–8.
- [23] M. Tamura, S. Horiguchi, and T. Murakami, “Omnidirectional pedestrian detection by rotation invariant training,” in *Proc. the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1989–1998.
- [24] J. Ruan *et al.*, “A review of occluded objects detection in real complex scenarios for autonomous driving,” *Green Energy and Intelligent Transportation*, vol. 2, 100092, 2023.
- [25] S.-E. Ryu and K.-Y. Chung, “Detection model of occluded object based on YOLO using hard-example mining and augmentation policy optimization,” *Applied Sciences*, vol. 11, p. 7093, 2021.
- [26] V. Pegoraro, *Handbook of Digital Image Synthesis: Scientific Foundations of Rendering*, AK Peters/CRC Press, 2016.
- [27] G. Billings and M. Johnson-Roberson, “SilhoNet-fisheye: Adaptation of a ROI based object pose estimation network to monocular fisheye images,” *IEEE Robotics and Automation Letters*, vol. 5, pp. 4241–4248, 2020.
- [28] G. Jocher, A. Chaurasia, and J. Qiu. (2023). Ultralytics YOLOv8 version 8.0.0. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [29] G. Jocher and J. Qiu. (2024). Ultralytics YOLO11 version 11.0.0. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [30] Y. Tian, Q. Ye, and D. Doermann, “YOLO12: Attention-centric real-time object detectors,” *Advances in Neural Information Processing Systems*, vol. 38, pp. 78433–78457, 2025.
- [31] C. Del Blanco, P. Carballeira, F. Jaureguizar, and N. García, “Robust people indoor localization with omnidirectional cameras using a grid of spatial-aware classifiers,” *Signal Processing: Image Communication*, vol. 93, 116135, 2021.
- [32] C.-Y. Wang *et al.*, “YOLOv9: Learning what you want to learn using programmable gradient information,” in *Proc. European Conference on Computer Vision*, Cham: Springer Nature Switzerland, 2024, pp. 1–21.
- [33] A. Wang, H. Chen, L. Liu *et al.*, “YOLOv10: Real-time end-to-end object detection,” arXiv Preprint, arXiv:2405.14458, 2024.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).