YOLOv8-Based Distance Estimation for Blind Navigation: Performance Comparison of OpenCV and Coordinate Attention Techniques

Erwin Syahrudin[1], Ema Utami[2*], Anggit Dwi Hartanto[3]
[1−3]Master of Informatics, Faculty of Computer Science, Universitas Amikom Yogyakarta
Yogyakarta, Indonesia 55281
Email: [1]erwinsyahrudin@students.amikom.ac.id, [2]ema.u@amikom.ac.id, [3]anggit@amikom.ac.id

***Abstract*—Blindness presents a significant challenge in the development of assistive technologies, particularly for navigation, as it requires accurate distance perception to enable effective mobility for the visually impaired. The research addresses this issue by evaluating and comparing the performance of the YOLOv8 model integrated with OpenCV and the Coordinate Attention Weighting (CAW) technique for distance estimation in blind navigation systems. The main research objective is to improve distance estimation accuracy without the need for additional sensors. Initially, YOLOv8 with OpenCV shows less optimal results, prompting efforts to enhance its performance to surpass the effectiveness of CAW, while maintaining a sensor-free solution. The research then integrates YOLOv8 with OpenCV for baseline comparison and applies CAW for advanced feature attention in the distance estimation process. The research also integrates mathematical formulations for camera calibration and depth estimation, utilizing techniques such as triangulation and reprojection to refine the accuracy of object distance prediction. The results show that improved YOLOv8 + OpenCV significantly outperforms original YOLOv8 + OpenCV, with reduced Mean Squared Errors (MSE) across various distance intervals (0-1 m, 1-2 m, 2-3 m, 3-4 m, and 4-5 m). YOLOv8 + CAW also shows improvement compared to the original YOLOv8 + OpenCV but does not surpass the performance of the improved OpenCV integration. These findings demonstrate the potential of refined computer vision techniques in achieving high-accuracy and sensor-free distance estimation, enhancing real-time navigation systems for the blind. The research paves the way for further advancements in the development of accessible and reliable navigation technologies for the visually impaired.**

***Index Terms*—Computer Vision, YOLOv8, OpenCV, Coordinate Attention Weighting, Blind People**

Received: July 01, 2024; received in revised form: Nov. 18, 2024; accepted: Nov. 18, 2024; available online: April 17, 2025.
*Corresponding Author

I. Introduction

COMPUTER vision has advanced remarkably in recent years, fueled by rapid progress in deep learning techniques [1, 2]. These advancements have enabled systems to handle complex tasks with high accuracy and efficiency, proving essential in applications requiring precise spatial awareness. One of the practical applications of computer vision is assisting individuals with visual impairments in detecting objects and estimating distances. Such an application significantly enhances their independence and mobility [3]. By recognizing obstacles and gauging distances, they can navigate their surroundings more safely, fostering greater autonomy and confidence in daily life [4–6]. To achieve this goal, many researchers strive to develop assistive technologies, particularly through the optimization of existing computer vision frameworks [7–9].

One prominent real-time object detection approach is the You Only Look Once (YOLO) framework, known for its precision and efficiency. YOLOv8, the latest version, has demonstrated strong performance across various applications [10, 11]. When integrated with OpenCV, YOLOv8 creates a versatile system combining image processing, feature extraction, and geometric transformation tools to enhance object detection and distance estimation [12–14].

Generally, conventional OpenCV-based distance estimation methods rely on basic geometric calculations, which often lack precision in complex environments [15–17]. These methods typically use simple geometric principles such as triangulation or depth maps generated through stereo vision, which can be affected by noise or object occlusion in real-world scenarios. Recently, many researchers have made advancements

TABLE I
SAMPLE SUBSETS FROM DATASETS.

No	Dataset	Sample Size and Selected Attributes
1.	Common Objects in Context (COCO)	50 images with bounding boxes, object categories (person, car, dog)
2.	Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI)	10 images with bounding boxes, object categories (car, pedestrian), and Light Detection and Ranging (LIDAR) data
3.	Cityscapes	10 images with bounding boxes, object categories (pedestrian, car), and segmentation
4.	New York University Depth Dataset Version 2 (NYU Depth V2)	10 images with bounding boxes and depth maps for indoor scenes
5.	Custom Dataset	20 images with bounding boxes, object categories (chair, table), and ground truth distance

like Coordinate Attention Weighting (CAW), which have introduced more sophisticated approaches by incorporating spatial attention mechanisms, improving the accuracy of distance estimation in dynamic environments [18]. This method enhances the feature extraction process, allowing the system to focus on relevant regions and ignore irrelevant ones. As a result, the system's robustness in complex settings can be improved [11, 19, 20].

The research aims to address this gap by conducting a comprehensive comparative analysis of those two approaches: YOLOv8 integrated with OpenCV and YOLOv8 enhanced with Coordinate Attention Weighting (CAW). The primary objectives are to evaluate each approach's performance, accuracy, and reliability in object detection and distance estimation tasks. To achieve this, the researchers employ techniques such as camera calibration and depth estimation, utilizing mathematical formulations based on triangulation and reprojection [21, 22]. These methods refine the accuracy of object distance prediction, which is critical for blind navigation systems. The results from these models are compared using Mean Squared Errors (MSE) and other metrics, providing valuable insights toward the development of more sophisticated assistive technologies for visually impaired navigation.

Although the research does not propose new algorithms or mathematical formulations, it is still significant in the practical application of advanced models like YOLOv8 with OpenCV in assistive technologies. It optimizes YOLOv8 + OpenCV and evaluates the potential of integrating CAW to improve the spatial awareness of visually impaired individuals. Additionally, the researchers seek to advance the state-of-the-art in computer vision-based assistive systems, offering innovative solutions that enhance spatial awareness and safety for individuals with visual impairments.

II. Research Method

The research follows a structured approach to assess and compare the performance of two approaches for distance perception in navigation systems for visually impaired individuals: YOLOv8 integrated with OpenCV and YOLOv8 enhanced with CAW. This section details the essential components of the methodology, including the dataset, experimental setup, implementation procedures, evaluation criteria, and statistical analysis.

A. Dataset Subset for Testing

The research begins by examining datasets to evaluate the model's performance. A small and manageable portion of each dataset is selected for initial testing. These carefully chosen subsets encompass a wide variety of conditions, including different object types, distances, and lighting scenarios, ensuring that the model is tested in diverse real-world conditions [23]. This initial selection serves as a preliminary evaluation, helping to identify potential issues in the model's performance before scaling up to larger and more complex datasets. The subsets listed in Table I are used for both training and evaluation purposes, providing a solid foundation for further fine-tuning and validation of the model. A thorough initial assessment can be conducted by starting with these smaller subsets, and necessary adjustments can be made before proceeding to the full-scale testing phase.

The dataset subsets in Table I can be described as follows. First, a random sample of 50 images from the Common Objects in Context (COCO) dataset is selected, focusing on a few common object categories such as people, cars, and animals. The goal is to ensure the model can detect and estimate distances to various everyday objects in typical outdoor environments. Second, from the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset, 10 images are selected, including both outdoor scenes with vehicles and pedestrians. These images contain LIDAR data that can be used for distance validation and comparison. Third, 10 images from the Cityscapes dataset featuring urban environments are used. These images include typical urban objects like pedestrians, cars, and traffic signs, with corresponding segmentation and object category labels. Fourth, in the NYU Depth V2 dataset, 10 indoor scene images are

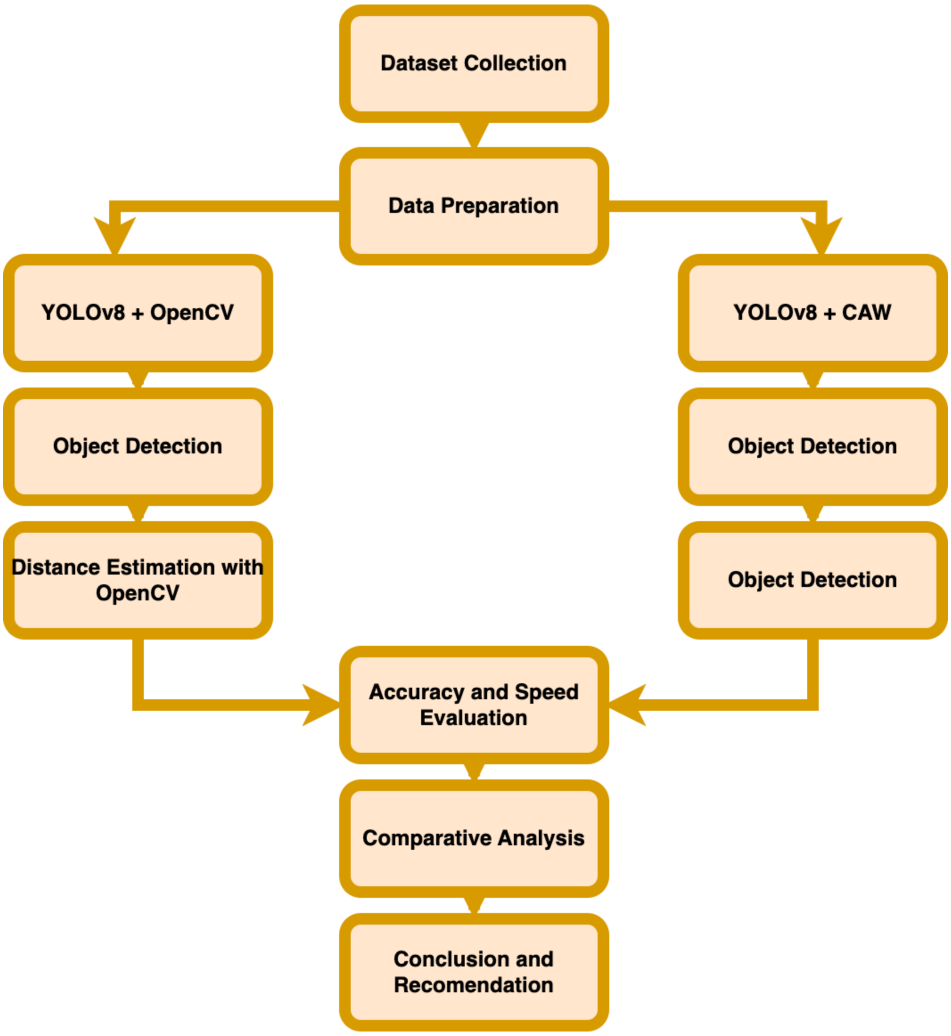

Fig. 1. Process of experimental setup. CAW is Coordinate Attention Weighting.

selected, including depth information. These images cover furniture, objects, and people in indoor environments, providing depth maps useful for testing depth estimation capabilities. Last, a set of 20 images from the custom dataset features common household and environmental objects (e.g., chairs, tables, and doors). These images include both object categories and annotated ground truth distances to reflect real-world navigation challenges for visually impaired individuals.

B. *Experimental Setup*

Once the subset is collected, an experimental setup is carried out using a structured and thorough approach. The detailed steps in setting up the experiment are illustrated in Fig. 1. A wide range of images and videos are gathered to represent different indoor and outdoor settings. These datasets are meticulously annotated with ground truth bounding boxes and distance details to ensure accurate evaluation. The datasets include publicly available sources, such as COCO and KITTI, along with a custom dataset specifically designed to simulate real-world navigation challenges faced by visually impaired individuals [24]. Additionally, the data collection process encompasses various images and videos with diverse lighting conditions, occlusions, and object types, ensuring the model's robustness across multiple environments. The data are then divided into three subsets: training, validation, and testing sets, to allow for a proper assessment of the model's ability to generalize.

For the analysis, YOLOv8 is implemented with both OpenCV and CAW for the distance estimation task. Object detection is applied to the dataset, and distances are calculated using two different approaches:

Cite this article as: E. Syahrudin, E. Utami, and A. D. Hartanto, “YOLOv8-Based Distance Estimation for Blind Navigation: Performance Comparison of OpenCV and Coordinate Attention Techniques”, CommIT Journal 19(1), 45–57, 2025.

Fig. 2. YOLOv8 Integration with OpenCV.

Fig. 3. YOLOv8 Integration with Coordinate Attention Weighting (CAW).

traditional geometric methods within OpenCV and an advanced spatial attention mechanism through CAW. Then, model performance is evaluated using several metrics, including accuracy (Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE)) and speed (Frames per Second (FPS) and inference time) [25]. A comparative analysis is also conducted to examine the statistical differences between the two methods, helping the researchers to determine the most effective approach for distance estimation in blind navigation systems. The research is concluded by offering recommendations for further improvements in assistive technologies, especially for visually impaired individuals.

C. Implementation Details

The experimental setup in the research is implemented in two primary configurations: YOLOv8 integrated with OpenCV and YOLOv8 enhanced with CAW. Each configuration follows a series of steps for object detection and distance estimation. Figures 2 and 3 outline the detailed processes involved in the implementation using OpenCV and CAW, respectively.

As illustrated in Fig. 2, the YOLOv8 model is employed to identify objects within images. It produces bounding boxes and class labels for each detected object. Initially, the model is loaded with pre-trained weights, which are then fine-tuned using a custom dataset to improve detection performance for specific situations encountered by visually impaired users. Geometric calculations are employed and facilitated by OpenCV functions to estimate the distance of the detected objects. This process uses the known physical size of the objects and their image size to determine the distance [26, 27]. Methods such as monocular depth estimation or stereo vision are applied for enhanced accuracy.

As shown in Fig. 3, the CAW is incorporated into the YOLOv8 framework. This process is done by adding a CAW layer, which enhances spatial attention and boosts the model’s focus on important features [27]. The modified YOLOv8+CAW model is then trained on a custom dataset to optimize its detection performance. The output from the YOLOv8+CAW model is utilized for more accurate distance estimation. Previous research argues that the CAW mechanism enhances spatial awareness, resulting in more precise distance measurements [28]. While similar geometric and vision-based methods have been used in the OpenCV approach, the inclusion of CAW provides improved attention to object features for better outcomes.

III. Results and Discussion

The research results are presented in two main parts: evaluation metrics and statistical analysis. In each section, a detailed comparison between the performance of YOLOv8 integrated with OpenCV and YOLOv8 enhanced with CAW is made. The goal is to assess how these approaches impact distance estimation in navigation systems intended for individuals with visual impairments.

A. Evaluation Metrics

The evaluation of between YOLOv8 combined with OpenCV and YOLOv8 enhanced with CAW is conducted using several important performance metrics, including MAE, RMSE, FPS, and inference time. The findings are presented in Tables II to IV, which outline the strengths and weaknesses of each model across different scenarios. The first metric, MAE, is obtained with Eq. (1). It has yi as the actual distance, $\hat{y}i$ as the estimated distance, and n as the number of observations [28]. Meanwhile, the second metric, RMSE is expressed using Eq. (2). RMSE is calculated as the square root of the mean of the squares of the differences between the observed values yi and the predicted values $\hat{y}i$ [29]. Both metrics are used to measure the accuracy of distance estimation, where lower values in these metrics indicate higher accuracy. MAE represents the average error, while RMSE

Cite this article as: E. Syahrudin, E. Utami, and A. D. Hartanto, “YOLOv8-Based Distance Estimation for Blind Navigation: Performance Comparison of OpenCV and Coordinate Attention Techniques”, CommIT Journal 19(1), 45–57, 2025.

TABLE II
ACCURACY METRICS OF MEAN ABSOLUTE ERROR (MAE) AND ROOT MEAN SQUARED ERROR (RMSE).

Metric	YOLOv8 + OpenCV	YOLOv8 + Coordinate Attention Weighting (CAW)
MAE (m)	0.42	0.35
RMSE (m)	0.58	0.50

TABLE III
SPEED OF OBJECT DETECTION AND DISTANCE ESTIMATION.

Metric	YOLOv8 + OpenCV	YOLOv8 + Coordinate Attention Weighting (CAW)
Frames per Second (FPS)	25	20
Inference Time (ms)	41	51

emphasizes larger discrepancies by squaring the error differences [30].

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |yi - \hat{y}i|, \quad (1)$$

$$\text{RSME} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2}. \quad (2)$$

Table II shows that YOLOv8 + CAW achieves lower MAE (0.35 m) and RMSE (0.50 m) than YOLOv8 + OpenCV with MAE at 0.35 m and RMSE at 0.50 m. The result suggests that the CAW mechanism improves the model’s focus on essential features, enabling more precise distance estimations by enhancing spatial attention and reducing the impact of irrelevant background elements. The improved feature weighting allows the model to better differentiate objects from their surroundings, leading to more consistent and reliable distance measurements across various lighting conditions and object types.

After that, the FPS and metrics inference time are assessed using Eqs. (3) and (4). They evaluate the model’s efficiency in real-time applications. The FPS measurement determines how smoothly the system can process and detect objects per second, while the inference time analysis helps assess the computational load and responsiveness of the model. These evaluations are crucial for ensuring that the proposed approach is not only accurate but also practical for real-world deployment.

$$\text{FPS} = \frac{\text{Number of Frames}}{\text{Total Time (seconds)}}, \quad (3)$$

$$\text{Inference Time (ms)} = \frac{\text{Total Time (ms)}}{\text{Number of Frame}}. \quad (4)$$

TABLE IV
ROBUSTNESS MODEL WITH CONDITION LOW LIGHT AND HIGH OBJECT DENSITY.

Condition	YOLOv8 + OpenCV	YOLOv8 + CAW
Low Light	0.61 MAE	0.41 MAE
Occluded Objects	0.56 MAE	0.39 MAE
High Object Density	0.66 MAE	0.46 MAE

Note: Coordinate Attention Weighting (CAW) and Mean Absolute Error (MAE).

Real-time performance is essential in blind navigation systems, where high-speed processing ensures timely object detection and distance estimation. As seen in Table III, YOLOv8 + OpenCV achieves a higher FPS (25) and lower inference time (41 ms) than YOLOv8 + CAW, with an FPS of 20 and an inference time of 51 ms. Although OpenCV is faster, the difference is relatively minimal, and both configurations can operate in real time. The slight speed advantage of YOLOv8 + OpenCV is counterbalanced by the increased accuracy of YOLOv8 + CAW, making both models viable for practical applications depending on specific needs.

Table IV examines model robustness under low-light conditions, occlusions, and high object density. In these challenging scenarios, YOLOv8 + CAW outperforms YOLOv8 + OpenCV, showing MAE values of 0.41, 0.39, and 0.46. In contrast, YOLOv8 + OpenCV exhibits higher MAE values with 0.61, 0.56, and 0.66. This robustness in CAW is critical for applications involving visually impaired users, where environmental variations can heavily impact usability.

B. Enhanced YOLOv8 with OpenCV Integration for Improved Distance Detection

From the previous results, several strategies can be applied to enhance the accuracy performance of YOLOv8 for distance detection using OpenCV and bring it closer to the capabilities of CAW. These strategies may include optimizing the preprocessing pipeline, fine-tuning the model’s hyperparameters, and incorporating advanced filtering techniques such as Kalman filters to refine distance estimation. The enhancement presents a viable alternative to CAW without the need for extra sensors, reducing hardware dependency while maintaining computational efficiency. This approach, as illustrated in Fig. 4, ensures that the system remains accessible, cost-effective, and practical for real-world deployment, particularly in assistive technologies for visually impaired users.

In particular, Table V outlines the specific steps involved in the process. These steps include im-

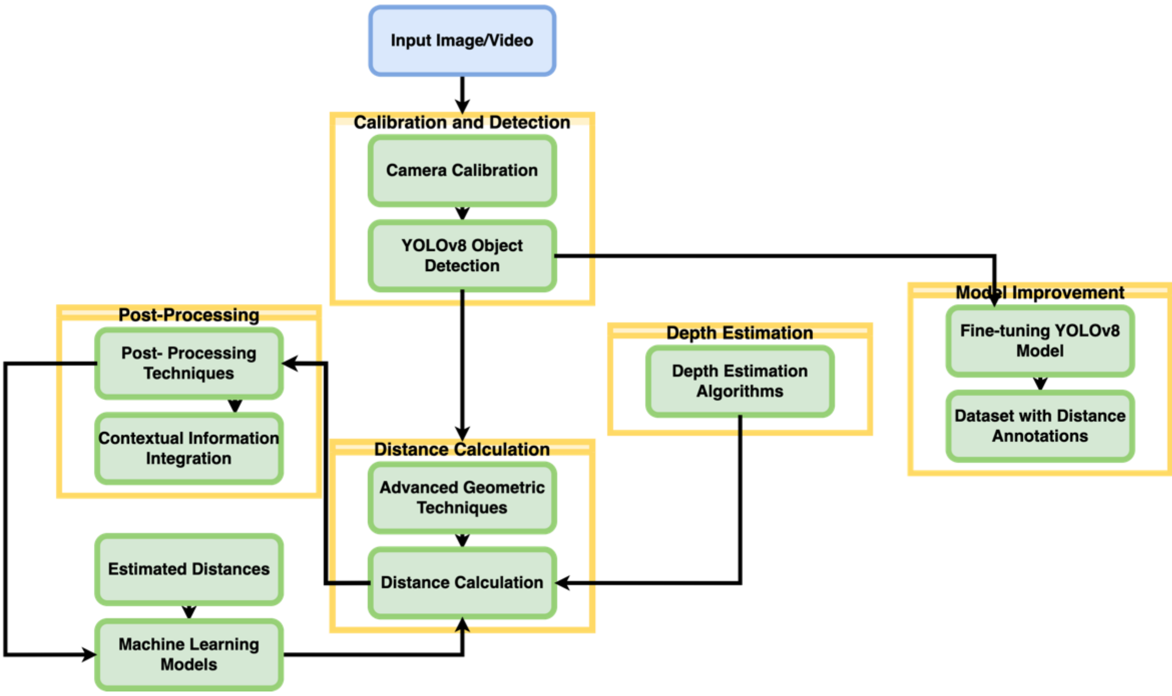

Fig. 4. YOLOv8 with OpenCV for distance detection.

TABLE V
PROCESS FLOW FOR YOLOV8-BASED OBJECT DETECTION AND DISTANCE ESTIMATION SYSTEM.

Step	Purpose	Techniques Used	Expected Outcome
1. Input image/video	Provide initial visual data for analysis	Image or video feed	Visual input for object detection and distance estimation
2. Calibration and detection	Ensure accurate object detection and measurement	Camera calibration, YOLOv8 object detection	Calibrated camera para m and detected objects with bounding boxes
2.1 Camera calibration	Adjust camera parameters to reduce distortion	Calibration algorithms	Optimized camera setup for precise distance measurements
2.2 YOLOv8 object detection	Identify and classify objects in the scene	YOLOv8 model for detection	Object classes and bounding boxes
3. Distance calculation	Calculate the distance between the camera and the detected objects	Advanced geometric techniques, distance calculation	Estimated object distances
4. Depth estimation	Enhance spatial understanding of detected objects	Depth estimation algorithms (stereo or monocular)	Refined distance information, adding depth context
5. Model improvement	Improve YOLOv8's detection and distance estimation performance	Fine-tuning YOLOv8, a dataset with distance annotations	Enhanced detection and distance accuracy
6. Post-processing	Refine and filter distance estimation results	Post-processing techniques, contextual integration	Smoothed and contextually relevant distance data
6.1 Post-processing techniques	Filter noise and refine distance estimations	Smoothing and filtering irrelevant data	Reliable and accurate distance results
6.2 Contextual information integration	Incorporate environmental context for relevance	Environmental/contextual data analysis	More meaningful distance estimates with context
7. Estimated distances	Provide final distances for detected objects	Consolidation of results	Final object distance estimates ready for use
7.1 Machine learning models	Enable further analysis or predictive insights	Machine learning algorithms	Enhanced accuracy or customization for specific applications

proving camera calibration, setting up a stereo camera system for depth perception, utilizing advanced depth estimation techniques to create depth maps, and combining these with point cloud data for precise distance measurement. Additional techniques include post-processing methods such as filtering, integrating contextual information, and fine-tuning the YOLOv8 model using annotated distance data to refine accuracy further.

The approach takes advantage of a well-established, open-source computer vision library recognized for its powerful tools in real-time image processing and depth estimation using OpenCV as a replacement for CAW. This method ensures both high accuracy and

TABLE VI
INPUT IMAGE OR VIDEO.

Input Type	Description	Example	Effect on System
Image	Single-frame image for object detection.	A static image from a camera.	Providing a snapshot for object detection.
Video	A continuous video stream for real-time processing.	A video feed from a camera.	Enabling real-time object tracking and distance estimation.

TABLE VII
CAMERA CALIBRATION.

Calibration Step	Methodology	Description	Effect on System
Intrinsic Calibration	Camera matrix and distortion coefficients.	Correcting lens distortion and ensuring accurate image capture.	Improving accuracy in object detection and distance estimation.
Extrinsic Calibration	Camera positioning and orientation.	Ensuring correct placement of the camera in space.	Being essential for spatial understanding and depth calculation.

TABLE VIII
RESULTS OF DISTANCE CALCULATION.

Distance Range	YOLOv8 + OpenCV MAE	YOLOv8 + OpenCV Improved MAE	YOLOv8 + CAW MAE	YOLOv8 + OpenCV RMSE	YOLOv8 + OpenCV Improved RMSE	YOLOv8 + CAW RMSE
0-1 m	0.42 m	0.38 m	0.39 m	0.52	0.48	0.48
1-2 m	0.48 m	0.44 m	0.45 m	0.57	0.54	0.53
2-3 m	0.56 m	0.51 m	0.52 m	0.64	0.60	0.60
3-4 m	0.61 m	0.58 m	0.58 m	0.69	0.65	0.66
4-5 m	0.73 m	0.68 m	0.67 m	0.78	0.74	0.74

Note: Coordinate Attention Weighting (CAW), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

efficiency while being cost-effective and accessible for developing assistive technologies for individuals with visual impairments.

C. Process Implementation

The following section delves into a comprehensive comparison of the various configurations of YOLOv8, specifically focusing on the original YOLOv8 integrated with OpenCV, an enhanced version of YOLOv8 with OpenCV, and enhanced YOLOv8 with CAW. Table VI provide detailed quantitative insights into how each configuration performs across varying distance ranges. It showcases how even small adjustments can significantly impact the model’s ability to provide reliable distance perception, especially for visually impaired navigation.

In Table VI, two main types of input are discussed: image and video. Image refers to the use of a single image acquired from a camera for object detection, providing a static image useful for object analysis. Video, on the other hand, refers to the use of a continuous video stream, allowing real-time processing for continuous object tracking and distance estimation. Static images are more suitable for one-off analysis applications, while video is more suitable for applications that require real-time interaction, such as navigation for the visually impaired [31].

Table VII discusses the steps in camera calibration. In the research, intrinsic calibration uses the camera matrix and distortion coefficients to correct lens distortion, which improves the accuracy of object detection and distance estimation [32]. Meanwhile, extrinsic calibration, which involves determining the position and orientation of the camera in space, is essential for understanding object mapping accurately in real environments [33, 34]. Without proper calibration, the system cannot produce accurate distance estimation, especially in applications for blind people that require high precision in object detection [35].

Table VIII reveals the performance of three machine learning models for distance estimation. The models are evaluated at five different distance ranges (0-1 to 4-5 m) using two main metrics: MAE and RMSE. The models tested are YOLOv8 + OpenCV, improved YOLOv8 + OpenCV, and YOLOv8 + CAW. It provides an in-depth look at the effectiveness of each model in estimating distance accurately.

MAE measures the average absolute error between the actual and predicted distances, where a lower value indicates higher estimation accuracy [25]. In the 0-1 m range, the improved YOLOv8 + OpenCV model performs the best with an MAE of 0.38 m, which is lower than the standard YOLOv8 + OpenCV model with an MAE of 0.42 m, and YOLOv8 + CAW with

TABLE IX
RESULTS OF DEPTH ESTIMATION.

Depth Estimation Method	YOLOv8 + OpenCV	YOLOv8 + OpenCV Improved	YOLOv8 + CAW
Stereo Vision	0.60 MAE	0.55 MAE	0.50 MAE
Monocular Depth Estimation	0.45 MAE	0.40 MAE	0.42 MAE
Depth Map Integration	0.52 MAE	0.48 MAE	0.49 MAE

Note: Coordinate Attention Weighting (CAW) and Mean Absolute Error (MAE).

TABLE X
MODEL IMPROVEMENTS.

Improvement Type	Implementation	Improved YOLOv8 + OpenCV
Stereo Vision	Enhancing the camera's ability to capture accurate images.	Reducing distortion errors.
Feature Attention	Using Coordinate Attention Weighting (CAW).	Improving object recognition and distance perception.
Model Optimization	Fine-tuning the YOLOv8 model for specific tasks.	Increasing accuracy of distance estimation.
YOLOv8 Improvement	Improving model architecture and fine-tuning for specific environments.	Further enhancing detection accuracy, especially in complex scenarios.

0.39 m. This finding shows that the improved model has an advantage in detecting objects at close range with high accuracy.

In the 1-2 m range, improved YOLOv8 + OpenCV also excels with an MAE of 0.44 m, followed by YOLOv8 + CAW at 0.45 m. On the other hand, YOLOv8 + OpenCV still has a larger error at 0.48 m. This trend continues in the following distance ranges. At a distance of 2-3 m, the improved model has an MAE value of 0.51 m, smaller than YOLOv8 + OpenCV at 0.56 m and YOLOv8 + CAW at 0.52 m. Likewise, at the 3-4 m and 4-5 m ranges, improved YOLOv8 + OpenCV maintains its advantage with MAEs of 0.58 m and 0.68 m, respectively, indicating a consistent improvement in accuracy in distance estimation.

RMSE gives more weight to larger errors, which can reveal the system's performance in managing more extreme error variations [36]. At 0-1 m, the improved YOLOv8 + OpenCV and YOLOv8 + CAW models have the same RMSE value at 0.48, lower than YOLOv8 + OpenCV, which is at 0.52. This finding shows that improving YOLOv8 with additional methods, such as improved OpenCV and CAW, can reduce significant errors at close range. At 1-2 m, improved YOLOv8 + OpenCV achieves an RMSE of 0.54, better than YOLOv8 + OpenCV with an RMSE of 0.57. It is also slightly better than YOLOv8 + CAW which reaches 0.53.

As the distance increases, the improved YOLOv8 + OpenCV model continues to show superiority in reducing errors, with RMSE values of 0.60 at 2-3 meters, 0.65 at 3-4 m, and 0.74 at 4-5 m, respectively. On the other hand, YOLOv8 + CAW shows similar RMSE values as improved YOLOv8 + OpenCV at 2-3 m and 4-5 m, 0.60 and 0.74, respectively. These values indicate that CAW is also quite effective in maintaining the stability of distance estimation at larger distances.

Then, depth estimation is evaluated using three different methods: stereo vision, monocular depth estimation, and depth map integration, as shown in Table IX. Improved YOLOv8 + OpenCV shows an improvement in depth estimation accuracy compared to the standard model, both in stereo vision and monocular depth estimation [37, 38]. It suggests that enhancing the YOLOv8 model with OpenCV can improve the accuracy of depth measurement, which is important for applications involving spatial navigation, such as for the visually impaired.

The research also describes various types of model improvements, such as camera calibration, use of feature attention (CAW) techniques, optimization of the YOLOv8 model, and improvements to the YOLOv8 architecture itself. Table X shows that each improvement aims to improve the accuracy of object detection and distance estimation. For example, the use of CAW in the model can improve object recognition and depth understanding, which are very important in applications for the visually impaired.

Table XI explains the post-processing techniques applied to improve the detection results. In the research, all models apply filtering and depth smoothing techniques to reduce noise and produce more stable depth estimation. However, only improved YOLOv8 + OpenCV and YOLOv8 + CAW models applied contextual refinement, which integrates contextual information to improve the results of object detection and distance estimation. It provides significant improvements in environment recognition for the visually impaired.

In Table XII, two main types of contextual information are integrated into the system to improve the accuracy of detection and distance estimation, namely envi-

TABLE XI
POST-PROCESSING TECHNIQUES.

Technique	YOLOv8 + OpenCV	Improved YOLOv8 + OpenCV	YOLOv8 + CAW
Filtering	Yes	Yes	Yes
Depth Smoothing	Yes	Yes	Yes
Contextual Refinement	Yes	Yes	Yes

Note: Coordinate Attention Weighting (CAW).

TABLE XII
CONTEXTUAL INFORMATION INTEGRATION.

Information Type	Integration Method	Effect on Accuracy
Environmental Context	Location data integration.	Increasing the accuracy of object detection in different environments.
Object Tracking	Integration with tracking algorithms.	Improving distance estimation by reducing errors caused by object movement.

TABLE XIII
RESULT DETECTION DISTANCE IN THE REAL WORLD.

Object	YOLOv8 + OpenCV	YOLOv8 + CAW	Improved YOLOV8 + OpenCV
People	1.35 m	1.28 m	1.05 m
Motorcycle	3.47 m	3.05 m	2.71m
Motorcycle	4.53 m	3.83 m	3.53 m
Motorcycle	5.37 m	4.42 m	4.19 m
Bus	2.04 m	1.89 m	1.59 m
People	5.10 m	4.23 m	3.98 m
Truck	2.35 m	2.14 m	1.83 m
Motorcycle	14.11 m	9.02 m	11.01 m
People	2.01 m	1.86 m	1.57 m
Motorcycle	2.22 m	2.03 m	1.73 m
MSE	4.5623	3.2621	3.0834

Note: Coordinate Attention Weighting (CAW) and Mean Squared Error (MSE).

ronmental context and object tracking. Environmental context includes the integration of location data that allows the system to recognize different environments and adjust object detection to specific conditions at a particular location, such as indoors or outdoors [39]. It helps to improve the accuracy of detecting objects that may change depending on environmental conditions.

Next, object tracking is implemented with a tracking algorithm to follow the movement of objects continuously. It is very important in reducing distance estimation errors that may occur due to the movement of objects in the video frame. By following the position of the object, the system can estimate the distance more accurately and stably, which is very valuable in real-time navigation applications for blind people.

Figure 5 illustrates the comparison of distance estimation results using three different approaches: YOLOv8 + OpenCV, YOLOv8 + CAW, and improved YOLOv8 + OpenCV. As shown in Table II, YOLOv8 + CAW provides more accurate distance estimations compared to YOLOv8 + OpenCV, particularly for various objects such as people, motorcycles, buses, and trucks. However, the enhancements in improved YOLOv8 + OpenCV demonstrate that an optimized model can achieve closer accuracy to YOLOv8 + CAW and, in some cases, even surpass it. For instance, the initial distance estimation for a person at 1.35 m using YOLOv8 + OpenCV improves to 1.05 m after optimization. Similarly, for a motorcycle initially estimated at 5.37 m, the improved model refines it to 4.19 m. Overall, the enhancements in YOLOv8 + OpenCV show significant improvements without requiring additional attention mechanisms like CAW, suggesting that with better data processing, signal filtering, and refined distance modeling, improved YOLOv8 + OpenCV can serve as a more efficient alternative for accurate distance detection without the need for additional sensors.

The real-world test results in Table XIII provide a comparative analysis of distance estimation techniques using YOLOv8 with three approaches: OpenCV (baseline), CAW, and improved OpenCV. The evaluation focuses on the Mean Squared Error (MSE) value [40], which is calculated against the synthetic ground truth distance. The improved OpenCV method shows the lowest MSE of 3.0834, indicating its improved accuracy in distance prediction. It is visually represented in Fig. 5, where the improved calibration and optimization reduce the estimation error more compared to the baseline and CAW methods. Conversely, the baseline OpenCV approach has the highest MSE of

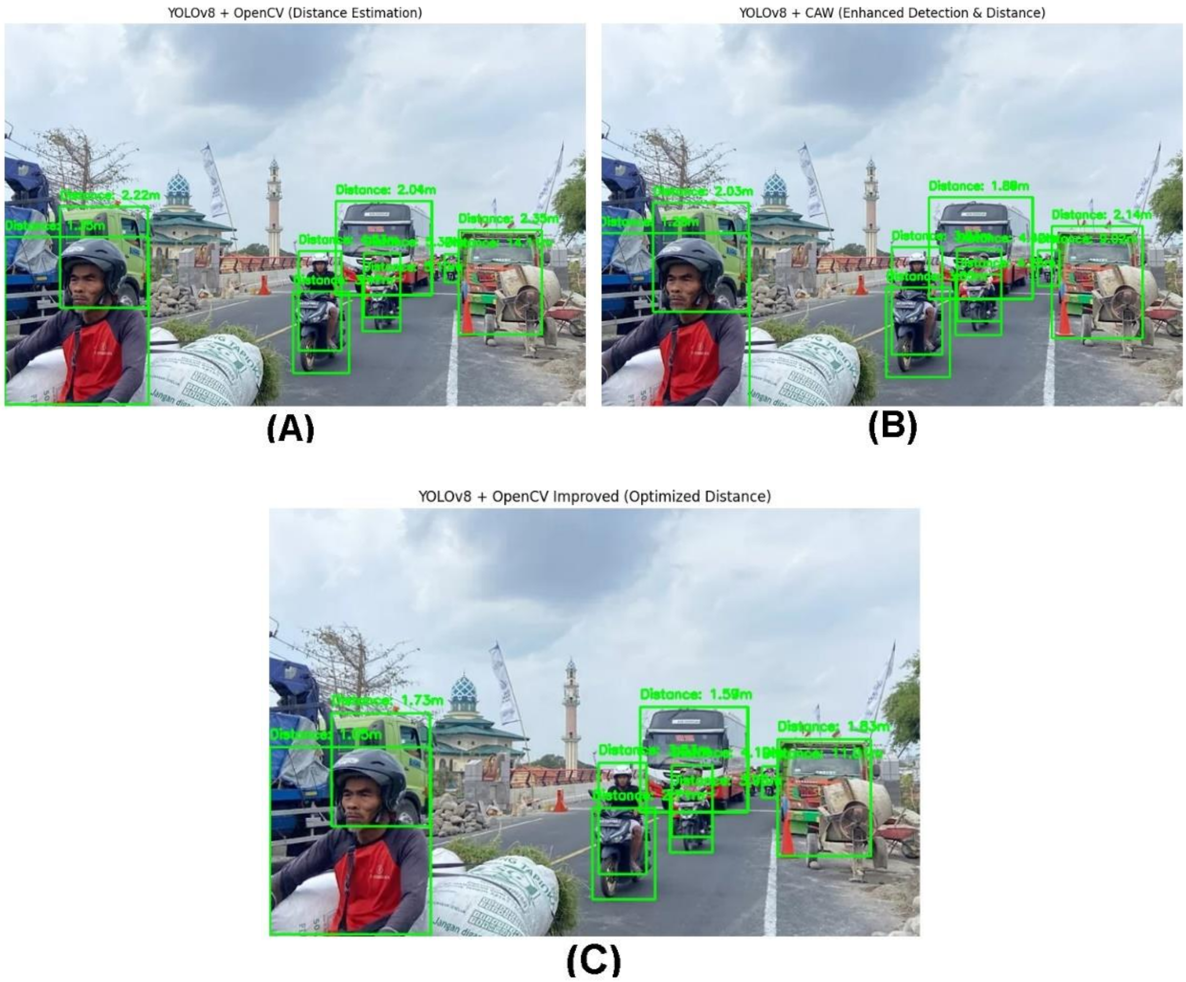

Fig. 5. Comparison of distance estimation approaches in the real world.

4.5623, indicating less precise distance prediction. The improved CAW approach achieves a medium MSE of 3.2621 by increasing feature attention and bounding box precision.

The MSE formula is adapted to incorporate a combination of geometric and optimization principles to explain this improvement mathematically. In Eq. (5), $\hat{d}_i$ represents the estimated distance, d_i is the ground truth distance, and N is the total number of objects. Then, the predicted distance $\hat{d}_i$ in the improved OpenCV method is derived using a calibrated version of the triangulation formula, incorporating optimized parameters for focal length (f^1) and real-world object width (W^1) [41].

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{d}_i - d_1)^2, \tag{5}$$

In Eq. (6), $\hat{d}_i$ is the pixel width of the detected object. Then, p is is obtained from the results of object detection in images using algorithms such as OpenCV. For example, if it detects a water bottle, and its bounding box in the image has a width of 80 pixels, p is 80. By adjusting f^1 and W^1 to account for real-world variations, the improved OpenCV method reduces estimation errors. The combination of these adjustments can be viewed as a systematic improvement modeled by weighted contributions from baseline and enhanced methods in Eq. (7). The α and β are weighting factors representing the contributions of the baseline and optimized methods. This formulation allows for a fine-tuned balance, leveraging the strengths of each approach. The results from Table XIII affirm that integrating geometric principles [42], calibration, and optimization techniques in the improved OpenCV method significantly enhances accuracy, making it well-suited for critical applications like blind navigation systems.

$$\hat{d_i} = \frac{f^1.W^1}{p}, \tag{6}$$

$$\hat{d_i} = a.d_{\text{baseline}} + \beta.d_{\text{enhanced}}, \quad \text{where } \alpha + \beta = 1. \tag{7}$$

IV. Conclusion

In the research, the researchers compare the performance of YOLOv8 integrated with OpenCV and CAW techniques for distance estimation in a blind navigation system. Initially, YOLOv8 with OpenCV exhibits lower accuracy compared to YOLOv8 with CAW. Thus, improvements are made in the OpenCV integration. By incorporating the spatial attention mechanism from CAW and refining OpenCV's processing, the researchers significantly enhance the accuracy of distance estimation. The main results show that improved YOLOv8 + OpenCV outperforms original YOLOv8 + OpenCV and approaches the performance of YOLOv8 + CAW in terms of accuracy, with a lower MSE, calculated using geometric principles and triangulation-based depth estimation.

These findings emphasize the potential of integrating OpenCV with deep learning models, such as YOLOv8, for practical, sensor-free blind navigation systems. Enhancing distance perception through methods like camera calibration and reprojection reduces the need for additional hardware sensors. The application of CAW further optimizes the feature extraction process, increasing the system's ability to detect and measure objects in dynamic and complex environments. This approach offers an accessible and cost-effective solution for navigation technologies, providing visually impaired individuals with a more reliable means of spatial awareness. However, limitations exist in the performance of each method depending on the environment. For example, the original YOLOv8 + OpenCV is more effective in straightforward settings with minimal obstacles, whereas YOLOv8 + CAW excels in complex environments with variable object sizes and occlusions. The MSE calculations, based on triangulation and reprojection methods, reveal how accuracy is affected in various conditions.

Further research should explore these environmental factors and test the models in real-world scenarios. In particular, future research can benefit from collecting feedback from visually impaired users, allowing for the refinement of these models to enhance their practical utility in everyday navigation. Additionally, combining deep learning models with real-time image processing, such as camera calibration and depth estimation techniques, can further strengthen the robustness and efficiency of blind navigation systems.

Acknowledgement

The authors would like to acknowledge the Ministry of Education, Research, and Technology for supporting the completion of the research through valuable funding under a postgraduate program research grant for the fiscal year 2024.

Author Contribution

Conceived and designed the analysis, E. U. ; Collected the data, E. U. ; Contributed data or analysis tools, E. S.; Perfomed the analysis, E. S.; Wrote the paper, A. D. H.; and Supervised the paper, E. U. and A. D. H.

Data Availability

The data that support the findings of this research are openly available in the following repositories: Taylor & Francis Online at https://doi.org/10.1080/02564602.2020.1819893, reference number 02564602, RESTI Journal at https://doi.org/10.29207/resti.v8i2.5529, reference number 5529, IEEE Xplore at https://doi.org/10.1109/ICCR56254.2022.9995808, reference number ICCR56254, Health Informatics Journal at https://doi.org/10.1177/14604582221112609, reference number 14604582, and IOP Conference Series: Materials Science and Engineering at https://doi.org/10.1088/1757-899X/1085/1/012006, reference number 1085/1/012006.

References

[1] A. Budrionis, D. Plikynas, P. Daniušis, and A. Indrulionis, "Smartphone-based computer vision travelling aids for blind and visually impaired individuals: A systematic review," *Assistive Technology*, vol. 34, no. 2, pp. 178–194, 2022.

[2] D. Coşkun, D. Karaboğa, A. Baştürk, B. Akay, Ö. U. Nalbantoğlu, S. Doğan, İ. Paçal, and M. A. Karagöz, "A comparative study of YOLO models and a transformer-based YOLOv5 model for mass detection in mammograms," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 31, no. 7, pp. 1294–1313, 2023.

[3] D. S. Bacea and F. Oniga, "Single stage architecture for improved accuracy real-time object detection on mobile devices," *Image and Vision Computing*, vol. 130, pp. 1–9, 2023.

[4] M. A. Rahman, S. Siddika, M. A. Al-Baky, and M. J. Mia, "An automated navigation system for blind people," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 201–212, 2022.

[5] ——, "An automated navigation system for blind people," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 201–212, 2022.

[6] S. Khan, S. Nazir, and H. U. Khan, "Analysis of navigation assistants for blind and visually impaired people: A systematic review," *IEEE Access*, vol. 9, pp. 26 712–26 734, 2021.

[7] F. E. Z. El-Taher, A. Taha, J. Courtney, and S. Mckeever, "A systematic review of urban navigation systems for visually impaired people," *Sensors*, vol. 21, no. 9, pp. 1–35, 2021.

[8] J. Hwang, K. H. Kim, J. G. Hwang, S. Jun, J. Yu, and C. Lee, "Technological opportunity analysis: Assistive technology for blind and visually impaired people," *Sustainability*, vol. 12, no. 20, pp. 1–17, 2020.

[9] R. B. Islam, S. Akhter, F. Iqbal, M. S. U. Rahman, and R. Khan, "Deep learning based object detection and surrounding environment description for visually impaired people," *Heliyon*, vol. 9, no. 6, pp. 1–19, 2023.

[10] S. Sun, B. Mo, J. Xu, D. Li, J. Zhao, and S. Han, "Multi-YOLOv8: An infrared moving small object detection model based on YOLOv8 for air vehicle," *Neurocomputing*, vol. 588, pp. 1–24, 2024.

[11] Z. J. Khow, Y. F. Tan, H. A. Karim, and H. A. A. Rashid, "Improved YOLOv8 model for a comprehensive approach to object detection and distance estimation," *IEEE Access*, vol. 12, pp. 63 754–63 767, 2024.

[12] M. Wang, Y. Jiang, C. Li, and M. Yang, "A new data processing model for distributed urban stagnant analysis based on improved YOLOv5 and OpenCV," 2023. [Online]. Available: https://www.preprints.org/manuscript/202305.1746/v1

[13] M. Shoeb, M. A. Ali, M. Shadeel, and M. A. Bari, "Self-driving car: Using OpenCV2 and machine learning," *The International Journal of Analytical and Experimental Modal Analysis (IJAEMA)*, vol. XIV, no. V, pp. 325–330, 2022.

[14] B. R. Patel, S. A. Goswami, P. S. KaPatel, and Y. M. Dhakad, "Realtime object's size measurement from distance using OpenCV and LiDAR," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 4, pp. 1044–1047, 2021.

[15] M. Zha, W. Qian, W. Yi, and J. Hua, "A lightweight YOLOv4-based forestry pest detection method using coordinate attention and feature fusion," *Entropy*, vol. 23, no. 12, pp. 1–18, 2021.

[16] Z. Li, M. Xue, Y. Cui, B. Liu, R. Fu, H. Chen, and F. Ju, "Lightweight 2D human pose estimation based on joint channel coordinate attention mechanism," *Electronics*, vol. 13, no. 1, pp. 1–16, 2023.

[17] R. Wang, F. Liang, B. Wang, and X. Mou, "ODCA-YOLO: An omni-dynamic convolution coordinate attention-based YOLO for wood defect detection," *Forests*, vol. 14, no. 9, pp. 1–18, 2023.

[18] F. Xie, B. Lin, and Y. Liu, "Research on the coordinate attention mechanism fuse in a YOLOv5 deep learning detector for the SAR ship detection task," *Sensors*, vol. 22, no. 9, pp. 1–16, 2022.

[19] J. Wu, J. Dong, W. Nie, and Z. Ye, "A lightweight YOLOv5 optimization of coordinate attention," *Applied Sciences*, vol. 13, no. 3, pp. 1–13, 2023.

[20] C. Xie, H. Zhu, and Y. Fei, "Deep coordinate attention network for single image super-resolution," *IET Image Processing*, vol. 16, no. 1, pp. 273–284, 2022.

[21] Z. Hu, X. Zhao, J. Zhang, S. Ba, Z. Zhao, and X. Wang, "Parameter calibration and verification of elastoplastic wet sand based on attention-retention fusion deep learning mechanism," *Applied Sciences*, vol. 14, no. 16, pp. 1–23, 2024.

[22] X. Yu, J. Liu, Y. Lu, S. Funahashi, T. Murai, J. Wu, Q. Li, and Z. Zhang, "Early diagnosis of Alzheimer's disease using a group self-calibrated coordinate attention network based on multimodal MRI," *Scientific Reports*, vol. 14, no. 1, pp. 1–18, 2024.

[23] J. Kaur and W. Singh, "Tools, techniques, datasets and application areas for object detection in an image: A review," *Multimedia Tools and Applications*, vol. 81, no. 27, pp. 38 297–38 351, 2022.

[24] B. Strbac, M. Gostovic, Z. Lukac, and D. Samardzija, "YOLO multi-camera object detection and distance estimation," in *2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*. Novi Sad, Serbia: IEEE, May 26–27, 2020, pp. 26–30.

[25] M. Vajgl, P. Hurtik, and T. Nejezchleba, "Dist-YOLO: Fast object detection with distance estimation," *Applied Sciences*, vol. 12, no. 3, pp. 1–13, 2022.

[26] A. B. Abadi and S. Tahcfulloh, "Digital image processing for height measurement application based on Python OpenCV and regression analysis," *JOIV: International Journal on Informatics Visualization*, vol. 6, no. 4, pp. 763–770, 2022.

[27] H. Varçin, F. Üneş, E. Gemici, and M. Zelenakova, "Development of a three-dimensional CFD model and OpenCV code by comparing with experimental data for spillway model stud-

ies," *Water*, vol. 15, no. 4, pp. 1–31, 2023.

[28] S. M. Robeson and C. J. Willmott, "Decomposition of the Mean Absolute Error (MAE) into systematic and unsystematic components," *PLOS ONE*, vol. 18, no. 2, pp. 1–8, 2023.

[29] D. Ganga, V. Bharath, P. N. Sri, T. Tulasi, and S. K. Sharook, "Social distance detector using OpenCV YOLO, CNN algorithm in deep learning," *ZKG International*, vol. 8, pp. 893–897.

[30] Z. Sun, M. Lin, X. Sun, Z. Tan, H. Li, and R. Jin, "MAE-DET: Revisiting maximum entropy principle in zero-shot NAS for efficient object detection," 2021. [Online]. Available: https://arxiv.org/abs/2111.13336

[31] W. Li, J. Liu, and H. Mei, "Lightweight convolutional neural network for aircraft small target real-time detection in airport videos in complex scenes," *Scientific Reports*, vol. 12, pp. 1–12, 2022.

[32] F. Hao, J. Wu, H. Lu, J. Du, J. Xu, and X. Xu, "Large coordinate kernel attention network for lightweight image super-resolution," 2024. [Online]. Available: https://arxiv.org/abs/2405.09353

[33] R. P. Duarte, C. Cunha, and J. C. Pereira Cardoso, "Automatic camera calibration using a single image to extract intrinsic and extrinsic parameters," *International Journal of Intelligent systems and Applications in Engineering*, vol. 12, no. 3, pp. 1766–1778, 2024.

[34] C. Gao, G. Jiang, J. Gao, and P. Li, "A rapid method for obtaining camera intrinsic parameters based on a stereo calibration target," *Optica Open*, 2024.

[35] Y. Li, "A calibration method of computer vision system based on dual attention mechanism," *Image and Vision Computing*, vol. 103, 2020.

[36] M. Ahmed, B. Rasheed, H. Salloum, M. Hegazy, M. R. Bahrami, and M. Chuchkalov, "Seal pipeline: Enhancing dynamic object detection and tracking for autonomous unmanned surface vehicles in maritime environments," *Drones*, vol. 8, no. 10, pp. 1–30, 2024.

[37] Z. Elri and A. Ergüzen, "Advancing defense capabilities through integration of electro-optical systems and computer vision technologies," *Journal of Computer & Electrical and Electronics Engineering Sciences*, vol. 2, no. 1, pp. 17–24, 2024.

[38] X. Sun, Y. Jiang, Y. Ji, W. Fu, S. Yan, Q. Chen, B. Yu, and X. Gan, "Distance measurement system based on binocular stereo vision," in *IOP Conference Series: Earth and Environmental Science*, vol. 252, no. 5. IOP Publishing, 2019, pp. 1–7.

[39] S. Daram, P. S. R. R. Mallidi, S. Vangipuram, P. S. Karpe, and L. O. G., "Advancing public transit security via OpenCV powered facial recognition & cloud authentication framework," 2024. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4818830

[40] H. Zhu, P. Zhang, L. Wang, X. Zhang, and L. Jiao, "A multiscale object detection approach for remote sensing images based on MSE-DenseNet and the dynamic anchor assignment," *Remote Sensing Letters*, vol. 10, no. 10, pp. 959–967, 2019.

[41] P. Costa and I. Ruan, "Optimization of geometric measures of sets of moving objects," Master's thesis, University of Manitoba, 2024.

[42] W. Zhao, L. Yan, and Y. Zhang, "Geometric-constrained multi-view image matching method based on semi-global optimization," *Geo-Spatial Information Science*, vol. 21, no. 2, pp. 115–126, 2018.