Performance evaluation and integration of distortion mitigation methods for fisheye video object detection

John Benedict Du, Gian Paolo Mayuga, Maria Leonora Guico
Department of Electronics, Computer, and Communications Engineering, Ateneo de Manila University, Quezon City, Philippines

Article Info

Article history:

Received Feb 7, 2024
Revised Apr 13, 2024
Accepted Jun 19, 2024

Keywords:

Distortion mitigation
Fisheye cameras
Image remapping
Longitudinal-latitude correction
Real-time tracking
Video object detection
YOLOv3

ABSTRACT

The distortion observed in fisheye cameras has proven to be a persistent challenge for numerous state-of-the-art object detection algorithms, instigating the development of various techniques aimed at mitigating this issue. This study aims to evaluate various methods for mitigating distortion in fisheye camera footage and their impact on video object detection accuracy and speed. Using Python, OpenCV, and third-party libraries, the researchers modified and optimized said methods for video input and created a framework for running and testing different distortion correction methods and object detection algorithm configurations. Through experimentation with different datasets, the study found that undistorting the image using the longitude-latitude correction with the YOLOv3 object detector provided the best results in terms of accuracy (PASCAL: 68.9%, VOC-360: 75.1%, WEPDTOF: 15.9%) and speed (38 FPS across all test sets) for fisheye footage. After measuring the results to determine the best configuration for video object detection, the researchers also developed a desktop application that incorporates these methods and provides real-time object detection and tracking functions. The study provides a foundation for improving the accuracy and speed of fisheye camera setups, and its findings can be valuable for researchers and practitioners working in this field.

Corresponding Author:

John Benedict Du
Department of Electronics, Computer, and Communications Engineering, Ateneo de Manila University
Katipunan Ave, Quezon City, 1108 Metro Manila, Philippines
Email: john.du@obf.ateneo.edu

1. INTRODUCTION

This study focuses on the performance evaluation and integration of distortion mitigation methods for fisheye video object detection. Many establishments and public areas today employ a combination of regular and fish-eye cameras for surveillance and monitoring. While established algorithms and models excel in object detection using standard cameras, they face significant challenges when applied to fish-eye footage due to the inherent distortion caused by the wider lens and its increased field of view [1].

To address these challenges and enhance the accuracy and speed of object detection in fisheye camera setups, this research endeavors to systematically evaluate various distortion mitigation techniques. The aim is not only to identify the most effective approach but also to provide valuable insights for researchers and practitioners in this field. Moreover, beyond performance evaluation, this study explores the practical implementation of its findings in a real-time monitoring system that leverages fish-eye cameras. Such a system holds promise for a wide range of applications, including enhanced surveillance in public spaces, improved navigation in autonomous vehicles, and more accurate object tracking in various industries. This paper will delve into its potential benefits and applications in further detail.

Hemispherical lenses are increasingly favored due to their ability to capture panoramic shots, providing an extended field of view. However, such shots necessitate compensation for lens distortion through computer vision techniques. This distortion correction can take place before, during, or after object detection and tracking, with each approach offering distinct advantages and drawbacks in terms of speed, accuracy, preparation, system requirements, and compatibility. The complexity of these setups escalates as various combinations are explored to attain optimal outcomes.

Unlike conventional perspective cameras, single-fisheye cameras offer wider coverage at a lower cost. When mounted on ceilings, these cameras also reduce object occlusion [2]. Their popularity in expansive areas like parking lots, airports, malls, and warehouses is bolstered by the potential to enhance security. Their extensive field of view has implications beyond surveillance, benefiting vehicle safety systems, robot navigation, drones, virtual tours, endoscopy, and various other applications [3]-[11]. Despite inherent distortions, the trade-off between an improved field of view and distortion remains favorable, with fisheye cameras excelling in cost-effectiveness, compactness, and panoramic imaging. This trade-off is especially advantageous in constrained spaces, making fisheye cameras a preferred choice for a variety of scenarios.

The ubiquity of monitoring applications relies heavily on effective object detection, segmentation, and tracking, particularly in scenarios demanding heightened security and crisis management. While conventional cameras fulfill this role to a great extent, the utilization of fisheye cameras in security systems, owing to their wide-angle perspective, introduces unique challenges. Unlike their traditional counterparts, fisheye cameras lack standardized hardware optimization and necessitate substantial preprocessing. This distinct nature mandates careful consideration of setup parameters to ensure compatibility with a variety of detection algorithms. One of the primary obstacles lies in seamlessly integrating perspective and fisheye cameras within existing multi-camera systems. The inherent distortion in fisheye footage demands additional processing power to achieve detection accuracy and speed comparable to conventional images. Bridging the gap between perspective and fisheye setups is pivotal for harnessing the full potential of advanced detection techniques in diverse surveillance scenarios.

2. LITERATURE REVIEW

2.1. Object detection for perspective images

The domain of computer vision has extensively explored object detection in perspective images. Indulkar's [12] work showcases this, utilizing the YOLOv4 algorithm for social distancing and face mask detection, achieving a notable mean average precision (mAP) of around 90%. However, transposing these techniques to fisheye cameras presents unique hurdles. An example study using Faster-RCNN ResNet101 and YOLOv2 models for crowded restaurant detection reveals accuracy challenges in fisheye images, even with advanced techniques like inverse mapping and Supersampling [1]. Faster-RCNN outperformed YOLOv2, achieving a 76% detection rate compared to 61%, highlighting the need for precision enhancements [1]. While pursuing fisheye-specific strategies introduces computational complexities, it also opens avenues to optimize setups for a balance between accuracy and computational load, thereby advancing state-of-the-art detection algorithms.

2.2. Object detectors and datasets for fisheye

The adaptation of object detectors for fisheye imagery entails innovative modifications to tackle inherent challenges. To address distortions affecting smaller object detection, methods prioritize the preservation of image features. Notably, an altered YOLOv3 network, employing up-sampling and concatenation, elevates detection accuracy by up to 89.75% for cars and 87.23% for pedestrians [13]. Rotation-aware people detection (RAPiD) and anchor-free rotation-aware people detection (ARPD) networks, integrating rotation-aware functions, achieve remarkable accuracies of up to 97% [14]. Additionally, the FisheyeDet algorithm introduces distortion-shape matching, incorporating irregular quadrilateral bounding boxes, as shown in Figure 1, to enhance accuracy by 74.87% compared to YOLOv3's 68.92% [15]. While excelling in fisheye image precision, these models lag in model diversity, dataset availability, and hardware optimization compared to their perspective counterparts.

Complementing algorithmic strides, tailored datasets for fisheye images advance the field. Noteworthy is the in-the-wild events for people detection and tracking from overhead fisheye cameras (WEPDTOF) dataset, which adapts bounding boxes to counter fisheye's rotational distortion [16]. Given the limited fisheye data, synthetic datasets are created through image processing. The VOC-360 dataset leverages PASCAL VOC 2012, employing equidistant projection-based transformation for fisheye-specific annotations, as shown in Figure 2 [17].

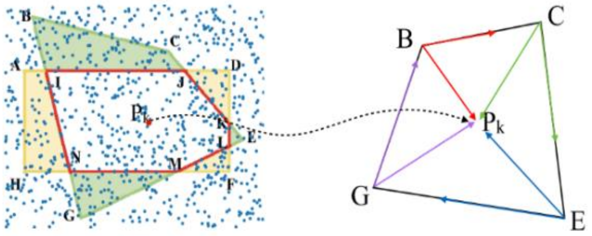


Figure 1. Distortion shape matching intersection over union (IoU) visualization and example [2]

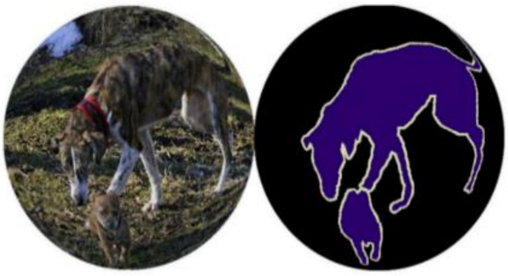


Figure 2. Sample VOC-360 image and segmentation mask [3]

2.3. Distortion correction methods

Various methods have emerged to address the challenge of fisheye object detection through distortion correction, encompassing techniques aimed at rectifying distortions prior to object detection. The longitude-latitude coordinate correction is employed in (1) for horizontal pixel correction, coupled with a straightforward vertical adjustment in (2). Here, $O(X_O, Y_O)$ represents the center of the image, $A(\mathrm{x}, \mathrm{y})$ denotes the original distorted point, and $B(\mathrm{u}, \mathrm{v})$ represents the coordinates of the corrected point, respectively. Additionally, in this context, R denotes the distance between point O and point A [18].

$$\frac{x-X_O}{u-X_O} = \frac{\sqrt{R^2-(y-Y_O)^2}}{R} \tag{1}$$

$$v = y \tag{2}$$

The spherical perspective projection correction, with a defined radius (r), mirrors the longitude-latitude method for accurate point correction as shown in (3) and (4), where $O(X_O, Y_O)$ represents the center of the fish-eye image, $P(X_P, Y_P)$ is the point in the corrected image, and point $m(X_m, Y_m)$ is the distorted point in the fish-eye image [18].

$$X_P = \frac{d(X_m-X_O)}{\sqrt{r^2-(X_m-X_O)^2-(Y_m-Y_O)^2}} \tag{3}$$

$$Y_P = \frac{d(Y_m-Y_O)}{\sqrt{r^2-(X_m-X_O)^2-(Y_m-Y_O)^2}} \tag{4}$$

Radial and tangential distortion correction, incorporated in OpenCV, employs a quintic polynomial for precise correction [19]. This function involves nine correction parameters $(c_1, c_2, c_3, c_4, c_5, a_1, a_2, a_3, a_4)$ that need to be constrained after setting a viewing angle (θ). The radial (ϕ') and tangential (θ') deformation formulas are provided in (5), (6), and (7) to correct the distortion. In these equations, ϕ' is a function of the radius (r), which is the distance from the center of the fisheye image. Additionally, θ' is a function of θ, representing the viewing angle. Parameter a_5 is calculated as shown in (7). These equations collectively describe the radial and tangential distortion correction process [20].

$$\phi' = c_1 r + c_2 r^2 + c_3 r^3 + c_4 r^4 + c_5 r^5 \tag{5}$$

$$\theta' = a_1 \theta + a_2 \theta^2 + a_3 \theta^3 + a_4 \theta^4 + a_5 \theta^5 \tag{6}$$

$$a_5 = \frac{(1-a_1-2\pi a_2-4\pi^2 a_3-8\pi^3 a_4)}{16\pi^4} \tag{7}$$

Generating panoramic images from fisheye footage introduces innovative solutions, including methods that utilize a coordinate conversion formula to remap each pixel, resulting in the equirectangular coordinates of a panorama [21]. The automation of distortion correction employs various methods, with template calibration being a prevalent approach. This technique, exemplified by chessboard and elliptical templates, automates the acquisition of correction parameters through image detection and corner identification [3], [18]. More complex scenarios are addressed using intricate algorithms involving neural networks or complex models [18].

Image reconstruction techniques are also employed, such as the fisheye GAN (FEGAN)-based fisheye image rectification network with a flow prediction and warping module [22]. An alternative approach by Beijing Jiaotong University utilizes image flow for separate structure correction and content reconstruction, enhancing image quality and accuracy [23]. These automated methods encompass diverse correction approaches and algorithms, contributing to enhanced image fidelity and accuracy. Furthermore, all the discussed correction methods are invaluable for synthetic dataset generation, enabling artificial image distortion and congruent bounding box adjustment.

2.4. Region splitting and composite images

This section delves into innovative approaches for addressing distortion in fisheye images by prioritizing representation over complete correction. These strategies involve different coordinate systems, image splitting, and projection techniques. For instance, composite images generated through equidistant mapping enable region-specific detectors and subsequent synthesis of detection results [24]. Techniques like cubic box mapping and icosahedral remapping offer unique perspectives, aiming for minimal distortion through multi-plane or spherical projections [25], [26]. These methods contribute to the development of versatile video monitoring systems, where the trade-off between correction and adaptability plays a crucial role. The combination of these techniques and the exploration of various approaches will guide the balance between accurate detection and computational efficiency while accommodating different camera setups and environments.

2.5. Intended output of the study

To address the existing gap, this research highlights the limitations of confined testing of distortion mitigation methods within their specific domains. To bridge this void, the study pioneers the development of an integrated framework that combines these techniques, assessing their collective impact on both speed and accuracy. Central to this exploration is the creation of a real-time monitoring system using a fish-eye camera, which is challenged by image distortion that negatively affects the precision of object detection algorithms. As a result, the study's objectives encompass the implementation of various distortion-handling methods, utilizing Python, OpenCV, and third-party libraries. These methods are then applied to an object detection algorithm, enabling a comprehensive assessment of accuracy and speed outcomes to identify the optimal configuration for video object detection. An additional aim is the development of a desktop application that not only visualizes the rectified feed but also incorporates real-time object detection and tracking functionalities. Through this collaborative effort, the research aims to establish a robust foundation for enhancing the efficiency and precision of fisheye camera setups, providing valuable insights to researchers and practitioners in the field.

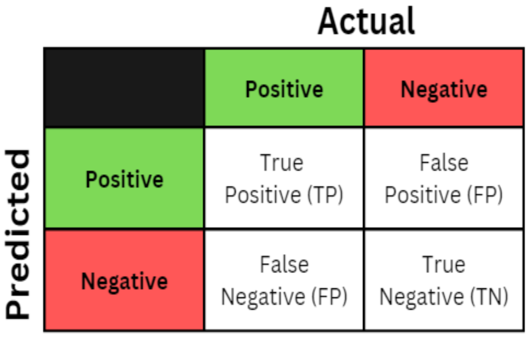
3. METHODOLOGY

3.1. Theoretical framework

Addressing fisheye distortion involves three key areas: preprocessing, training, and the detection algorithm itself. These areas offer numerous configurations to mitigate distortion effects. In the preprocessing stage, options include remapped images, distortion correction, or the generation of composite images. The training step encompasses choices between perspective datasets, real fisheye datasets, or synthetically distorted datasets. Algorithm selection ranges from state-of-the-art perspective models to custom-made fisheye detection models. Exploring various combinations of these options is crucial to identifying the optimal approach for fisheye object detection.

Different areas of distortion mitigation require distinct evaluation methodologies. In the context of fisheye object detectors, we measure accuracy using the mAP metric, while speed is quantified in terms of frames per second (FPS) or inference time. These metrics are then compared with the results obtained by state-of-the-art or fisheye detectors [15]. The computation of mAP involves several key components, including the confusion matrix, recall, IoU, and precision.

The confusion matrix, as illustrated in Figure 3, consists of four essential attributes. It defines true positives (TP) as instances where the model correctly predicts a label matching the ground truth, while true negatives (TN) are situations where both the prediction and the ground truth are absent. False positives (FP) occur when the model predicts an incorrect label, and false negatives (FN) emerge when the model fails to detect an object present in the ground truth. For object detectors, IoU assesses the bounding box overlap between predictions and the ground truth. This concept significantly impacts the outcome of the confusion matrix, as shown in Figure 4 and formulated in (8). Precision and recall, as outlined in (9) and (10), respectively, are then calculated [27].



		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FP)	True Negative (TN)

Figure 3. Confusion matrix

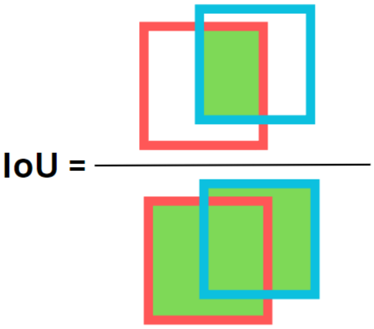


Figure 4. The IoU

$$\text{IoU} = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{8}$$

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

Precision quantifies the accuracy of the model's positive predictions, while recall measures the model's ability to correctly identify positives within the image. To calculate the average precision (AP), precision-recall curves are plotted by varying IoU thresholds. The AP contributes to the computation of the mAP using (11) and following the block diagram in Figure 5 [27].

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{11}$$

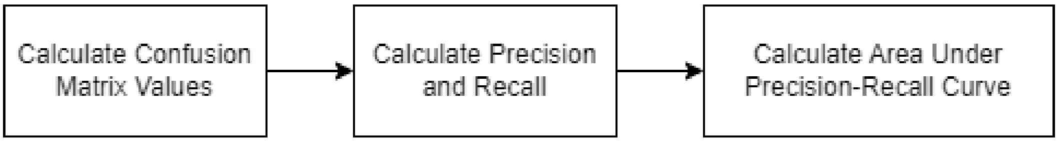

Calculate Confusion Matrix Values
Calculate Precision and Recall
Calculate Area Under Precision-Recall Curve

Figure 5. Average precision block diagram

For distortion correction and remapping techniques, runtime speed measurement is complemented by qualitative methods or reprojection error calculations, which evaluate the effectiveness of the correction [18], [19], [28]. In neural network-based distortion correction, image similarity is assessed through peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) scores. These metrics measure the resemblance of images against original perspective and synthetic fisheye images within synthetic datasets [29]. Evaluating correction results against ground truth perspective images also gauges network effectiveness [22]. While reprojection error or similarity scores provide insights into image clarity, some methodologies stop at this stage. Hence, integrating these methods with object detection algorithms reveals their impact on overall mAP enhancement.

The study categorizes the myriad methods into three main segments: input preprocessing, training datasets, and object detection, with an optional fourth segment for post-processing composite images. Evaluations involve combinations of preprocessing, object detection, and training set choices. For example, one configuration includes fisheye input undergoing a latitude-longitude correction, followed by training YOLOv3 on the VOC-360 dataset. Another approach divides fisheye images into tangent images, which are then processed by the RAPiD algorithm, trained on a mixed dataset of perspective and fisheye images. This structured framework, shown in Figure 6, facilitates the assessment and comparison of distortion correction methods, synthetic datasets, and mapping techniques in conjunction with state-of-the-art and fisheye object detection algorithms.

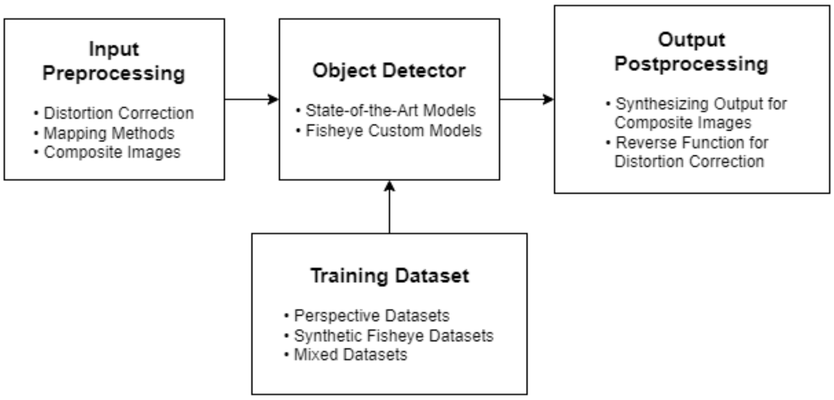


Figure 6. Proposed framework methodology

3.2. Distortion mitigation

3.2.1. Implementation

The study is conducted within the Python environment, leveraging essential packages for the development process. Notably, OpenCV is utilized for image processing functions, PyTorch facilitates neural network and algorithm implementation, and the spherical distortion package is harnessed for generating tangent images [19], [30], [31]. The implementation follows a structured approach, encompassing the integration of distortion mitigation methods with object detection algorithms. Figure 7 provides an overview of the methodology, illustrating the key steps.

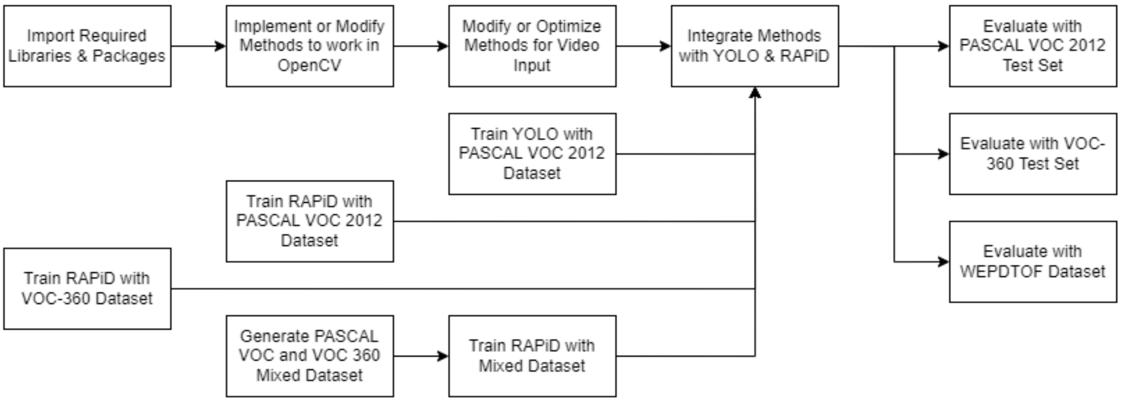

Import Required Libraries & Packages
Implement or Modify Methods to work in OpenCV
Modify or Optimize Methods for Video Input
Integrate Methods with YOLO & RAPiD
Evaluate with PASCAL VOC 2012 Test Set
Train YOLO with PASCAL VOC 2012 Dataset
Evaluate with VOC-360 Test Set
Train RAPiD with PASCAL VOC 2012 Dataset
Evaluate with WEPDTOF Dataset
Train RAPiD with VOC-360 Dataset
Generate PASCAL VOC and VOC 360 Mixed Dataset
Train RAPiD with Mixed Dataset

Figure 7. Methodology block diagram

In pursuit of the optimal combination, the study undertakes a comprehensive exploration of diverse distortion mitigation methods, encompassing longitude-latitude, perspective, cylindrical, panorama, and radial-tangential remapping techniques. Furthermore, the research delves into distortion correction neural networks such as FEGAN and progressively complementary network (PCN), as well as the tangent image method.

When dealing with video footage or streams, where fixed pixel resolution and camera parameters are expected, the system strategically enhances computational efficiency by pre-generating fixed maps of new x and y coordinates during initialization. These maps are then harnessed in conjunction with OpenCV's remap function to effectively implement distortion correction [19]. For popular techniques like longitude-latitude, perspective, and cylindrical projection, the planar grid methodology is employed for map generation. In the case of radial and tangential distortion correction, the integration leverages OpenCV's in-built functions coupled with default camera parameters [19], [32], [33]. Panorama generation takes a modified formulaic approach for map generation, utilizing OpenCV's remap function for seamless conversion [21], [34]. Distortion correction through neural networks such as FEGAN and PCN, alongside the tangent image method, necessitates framework adaptation for video input, as well as adjustments to postprocessing functions to ensure compatibility with YOLOv3 and RAPiD's bounding box annotations.

3.2.2. Integration

After implementing and adapting the distortion mitigation methods for video input, the study seamlessly integrates them with the YOLOv3 and RAPiD object detection algorithms for comprehensive testing and evaluation. To facilitate thorough examination, a simulated fisheye module is introduced, leveraging VOC-360's formula to illustrate the process using a regular camera. Notably, the fisheye video input undergoes panorama generation prior to tangent image creation, as the panorama image serves as a prerequisite input. These methods are harmoniously integrated with object detection algorithms, encompassing 1 checkpoint of YOLOv3 and 3 checkpoints of RAPiD, each utilizing distinct datasets: perspective, fisheye, and mixed. This integration results in a matrix of 36 distinct combinations, facilitating meticulous evaluation and testing. A comprehensive program flow diagram detailing the methodology is depicted in Figure 8.

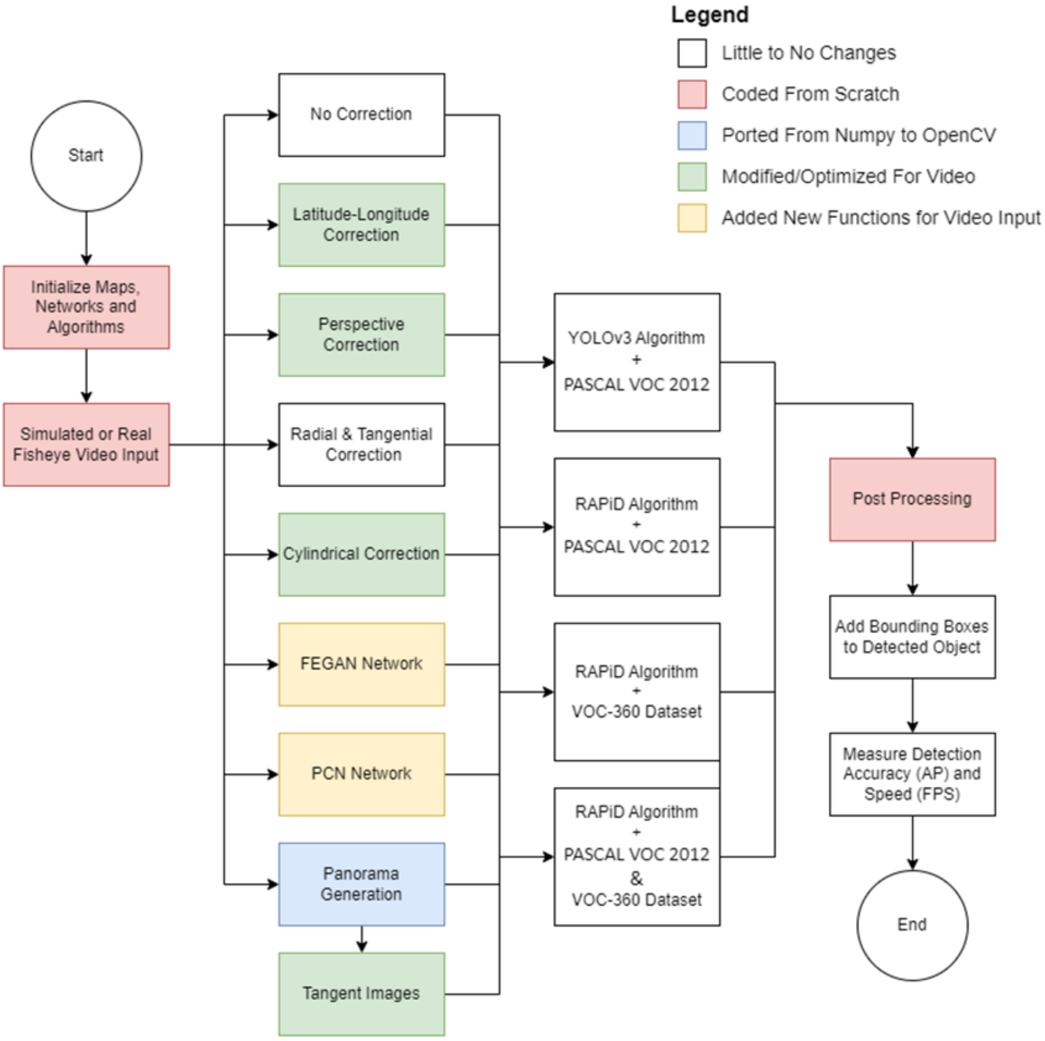

Legend
Little to No Changes
Coded From Scratch
Ported From Numpy to OpenCV
Modified/Optimized For Video
Added New Functions for Video Input
Start
Initialize Maps, Networks and Algorithms
Simulated or Real Fisheye Video Input
No Correction
Latitude-Longitude Correction
Perspective Correction
Radial & Tangential Correction
Cylindrical Correction
FEGAN Network
PCN Network
Panorama Generation
Tangent Images
YOLOv3 Algorithm + PASCAL VOC 2012
RAPiD Algorithm + PASCAL VOC 2012
RAPiD Algorithm + VOC-360 Dataset
RAPiD Algorithm + PASCAL VOC 2012 & VOC-360 Dataset
Post Processing
Add Bounding Boxes to Detected Object
Measure Detection Accuracy (AP) and Speed (FPS)
End

Figure 8. Program flow diagram

3.3. Training and evaluation

Training is conducted using both the PASCAL VOC dataset and its fisheye-augmented counterpart, VOC-360, offering access to both perspective and fisheye images. YOLOv3 is exclusively trained on the PASCAL VOC dataset, while RAPiD undergoes training using variations of both the PASCAL VOC and VOC-360 datasets. Evaluation involves employing the PASCAL VOC test set, VOC-360 test set, and the WEPDTOF dataset, details of which are provided in Table 1. Given the focus on dataset performance differences, 20,000 training iterations are executed for each configuration. Distortion mitigation methods undergo testing using different test sets, with accuracy quantified using AP0.5 to ensure consistency with the evaluation metric of the PASCAL VOC dataset.

Table 1. Datasets for evaluation

Test set	Total images	Total labels	Description
PASCAL VOC	569	1043	Regular image dataset
VOC 360	651	1111	PASCAL VOC synthetic dataset
WEPDTOF	10463	93364	Real fisheye dataset with rotation angle

To account for distortion correction's influence on bounding boxes, a remapping step is applied to bounding boxes before evaluation. In this study, remapping involves utilizing bounding box edges and a reverse function to ensure accurate evaluation. Speed evaluation encompasses pre-process time (t_1), inference time (t_2), post-process time (t_3), and frames per second (FPS), calculated using (12). The comprehensive evaluation entails ranking combinations based on speed, accuracy, and a combined aggregate. This evaluation approach incorporates a weighted scoring mechanism, with a specific emphasis on relevant dataset performance, as illustrated by (13) and (14).

$$FPS = \frac{1}{t_1+t_2+t_3} \tag{12}$$

$$S = [(AP \times 0.5) + [MIN(FPS, 30)/30 \times 0.5]] \times 100\% \tag{13}$$

$$S_{Overall} = [0.7(S_{WEPDTOF}) + 0.2(S_{VOC360}) + 0.1(S_{PASCAL})] \times 100\% \tag{14}$$

The methodology presents a systematic and rigorous framework, systematically exploring diverse combinations of distortion correction methods and algorithm setups. A threshold recommendation score of 61.333 is established, derived by substituting AP values of 70 for PASCAL and VOC-360, 50 for WEPDTOF, and 20 for all test set FPS into (13), and using (14) to obtain the final value. This score aids in determining optimal strategies for fisheye object detection, striking a balance between speed and accuracy considerations.

4. RESULTS AND DISCUSSION

The evaluation process entailed assessing the accuracy and speed of each combination using the PASCAL VOC test set, VOC-360 test set, and the WEPDTOF dataset. Combinations were classified according to their speed, accuracy, and overall performance. This process facilitated the identification of optimal combinations within each category and dataset. This chapter presents the evaluation outcomes, contrasting the highest-performing combinations across different datasets and categories. Furthermore, the strengths, weaknesses, and potential real-world applications of each combination are thoroughly discussed.

4.1. Overall results

The results, presented in Table 2, reveal variations in optimal combinations across datasets. Notably, the no correction and YOLOv3 combination exhibited the highest accuracy on the PASCAL test set, while the pairing of FEGAN and YOLOv3 outperformed others on the VOC-360 test set, which encompasses synthetic fisheye images. In contrast, the WEPDTOF dataset, featuring top-view angle fisheye images, demonstrated lower accuracy across all combinations due to variations in object appearance and perspective compared to the training sets.

Table 2. The best combination results in accuracy, speed, and score

Test set	Accuracy		Speed		Score	
	Combination	AP0.5	Combination	FPS	Combination	Score
PASCAL VOC	No Correction + YOLOv3-PASCAL	0.776	Panorama + YOLOv3-PASCAL	17.331	No Correction + YOLOv3-PASCAL	50.662
VOC-360	FEGAN + YOLOv3-PASCAL	0.769	Panorama + YOLOv3-PASCAL	17.391	No Correction + YOLOv3-PASCAL	49.723
WEPDTOF	Longitude-Latitude + YOLOv3-PASCAL	0.159	Panorama + YOLOv3-PASCAL	17.073	Panorama + YOLOv3-PASCAL	30.370

In terms of speed, the Panorama and YOLOv3 combination consistently achieved the highest frames per second (FPS) across all datasets. This outcome can be attributed to the elongated shape of panorama images, aligning well with YOLOv3's input specifications and thus leading to swifter processing. Notably, on the WEPDTOF dataset, the longitude-latitude and YOLOv3 combination attained the highest accuracy with an AP0.5 of 0.159. However, its score was surpassed by the panorama and YOLOv3 combination with a score of 30.370, underscoring the significant influence of speed on overall object detection performance.

Subsequent sections delve into comprehensive analyses of results and discussions for each dataset: PASCAL VOC, VOC-360, and WEPDTOF. These sections meticulously scrutinize combinations of distortion mitigation methods, considering both accuracy and speed. Furthermore, the study explores how each method impacts dataset performance and sheds light on potential sources of performance discrepancies.

4.2. Test set results

4.2.1. PASCAL VOC test set

Table 3 displays the scores for each individual combination, while Table 4 showcases the top 10 scores from the PASCAL test set alongside their corresponding AP0.5 and FPS values. Notably, the YOLOv3-PASCAL algorithm occupies the first eight positions, aligning with expectations given the dataset's characteristics. However, RAPiD-MIXED and RAPiD-VOC-360 combinations emerge as standouts among RAPiD variants, securing the 9th and 10th spots, respectively. In contrast, the panorama correction method, despite its higher speed, occupies the 8th position due to diminished accuracy, thus illustrating the inherent trade-off between speed and precision in this approach.

Discrepancies in performance between RAPiD and YOLOv3 on the PASCAL test set may stem from dataset characteristics. RAPiD's flexible bounding box strategy might yield predictions that are less aligned with the ground truth, especially when ground truth bounding boxes maintain a fixed 0-degree orientation. Conversely, YOLOv3's fixed grid system could be better suited for this dataset, consistently predicting rectangular bounding boxes that closely match the ground truth.

The top three performers in terms of accuracy are YOLOv3-PASCAL without correction, YOLOv3-PASCAL with cylindrical correction, and YOLOv3-PASCAL with longitude-latitude correction, as depicted in Table 4. Distortion correction methods could introduce reverse distortion, leading to relatively lower accuracy compared to the no-correction approach. However, the cylindrical and longitude-latitude methods exhibit relatively favorable accuracy, suggesting they might introduce less reverse distortion or better preserve image details and shape, thus enhancing object detection accuracy.

Table 3. PASCAL test set score results

Correction	Algorithm				Avg. correction score
	YOLOv3-PASCAL	RAPiD-PASCAL	RAPiD-VOC360	RAPiD-MIXED	
No Correction	50.662	17.603	17.723	17.827	25.954
Longitude-Latitude	45.992	14.752	14.989	15.474	22.802
Perspective	37.481	13.254	13.351	13.641	19.431
Cylindrical	47.792	15.244	15.826	15.717	23.645
Radial + Tangential	34.232	9.619	9.487	10.060	15.850
FEGAN	38.262	9.792	9.518	9.949	16.880
PCN	39.375	9.879	10.730	10.689	17.668
Panorama	32.355	11.672	11.869	11.788	16.921
Tangent Images	7.409	5.508	5.585	5.598	6.025
Avg. Algorithm Score	37.062	11.925	12.120	12.305	

Table 4. Top 10 scores from PASCAL test set

Rank	Correction	Algorithm	AP0.5	FPS	Score
1	No Correction	YOLOv3-PASCAL	0.776	7.117	50.662
2	Cylindrical	YOLOv3-PASCAL	0.725	6.925	47.792
3	Longitude-Latitude	YOLOv3-PASCAL	0.689	6.925	45.992
4	PCN	YOLOv3-PASCAL	0.644	4.305	39.375
5	FEGAN	YOLOv3-PASCAL	0.634	3.937	38.262
6	Perspective	YOLOv3-PASCAL	0.517	6.978	37.481
7	Radial + Tangential	YOLOv3-PASCAL	0.508	5.299	34.232
8	Panorama	YOLOv3-PASCAL	0.069	17.331	32.355
9	No Correction	RAPiD-MIXED	0.129	6.826	17.827
10	No Correction	RAPiD-VOC360	0.126	6.854	17.723

Qualitative insights from Figure 9 reveal that while YOLOv3-PASCAL generally maintains consistent object detection, there are instances where all YOLO combinations struggle to detect objects that RAPiD combinations successfully identify. This divergence might result from YOLO's reliance on a fixed grid system, which could be less effective in certain image types or object contexts as shown in Figure 9(a). In contrast, RAPiD's adaptable bounding box approach might confer an advantage in accommodating diverse image and object scenarios, despite its relatively lower overall accuracy score as shown in Figure 9(b).

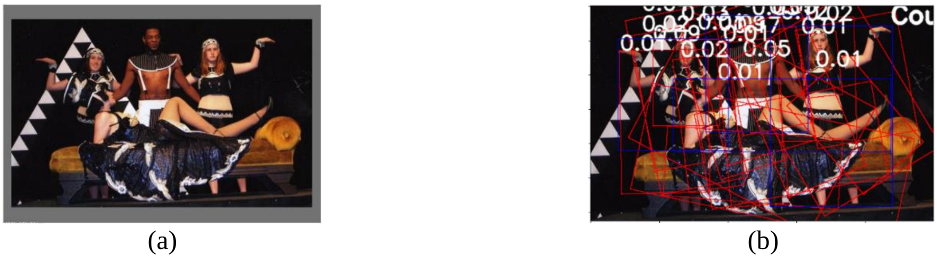

(a) (b)

Figure 9. Qualitative results of different algorithms for PASCAL test set of (a) No Correction + YOLOv3-PASCAL and (b) No Correction + RAPiD-MIXED

4.2.2. VOC-360 test set results

Like the previous section, two tables, namely Tables 5 and 6, present the scores for each individual combination and the top 10 combinations from the VOC-360 test set, accompanied by their corresponding AP0.5 and FPS values. Within the domain of speed, the panorama + YOLOv3 combination maintains its dominance. However, this section contextualizes the significance of speed within the score. Despite FEGAN + YOLOv3 achieving a remarkable accuracy of 76.9%, its score is tempered by its slower speed, which is only slightly more than half of the leading combination's speed: no correction + YOLOv3. The no correction + YOLOv3 combination claims the top position in the score, showcasing its adept balance between speed and accuracy for this dataset.

In terms of accuracy, the leading combinations are FEGAN with YOLOv3-PASCAL, No Correction with YOLOv3-PASCAL, and longitude-latitude with YOLOv3-PASCAL, as depicted in Table 6. Significantly, the YOLOv3-PASCAL algorithm, which demonstrated excellence on the PASCAL test set, consistently exhibits robust accuracy on the VOC-360 test set, highlighting its versatility across diverse datasets. Notably, FEGAN, a generative adversarial network for image enhancement, pairs effectively with YOLOv3-PASCAL, suggesting the potential of image enhancement techniques to enhance object detection. The disparity in accuracy between corrected and uncorrected images has reduced for the VOC-360 test set compared to the PASCAL test set. While only the no correction and cylindrical methods achieved an accuracy of 70% or higher on the PASCAL test set, the FEGAN, no correction, longitude-latitude, cylindrical, and radial + tangential methods attained the same threshold on the VOC-360 dataset.

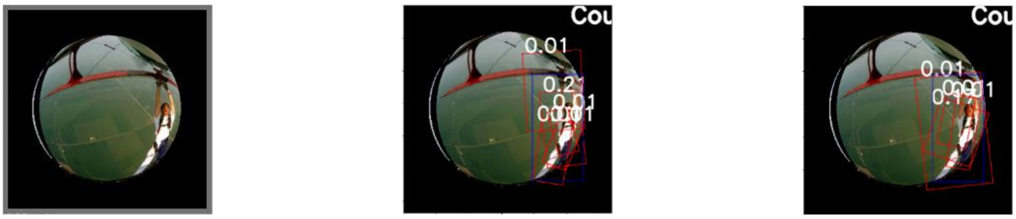
Table 5. VOC-360 test set score results

Correction	Algorithm				Avg. correction score
	YOLOv3-PASCAL	RAPiD-PASCAL	RAPiD-VOC360	RAPiD-MIXED	
No Correction	49.723	13.484	13.942	13.892	22.760
Longitude-Latitude	49.100	12.758	13.023	13.039	21.980
Perspective	46.155	12.483	12.633	12.619	20.973
Cylindrical	48.032	12.732	13.052	13.110	21.731
Radial + Tangential	44.748	8.868	9.248	9.156	18.005
FEGAN	44.978	8.435	8.439	8.715	17.642
PCN	34.190	8.587	9.040	9.024	15.210
Panorama	32.356	11.400	11.416	11.400	16.643
Tangent Images	8.691	5.508	5.898	5.565	6.415
Avg. Algorithm Score	39.775	10.473	10.743	10.724	

Table 6. Top 10 scores from the VOC-360 test set

Rank	Correction	Algorithm	AP0.5	FPS	Score
1	No Correction	YOLOv3-PASCAL	0.755	7.184	49.723
2	Longitude-Latitude	YOLOv3-PASCAL	0.751	6.930	49.100
3	Cylindrical	YOLOv3-PASCAL	0.729	6.949	48.032
4	Perspective	YOLOv3-PASCAL	0.690	6.993	46.155
5	FEGAN	YOLOv3-PASCAL	0.769	3.917	44.978
6	Radial + Tangential	YOLOv3-PASCAL	0.708	5.609	44.748
7	PCN	YOLOv3-PASCAL	0.540	4.314	34.190
8	Panorama	YOLOv3-PASCAL	0.067	17.391	32.356
9	No Correction	RAPiD-VOC360	0.051	6.835	13.942
10	No Correction	RAPiD-MIXED	0.050	6.835	13.892

Qualitative insights into the VOC-360 test set reveal challenges in detecting objects within synthetic fisheye images compared to regular images in the PASCAL test set, particularly at the image edges with higher distortion. Figure 10 illustrates that RAPiD-based combinations can detect some edge objects that YOLOv3 combinations miss, Figure 10(a) shows YOLOv3-PASCAL, Figure 10(b) shows no correction + RAPiD-MIXED, and Figure 10(c) shows FEGAN + RAPiD-MIXED. This suggests RAPiD's potential robustness to fisheye distortion in specific scenarios for this dataset.



(a) (b) (c)

Figure 10. Qualitative results of different algorithms for VOC-360 test set of (a) YOLOv3-PASCAL, (b) no correction + RAPiD-MIXED, and (c) FEGAN + RAPiD-MIXED

4.2.3. WEPDTOF dataset results

The extensive WEPDTOF dataset is segmented into 16 sections for evaluation, with Table 7 delineating superior combinations for each section across accuracy, speed, and score. Notably, accuracy varies across different sections of the WEPDTOF dataset. Longitude-latitude correction with YOLOv3 excels in WEPDTOF-CS and WEPDTOF-CC but lags in WEPDTOF-SG. Exceptional results are observed with cylindrical correction and YOLOv3 in WEPDTOF-JS2, while no correction with YOLOv3 performs well in sections like WEPDTOF-K and WEPDTOF-W.

While YOLOv3 consistently outperforms RAPiD across datasets, most top-performing combinations involve correction methods or RAPiD, emphasizing their potential for enhancing detection accuracy in complex fisheye datasets like WEPDTOF. Interestingly, instances favoring no correction + YOLOv3 suggest the effectiveness of simpler approaches in specific scenarios.

Table 7. WEPDTOF breakdown of best combination results for accuracy, speed, and score

Test Set	Accuracy		Speed		Score	
	Combination	AP 0.5	Combination	FPS	Combination	Score
WEPDTOF-CC	Longitude-Latitude + YOLOv3-PASCAL	0.055	Panorama + YOLOv3-PASCAL	17.036	Panorama + YOLOv3-PASCAL	28.493
WEPDTOF-CS	Longitude-Latitude + YOLOv3-PASCAL	0.301	Panorama + YOLOv3-PASCAL	17.452	Panorama + YOLOv3-PASCAL	35.137
WEPDTOF-ES	No Correction + RAPiD-MIXED	0.388	Panorama + YOLOv3-PASCAL	17.483	Panorama + YOLOv3-PASCAL	40.638
WEPDTOF-EX	No Correction + YOLOv3-PASCAL	0.147	Panorama + YOLOv3-PASCAL	17.271	Panorama + YOLOv3-PASCAL	28.785
WEPDTOF-EXS	No Correction + YOLOv3-PASCAL	0.079	Panorama + YOLOv3-PASCAL	17.241	Panorama + YOLOv3-PASCAL	28.786
WEPDTOF-IO	Perspective + YOLOv3-PASCAL	0.204	Panorama + YOLOv3-PASCAL	17.331	Panorama + YOLOv3-PASCAL	30.285
WEPDTOF-JS	Cylindrical + YOLOv3-PASCAL	0.241	Panorama + YOLOv3-PASCAL	17.422	Panorama + YOLOv3-PASCAL	29.936
WEPDTOF-JS2	Cylindrical + YOLOv3-PASCAL	0.528	Panorama + YOLOv3-PASCAL	17.331	Cylindrical + YOLOv3-PASCAL	37.315
WEPDTOF-K	No Correction + YOLOv3-PASCAL	0.203	Panorama + YOLOv3-PASCAL	15.291	Panorama + YOLOv3-PASCAL	30.534
WEPDTOF-LO	Perspective + YOLOv3-PASCAL	0.070	Panorama + YOLOv3-PASCAL	17.452	Panorama + YOLOv3-PASCAL	29.087
WEPDTOF-LO2	Perspective + YOLOv3-PASCAL	0.193	Panorama + YOLOv3-PASCAL	17.452	Panorama + YOLOv3-PASCAL	29.337
WEPDTOF-PS	Perspective + RAPiD-MIXED	0.125	Panorama + YOLOv3-PASCAL	16.207	Panorama + YOLOv3-PASCAL	27.012
WEPDTOF-RS	Perspective + YOLOv3-PASCAL	0.279	Panorama + YOLOv3-PASCAL	16.722	Panorama + YOLOv3-PASCAL	28.571
WEPDTOF-SG	Longitude-Latitude + YOLOv3-PASCAL	0.298	Panorama + YOLOv3-PASCAL	17.361	Panorama + YOLOv3-PASCAL	28.935
WEPDTOF-TS	Panorama + YOLOv3-PASCAL	0.070	Panorama + YOLOv3-PASCAL	17.391	Panorama + YOLOv3-PASCAL	32.486
WEPDTOF-W	No Correction + YOLOv3-PASCAL	0.118	Panorama + YOLOv3-PASCAL	16.722	Panorama + YOLOv3-PASCAL	28.271

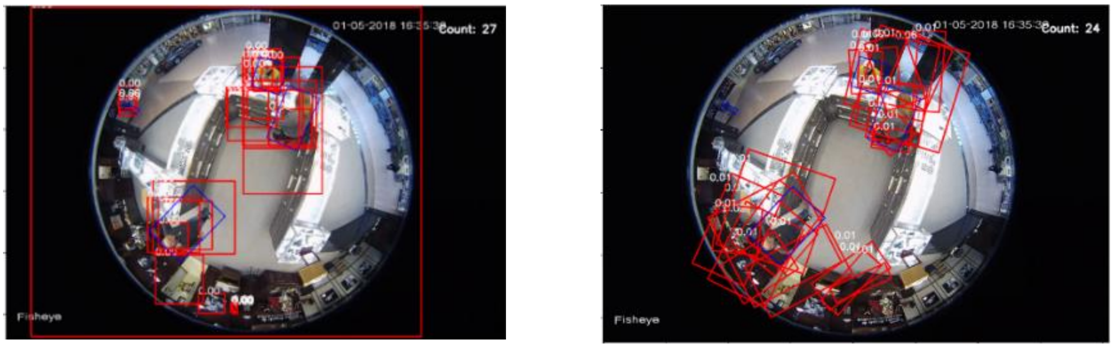
In terms of Table 8 scores and Table 9 rankings, panorama and YOLOv3 emerge as the frontrunners, propelled by their impressive speed. Similarly, longitude-latitude + YOLOv3, no correction + YOLOv3, and cylindrical + YOLOv3 all perform well, as shown in Figure 11. Correction methods and YOLOv3 are the dominant players among the top-performing combinations. Interestingly, RAPiD combinations, such as no correction + RAPiD, achieve competitive AP0.5 scores, albeit slightly lower than the leaders. The intriguing variation in RAPiD's performance across training datasets is noteworthy. Notably, no correction + RAPiD trained on PASCAL outperforms expectations, demonstrating its adaptability beyond its original dataset. However, these minor differences suggest that further experimentation is necessary to obtain definitive insights.

Table 8. WEPDTOF dataset score results

Correction	Algorithm				Avg. correction score
	YOLOv3-PASCAL	RAPiD-PASCAL	RAPiD-VOC360	RAPiD-MIXED	
No Correction	19.000	14.763	14.053	14.759	15.644
Longitude-Latitude	19.287	13.247	13.292	13.286	14.778
Perspective	17.599	14.009	12.639	13.588	14.459
Cylindrical	18.887	12.878	12.372	13.039	14.294
Radial + Tangential	10.475	10.001	9.700	10.501	10.169
FEGAN	7.326	9.732	8.349	9.134	8.635
PCN	7.054	7.474	7.203	7.244	7.244
Panorama	30.370	12.143	11.623	11.763	16.475
Tangent Images	7.970	5.780	5.997	5.806	6.388
Avg. Algorithm Score	15.330	11.114	10.581	11.013	

Table 9. Top 10 Scores from WEPDTOF dataset

Rank	Correction	Algorithm	AP0.5	FPS	Score
1	Panorama	YOLOv3-PASCAL	0.038	17.073	30.370
2	Longitude-Latitude	YOLOv3-PASCAL	0.159	6.676	19.287
3	No Correction	YOLOv3-PASCAL	0.142	6.971	19.000
4	Cylindrical	YOLOv3-PASCAL	0.150	6.766	18.887
5	Perspective	YOLOv3-PASCAL	0.125	6.789	17.599
6	No Correction	RAPiD-PASCAL	0.069	6.734	14.763
7	No Correction	RAPiD-MIXED	0.067	6.784	14.759
8	No Correction	RAPiD-VOC360	0.054	6.701	14.053
9	Perspective	RAPiD-PASCAL	0.062	6.519	14.009
10	Perspective	RAPiD-MIXED	0.052	6.576	13.588



(a)

(b)

Figure 11. Qualitative results of different algorithms for WEPDTOF dataset of (a) longitude-latitude + YOLOv3-PASCAL and (b) FEGAN + RAPiD-PASCAL

4.3. Additional speed results

The preceding section employed FPS results from a less powerful HP Pavilion computer, while this section explores improved speed outcomes achieved through tests on a more robust system, the Helios 300, equipped with an NVIDIA GeForce RTX 3060 GPU. The findings reveal significant speed enhancements across all methods, with more than half of them nearing the target of 30 FPS. The updated score tables, specifically Tables 10, 11, and 12, illustrate the enhanced performance resulting from increased FPS on the

Helios 300 setup. These improved scores correspond to higher rankings, emphasizing the pivotal role of robust computing in advancing fisheye object detection capabilities.

A closer analysis of these tables highlights revised overall score rankings, now incorporating the new FPS values from the Helios 300. Remarkably, the top three combinations for the PASCAL and VOC-360 datasets remain consistent, followed by a reshuffling of ranks that places greater emphasis on accuracy. Rankings for the WEPDTOF dataset experience a significant shift, except for ranks 3 and 8.

Table 10. HP Pavilion vs. Helios 300 overall score from the PASCAL test set

Rank	Correction	Algorithm	AP0.5	HP Pavilion		Helios 300		Δ Rank
				FPS	Score	FPS	Score	
1	No Correction	YOLOv3-PASCAL	0.776	7.117	50.662	43.860	88.800	-
2	Cylindrical	YOLOv3-PASCAL	0.725	6.925	47.792	38.168	86.250	-
3	Longitude-Latitude	YOLOv3-PASCAL	0.689	6.925	45.992	38.314	84.450	-
4	Perspective	YOLOv3-PASCAL	0.517	6.978	37.481	39.063	75.850	↑
5	PCN	YOLOv3-PASCAL	0.644	4.305	39.375	23.529	71.416	↓
6	No Correction	RAPiD-MIXED	0.129	6.826	17.827	37.175	56.450	↑
7	No Correction	RAPiD-VOC360	0.126	6.854	17.723	37.736	56.300	↑
8	No Correction	RAPiD-PASCAL	0.125	6.812	17.603	36.900	56.250	↑
9	Cylindrical	RAPiD-VOC360	0.094	6.676	15.826	33.557	54.700	↑
10	Cylindrical	RAPiD-MIXED	0.093	6.640	15.717	33.113	54.650	↑

Table 11. HP Pavilion vs. Helios 300 overall score from VOC-360 test set

Rank	Correction	Algorithm	AP0.5	HP Pavilion		Helios 300		Δ Rank
				FPS	Score	FPS	Score	
1	No Correction	YOLOv3-PASCAL	0.755	7.184	49.723	44.053	87.750	-
2	Longitude-Latitude	YOLOv3-PASCAL	0.751	6.930	49.100	38.314	87.550	-
3	Cylindrical	YOLOv3-PASCAL	0.729	6.949	48.032	38.610	86.450	-
4	Perspective	YOLOv3-PASCAL	0.690	6.993	46.155	39.683	84.500	-
5	PCN	YOLOv3-PASCAL	0.540	4.314	34.190	23.095	65.491	↑
6	Radial + Tangential	YOLOv3-PASCAL	0.708	5.609	44.748	15.361	61.002	-
7	FEGAN	YOLOv3-PASCAL	0.769	3.917	44.978	10.893	56.605	↓
8	Panorama	YOLOv3-PASCAL	0.067	17.391	32.356	72.464	53.370	-
9	No Correction	RAPiD-VOC360	0.051	6.835	13.942	37.175	52.550	-
10	No Correction	RAPiD-MIXED	0.050	6.835	13.892	37.175	52.500	-

Table 12. HP Pavilion vs. Helios 300 overall score from the WEPDTOF dataset

Rank	Correction	Algorithm	AP0.5	HP Pavilion		Helios 300		Δ Rank
				FPS	Score	FPS	Score	
1	Longitude-Latitude	YOLOv3-PASCAL	0.159	6.676	19.287	38.706	1	Longitude-Latitude
2	Cylindrical	YOLOv3-PASCAL	0.150	6.766	18.887	38.713	2	Cylindrical
3	No Correction	YOLOv3-PASCAL	0.142	6.971	19.000	40.696	3	No Correction
4	Perspective	YOLOv3-PASCAL	0.125	6.789	17.599	39.204	4	Perspective
5	No Correction	RAPiD-PASCAL	0.069	6.734	14.763	39.799	5	No Correction
6	No Correction	RAPiD-MIXED	0.067	6.784	14.759	39.546	6	No Correction
7	Perspective	RAPiD-PASCAL	0.062	6.519	14.009	34.601	7	Perspective
8	No Correction	RAPiD-VOC360	0.054	6.701	14.053	39.782	8	No Correction
9	Perspective	RAPiD-MIXED	0.052	6.576	13.588	34.735	9	Perspective
10	Longitude-Latitude	RAPiD-MIXED	0.046	6.550	13.286	34.393	10	Longitude-Latitude

While the scores from the HP Pavilion system fell short of the 61.333 threshold, the notably improved FPS values on the Helios 300 system have enabled the top four PASCAL combinations to successfully surpass this threshold using (14), as outlined in Table 13. These configurations longitude-latitude + YOLOv3-PASCAL, no correction + YOLOv3-PASCAL, Cylindrical + YOLOv3-PASCAL, and Perspective + YOLOv3-PASCAL are recommended for effective video object detection in challenging fisheye footage.

Table 13. Recommended combinations for video object detection

Rank	Correction	Algorithm	Overall score
1	Longitude-Latitude	YOLOv3-PASCAL	66.667
2	No Correction	YOLOv3-PASCAL	66.597
3	Cylindrical	YOLOv3-PASCAL	66.242
4	Perspective	YOLOv3-PASCAL	63.884

4.4. Correction breakdown and analysis

Previous sections demonstrated significant improvements in object detection accuracy using correction methods and remapping techniques in fisheye datasets. However, these approaches exhibit potential for further refinement, as they encounter accuracy challenges in specific scenarios. For instance, the PCN correction method can lead to data loss at image edges, affecting IoU results by truncating object portions. Similarly, during panorama-to-fisheye conversion, the remapping process might unintentionally truncate bounding boxes, impacting detection accuracy as shown in Figure 12.

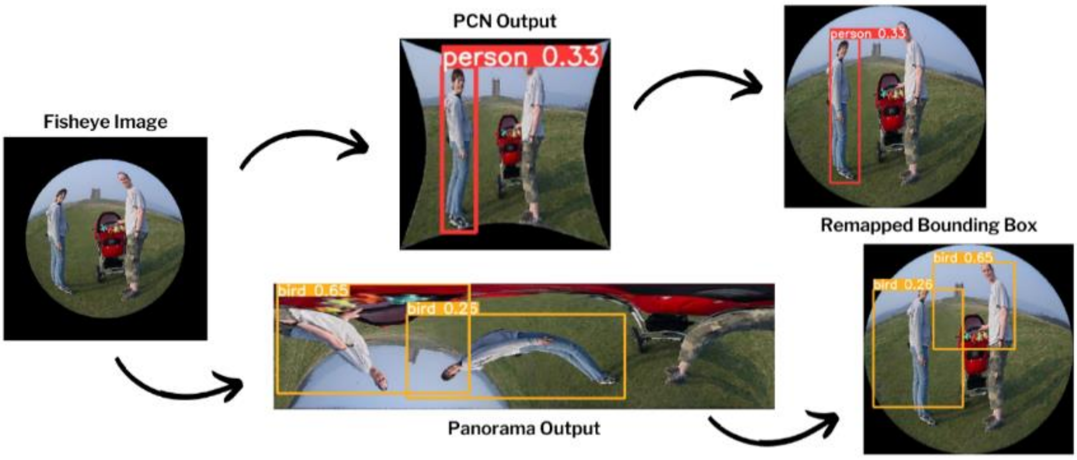


Figure 12. PCN and panorama correction and remapping process breakdown

Strategies aimed at effectively representing wrapped objects and exploring alternative remapping or distortion correction approaches could mitigate these issues. Another avenue for enhancement involves the generation of tangent images, which might result in fragmented bounding boxes for objects spanning segments, potentially affecting accuracy. Improved synthesis techniques to reconstruct original bounding boxes from fragments could alleviate this challenge. While correction methods and remapping offer substantial accuracy gains, addressing their limitations through refined processes could further elevate object detection accuracy in fisheye datasets.

4.5. The monitoring system

In parallel with the experiments conducted, a dedicated monitoring system desktop application was meticulously crafted to serve as a versatile platform, utilizing a wide array of correction methods and algorithms. Built upon the foundation of the YOLOv3 detect script, this application leverages the framework established in this study, seamlessly integrating distinct correction methodologies, including the option to employ the RAPiD algorithm.

With the capability to process images, videos, and camera inputs, including simulated fisheye data, this system provides a comprehensive environment for the evaluation, testing, and synergistic combination of the correction methods and algorithms investigated in this study. Significantly, it boasts the functionality to export corrected frames in various formats, streamlining the dissemination of results and insights while extending the capabilities of the existing YOLOv3 detect script.

This application has convincingly demonstrated its utility in experimental contexts, offering adaptability and customization for the evaluation of a diverse array of correction methods and algorithms. Its versatility in handling disparate inputs, including fisheye simulation, substantially broadens the scope of testing scenarios, thereby enhancing the comprehensive assessment of distortion correction techniques. Moreover, it is noteworthy that the qualitative image results featured in this paper were generated by said system.

5. CONCLUSION

This study provides a comprehensive evaluation of distortion correction and object detection methods for fisheye camera setups. It identifies YOLOv3 as a leading performer in accuracy and speed for undistorted footage, underlining its potential for real-world applications. Notably, the combination of longitude-latitude correction with the YOLOv3 object detector stands out as the best-performing configuration, achieving remarkable accuracy and surpassing the established threshold.

The development of a versatile desktop application further highlights the practicality of these methods. Looking ahead, several promising avenues for future research emerge. Firstly, there is room for fine-tuning object detectors, exploring advanced remapping techniques, and optimizing code to further enhance the performance of these methods. Additionally, extending the study to encompass different camera models, the integration of multiple cameras, and experimentation in diverse scenarios is of great significance.

This study, while primarily focused on improving object detection accuracy and speed, indirectly contributes to enhancing the overall capabilities of fisheye camera systems. Addressing the challenges of distortion correction, opens doors for researchers and practitioners to explore innovative solutions and applications, ultimately advancing the field of fisheye camera technology.

ACKNOWLEDGEMENTS

This research was made possible through a grant from the Engineering Research and Development for Technology (ERDT) program, which provided invaluable support and resources in conducting the study.

REFERENCES

[1] A. Oner, M. S. Albayrak, F. Guner, and I. M. Atakli, “An activity recognition application based on Markov decision process through fish eye camera,” in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, IEEE, Jun. 2019, pp. 251–258. doi: 10.1109/AIKE.2019.00050.

[2] T. Wang, C.-H. Liao, L.-H. Hsieh, A. W. Tsui, and H.-C. Huang, “People detection and tracking using a fisheye camera network,” in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, Dec. 2021, pp. 1–5. doi: 10.1109/VCIP53242.2021.9675451.

[3] F. Wang, K. Yu, J. Yang, Z. Liu, K. Su, and L. Xie, “A novel obstacle detection and tracking system using fisheye vision,” in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, IEEE, Jun. 2020, pp. 674–680. doi: 10.1109/ICAICA50127.2020.9182497.

[4] S. Oh-hara and A. Fujimori, “A Leader-follower formation control of mobile robots by position-based visual servo method using fisheye camera,” *ROBOMECH J.*, vol. 10, no. 1, 2023, doi: 10.1186/s40648-023-00268-6.

[5] O. Zia, J.-H. Kim, K. Han, and J. W. Lee, “360° panorama generation using drone mounted fisheye cameras,” in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, IEEE, Jan. 2019, pp. 1–3. doi: 10.1109/ICCE.2019.8661954.

[6] J. Newberry, “Virtual tour creation: Los Angeles, California,” PanoramicEye. Accessed: Sep. 02, 2022. [Online]. Available: http://www.panoramiceye.com/

[7] Europan Space Agency, “ISS virtual tour 1.0.,” ESA Multimedia. Accessed: Sep. 02, 2022. [Online]. Available: https://esamultimedia.esa.int/multimedia/virtual-tour-iss/

[8] N. Nicholaou, “Instant Google Street view,” Instant Google Street View. Accessed: Sep. 02, 2022. [Online]. Available: https://www.instantstreetview.com/

[9] YouTube, “Virtual reality,” YouTube. Accessed: Sep. 02, 2022. [Online]. Available: https://www.youtube.com/channel/UCzuqhhs6NWbgTzMuM09WKDQ

[10] P. Han, Y.-C. Tseng, and C.-M. Tsai, “Wide field of view lens design with uniform image illumination in capsule endoscope system,” *Microsyst. Technol.*, vol. 27, no. 4, pp. 1115–1122, Apr. 2021, doi: 10.1007/s00542-018-4104-y.

[11] M. Y. Shalaginov *et al.*, “Single-element diffraction-limited fisheye meta lens,” *Nano Lett.*, vol. 20, no. 10, pp. 7429–7437, Oct. 2020, doi: 10.1021/acs.nanolett.0c02783.

[12] Y. Indulkar, “Alleviation of COVID by means of social distancing & face mask detection using YOLO V4,” in *2021 International Conference on Communication information and Computing Technology (ICCICT)*, IEEE, Jun. 2021, pp. 1–8. doi: 10.1109/ICCICT50803.2021.9510168.

[13] X. Lei, B. Sun, J. Peng, and F. Zhang, “Fisheye image object detection based on an improved YOLO V3 algorithm,” in *2020 Chinese Automation Congress (CAC)*, IEEE, Nov. 2020, pp. 5801–5805. doi: 10.1109/CAC51589.2020.9326859.

[14] Q. N. Minh, B. Le Van, C. Nguyen, A. Le, and V. D. Nguyen, “ARPD: anchor-free rotation-aware people detection using Topview fisheye camera,” in *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, Nov. 2021, pp. 1–8. doi: 10.1109/AVSS52988.2021.9663768.

[15] T. Li, G. Tong, H. Tang, B. Li, and B. Chen, “FisheyeDet: a self-study and contour-based object detector in fisheye images,” *IEEE Access*, vol. 8, pp. 71739–71751, 2020, doi: 10.1109/ACCESS.2020.2987868.

[16] M. O. Tezcan, Z. Duan, M. Cokbas, P. Ishwar, and J. Konrad, “WEPDTOF: a dataset and benchmark algorithms for in-the-wild people detection and tracking from overhead fisheye cameras,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Jan. 2022, pp. 1381–1390. doi: 10.1109/WACV51458.2022.00145.

[17] J. Fu, I. V Bajić, and R. G. Vaughan, “Datasets for face and object detection in fisheye images,” *Data Br.*, vol. 27, p. 104752, Dec. 2019, doi: 10.1016/j.dib.2019.104752.

[18] Y. Wu and L. Zhang, “Spatio-temporal fish-eye image processing based on neural network,” in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, May 2020, pp. 356–362. doi: 10.1109/ICCCS49078.2020.9118472.

[19] G. Bradski, “The OpenCV library,” *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–125, 2000.

[20] Y. Liu, B. Zhang, N. Liu, H. Li, and J. Zhu, “Fisheye image distortion correction based on spherical perspective projection constraint,” in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, IEEE, Oct. 2020, pp. 1066–1070. doi: 10.1109/ICMA49215.2020.9233684.

[21] P. Bourke, “Converting dual fisheye images into a spherical (equirectangular) projection,” no. August. Paul Bourke, 2016. [Online]. Available: https://paulbourke.net/dome/dualfish2sphere/

[22] C.-H. Chao, P.-L. Hsu, H.-Y. Lee, and Y.-C. F. Wang, “Self-supervised deep learning for fisheye image rectification,” in *ICASSP*

2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, May 2020, pp. 2248–2252. doi: 10.1109/ICASSP40776.2020.9054191.

[23] S. Yang, C. Lin, K. Liao, C. Zhang, and Y. Zhao, "Progressively complementary network for fisheye image rectification using appearance flow," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2021, pp. 6344–6353. doi: 10.1109/CVPR46437.2021.00628.

[24] S.-H. Chiang, T. Wang, and Y.-F. Chen, "Efficient pedestrian detection in top-view fisheye images using compositions of perspective view patches," *Image Vis. Comput.*, vol. 105, p. 104069, Jan. 2021, doi: 10.1016/j.imavis.2020.104069.

[25] Z. Peng, X. Zhu, J. Wu, and Z. Qin, "A real-time fisheye video correction method based on Android smart-phone GPU," *Optik (Stuttg).*, vol. 220, p. 165108, Oct. 2020, doi: 10.1016/j.ijleo.2020.165108.

[26] M. Eder, M. Shvets, J. Lim, and J.-M. Frahm, "Tangent images for mitigating spherical distortion," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2020, pp. 12423–12431. doi: 10.1109/CVPR42600.2020.01244.

[27] Kukil, "Mean average precision (MAP) in object detection," LearnOpenCV. Accessed: Sep. 02, 2022. [Online]. Available: https://learnopencv.com/mean-average-precision-map-object-detection-model-evaluation-metric/

[28] Varun, "Camera calibration in Python with OpenCV," Python Wife. Accessed: Sep. 02, 2022. [Online]. Available: https://pythonwife.com/camera-calibration-in-python-with-opencv/

[29] Amro, "Similarity measurement (PSNR and SSIM)," GitHub. Accessed: Sep. 02, 2022. [Online]. Available: http://amroamroamro.github.io/mexopencv/opencv/image_similarity_demo.html

[30] M. Eder, "The backend code for my projects associated with spherical images," GitHub, Inc. Accessed: Sep. 02, 2022. [Online]. Available: https://github.com/meder411/Spherical-Package

[31] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, doi: 10.48550/arXiv.1912.01703.

[32] Astar-Ai and A. Zhang, "CaliCam_Mono: various Fisheye rectification modes," GitHub, Inc. Accessed: Sep. 02, 2022. [Online]. Available: https://github.com/astar-ai/calicam_mono

[33] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, Apr. 2007, pp. 3945–3950. doi: 10.1109/ROBOT.2007.364084.

[34] J. Angeley, "Converts FishEye pictures to Panoramic pictures," GitHub, Inc. Accessed: Sep. 02, 2022. [Online]. Available: https://github.com/Willfire19/FishToPan/blob/master/fishtopan.py

BIOGRAPHIES OF AUTHORS

John Benedict Du a recent graduate with a master's degree in Electronics Engineering from Ateneo de Manila University brings a keen understanding of advanced electronic systems. His academic journey has equipped him with expertise in cutting-edge technologies, and he is passionate about exploring innovations in the field. For any inquiries, you can reach out to John at john.du@obf.ateneo.edu.

Gian Paolo Mayuga received his B.S. degree in electronics and communications engineering and M.S. degree in electrical engineering from the University of the Philippines Diliman in Quezon City, Philippines. He received his Ph.D degree in information science from the Nara Institute of Science and Technology in Nara, Japan in 2016. He has published his research works in various international and local conferences and journals. His current research interests include intelligent computing and computer vision. He can be contacted at email: gmayuga@ateneo.edu.

Maria Leonora Guico is a licensed Electronics Engineer with over ten (10) years of combined experience working in the electronics and telecommunications industries, and more than twenty (20) years of teaching experience. She is currently an Assistant Professor in the Department of Electronics, Computer, and Communications Engineering (ECCE) at the School of Science and Engineering of the Ateneo de Manila University. She is also the head of the Mobile and Wireless Technologies Research Laboratory in the ECCE Department. Her current research areas of interest are wireless sensor networks and sensor technologies. She can be contacted at email: mguico@ateneo.edu.