Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 4 (2024) 579 - 588 e-ISSN: 2580-0760

Augmentation for Accuracy Improvement of YOLOv8 in Blind Navigation System

Erwin Syahrudin[1], Ema Utami[2], Anggit Dwi Hartanto[3]
[1,2,3]Magister of Informatics, Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia
[1]erwinsyahrudin@students.amikom.ac.id, [2]ema.u@amikom.ac.id, [3]anggit@amikom.ac.id

Abstract

This study addresses the critical need for enhanced accuracy in YOLOv8 models designed for visually impaired navigation systems. Existing models often struggle with consistency in object detection and distance estimation under varying environmental conditions, leading to potential safety risks. To overcome these challenges, this research implements a rigorous approach combining data augmentation and meticulous model optimization techniques. The process begins with the meticulous collection of a diverse dataset, essential for training a robust model. Subsequent preprocessing of images in the HSV color space ensures standardized input features, crucial for consistency in model training. Augmentation techniques are then applied to enrich the dataset, enhancing model generalization and robustness. The YOLOv8 model is trained using this augmented dataset, leading to significant enhancements in key performance metrics. Specifically, mean average precision (mAP) improved by 13.3%, from 0.75 to 0.85, precision increased by 10%, from 0.80 to 0.88, and recall rose by 10.3%, from 0.78 to 0.86. Further optimization efforts, including parameter tuning and the strategic integration of a Kalman Filter, notably improved object tracking and distance estimation capabilities. Final validation in real-world scenarios confirms the efficacy of the optimized model, demonstrating its readiness for practical deployment. This comprehensive approach showcases tangible advances in navigational assistance technology, significantly improving safety and reliability for visually impaired users.

Keywords: YOLOv8; object detection; data augmentation; model optimization; visual impairment

How to Cite: E. Syahrudin, E. Utami, and A. D. Hartanto, "Augmentation for Accuracy Improvement of YOLOv8 in Blind Navigation System", J. RESTI (Rekayasa Sist. Teknol. Inf.), vol. 8, no. 4, pp. 579 - 588, Aug. 2024.
DOI: https://doi.org/10.29207/resti.v8i4.5931

1. Introduction

Object detection technology has revolutionized various fields, including assisting visually impaired individuals in navigating their surroundings independently. Among the cutting-edge solutions, the YOLOv8 model stands out for its efficiency in real-time object detection, making it particularly promising for enhancing accessibility tools for the visually impaired [1], [2]. This study focuses on optimizing the YOLOv8 model through advanced techniques such as data augmentation and model tuning, aimed at significantly improving its accuracy and reliability in real-world navigation scenarios [3].

The ability to detect and classify objects in real-time is crucial for visually impaired individuals to navigate unfamiliar environments safely. Traditional methods often struggle with processing speed and accuracy, limiting their practicality [4], [5]. YOLOv8 addresses these challenges by leveraging deep learning capabilities to detect multiple objects swiftly and accurately from live camera feeds or static images [6], [7]. By enhancing its performance through systematic optimization, this research seeks to elevate the model's capability to identify crucial elements such as obstacles, pedestrian crossings, and landmarks with heightened accuracy [8].

Central to this research is the process of data augmentation, which involves enriching the training dataset by manipulating images in ways that simulate real-world variations [9]. By introducing diverse perspectives, lighting conditions, and environmental contexts, the augmented dataset enables the YOLOv8 model to generalize better, enhancing its ability to recognize objects under various conditions encountered by visually impaired users [10]. This approach not only improves the model's robustness but also ensures reliable performance in diverse real-world settings.

Received: 18-07-2024 | Accepted: 26-08-2024 | Published Online: 31-08-2024

Moreover, model optimization plays a pivotal role in refining the YOLOv8 algorithm to achieve higher precision and recall rates [11]. Techniques such as parameter tuning and the integration of Kalman Filters for object tracking and distance estimation are explored to fine-tune the model's responsiveness and accuracy. These optimizations are essential for mitigating errors and enhancing the overall reliability of the YOLOv8 model, ensuring it provides accurate object detection results crucial for assisting visually impaired users in real-time navigation tasks [12].

The evaluation metrics used in this study, including mean Average Precision (mAP), Precision, and Recall, serve as benchmarks to quantify the model's performance improvements throughout the optimization process [13]. By systematically measuring these metrics before and after optimization, this research aims to demonstrate tangible advancements in the YOLOv8 model's accuracy and efficacy. Ultimately, these advancements are expected to contribute significantly to the development of more effective and dependable assistive technologies for enhancing the independence and safety of visually impaired individuals in navigating complex environments [14].

To further establish the context and underscore the significance of this research, it is important to acknowledge the contributions of previous studies that have applied earlier versions of the YOLO model to similar challenges. explored the application of YOLOv3 in dynamic environments, particularly focusing on real-time object detection in cluttered and rapidly changing settings. Despite the model's strengths in speed and versatility, challenges such as accurately detecting smaller objects and managing complex background noise persisted [15]. Subsequent advancements, including YOLOv4, [16], addressed some of these limitations by improving the balance between speed and accuracy, making it a more robust option for various applications, including assistive technologies [17].

However, these studies largely focused on general object detection without specifically addressing the unique requirements of visually impaired navigation systems, such as the need for precise distance estimation and reliable object tracking in varied lighting conditions. This gap in the literature highlights the urgency of developing a specialized solution that leverages the latest YOLOv8 advancements to cater to these specific needs. By building on these foundational studies and addressing their limitations, the current research introduces novel enhancements tailored to the context of assistive technologies for the visually impaired, thereby contributing to the broader discourse on improving navigational aids for this community.

This study not only seeks to optimize the YOLOv8 model for general object detection but also aims to fine-tune its application in real-world scenarios that are critical for visually impaired individuals. By integrating advanced data augmentation techniques and model optimization strategies, this research aspires to bridge the gap between the model's theoretical capabilities and its practical deployment in assistive technologies, ensuring that the visually impaired can navigate their environments with greater confidence and independence.

2. Research Methods

This section delineates the comprehensive research methodology adopted to enhance the accuracy and reliability of the YOLOv8 model for visually impaired navigation systems. It includes the system architecture, preprocessing techniques, data augmentation strategies, model training and optimization, evaluation metrics, and the overall flow of the research process, represented through a detailed flowchart.

2.1 Research Flowchart

The Research flowchart in Figure 1 provides a visual representation of the systematic approach taken to enhance the accuracy of the YOLOv8 model for visually impaired navigation systems. Each step in the flowchart is crucial for the overall improvement of the model's performance. Figure 1 is a detailed explanation of each step in the research flowchart:

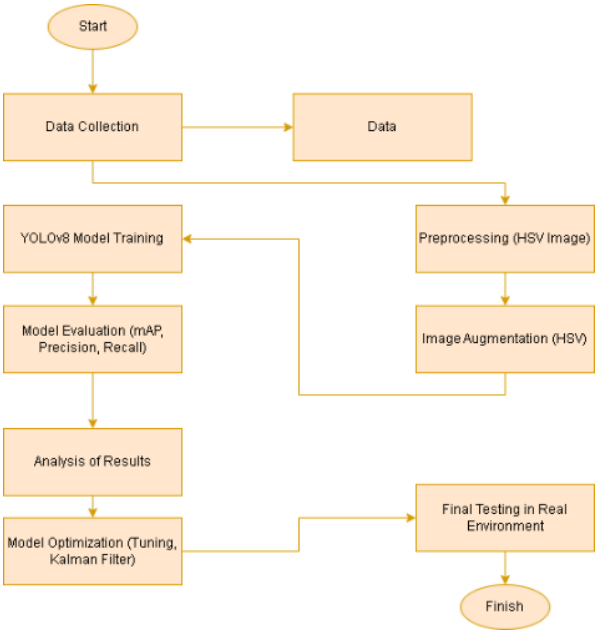

Figure 1. Research Flowchart

Data collection involves gathering a diverse set of images representing various environments and conditions encountered by visually impaired individuals [18]. This dataset encompasses a diverse array of images from both indoor and outdoor environments, meticulously capturing variations in lighting conditions, object scales, and perspectives [19]. For instance, indoor images might depict furniture, stairs, and walls from multiple angles and distances,

ensuring that the model can detect these objects regardless of their orientation or proximity. Similarly, outdoor images include sidewalks, street signs, and trees photographed from various perspectives and under different lighting conditions, such as direct sunlight, overcast skies, and low light scenarios. To further enhance the robustness of the dataset, special attention was given to the consistency and quality of the images. High-resolution cameras were used during the acquisition process, ensuring that all images maintain a consistent quality level across different scenes. The acquisition process was carefully designed to include multiple angles and scales, ensuring that objects of varying sizes are well-represented in the dataset. This approach guarantees that the model is trained to recognize and accurately detect objects regardless of their size, distance, or the angle from which they are viewed. By incorporating these considerations, the dataset effectively mirrors the complex and dynamic real-world environments that visually impaired individuals encounter daily, thereby laying a solid foundation for training a robust and reliable object detection model [20].

Preprocessing standardizes the input images to ensure consistent quality and format, which is essential for effective model training. This step involves several key operations. Firstly, all images are resized to a uniform dimension, typically 416x416 pixels, which is compatible with the input size expected by the YOLOv8 model [21]. Secondly, the images are converted from the RGB color space to the HSV (Hue, Saturation, Value) color space. The HSV color space is particularly useful for enhancing image features that are critical for object detection, as it separates chromatic content (color information) from intensity, making it easier to detect objects under varying lighting conditions. This conversion helps in improving the detection accuracy by focusing on color-based features that are less affected by shadows and lighting changes [22]. Data augmentation techniques are applied to the preprocessed images to increase the diversity of the training dataset. Augmentation involves a series of operations such as rotation, scaling, translation, and brightness adjustments [23]. For instance, images might be randomly rotated by up to 30 degrees, shifted horizontally or vertically by up to 10%, or scaled by a factor between 0.8 and 1.2. Additionally, brightness adjustments within a specified range ensure the model can handle varying lighting conditions. These augmentations simulate different environmental conditions and object orientations, helping the model generalize better by exposing it to a wider variety of scenarios [24], [25]. This step is critical to enhance the robustness of the model, enabling it to perform well on unseen data by learning to detect objects in diverse and unpredictable real-world conditions.

2.2 System Architecture

The flowchart Figure 2 illustrates the process of augmenting an image along with its bounding boxes using albumentations. It starts with loading the image and its corresponding bounding boxes. Next, an augmentation pipeline is defined, specifying various transformations such as rotation, flipping, and color adjustments. The augmentation is then applied to both the image and the bounding boxes. The augmented image, originally in tensor format, is converted to a numpy array to facilitate further processing. Bounding boxes are drawn on this augmented image, and finally, the augmented image with the bounding boxes is displayed. This process ensures that both the image and its bounding boxes are consistently transformed, maintaining the accuracy of object detection annotations.

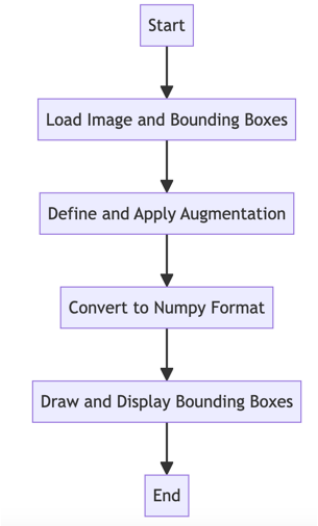

Figure 2. System Architecture

System architecture encapsulates a systematic approach to augmenting images and bounding boxes, essential for enriching training datasets used in object detection applications. By integrating advanced augmentation techniques and ensuring proper handling of image and bounding box data, the workflow enhances dataset variability and prepares robust models capable of accurately detecting objects across diverse real-world scenarios.

2.3 Data Augmentation Techniques

Table 1 summarizes the data augmentation techniques utilized in the context of the bounding box flowchart. These techniques are implemented using albumentations, a versatile library for image augmentation, ensuring both the image and its associated bounding boxes are transformed consistently.

In this study, a range of data augmentation techniques are employed to enhance the YOLOv8 model's robustness and generalization capabilities, specifically for visually impaired navigation systems. Techniques such as random rotation, flip, and transpose address variations in object orientation, while blur and motion blur simulate motion and environmental conditions to improve detection accuracy under dynamic

circumstances. Shift scale rotate and optical/grid distortion are applied to manage object position, scale, and lens distortions, ensuring the model remains reliable across different camera angles and environments. Additionally, adjustments in hue, saturation, brightness, and contrast prepare the model for varying lighting conditions encountered in both indoor and outdoor settings. Finally, resizing, normalization, and tensor conversion standardize the images for effective deep learning training. These carefully selected augmentations are crucial for creating a robust dataset that enhances the model's performance in real-world scenarios, providing reliable assistance for visually impaired individuals [26], [27].

Table 1. Detail Data Augmentation Techniques

Technique	Description
Random Rotate 90°	Rotates the image randomly by 0°, 90°, 180°, or 270°.
Flip	Flips the image horizontally or vertically with a certain probability.
Transpose	Transposes the image (flips rows and columns).
Motion Blur	Applies motion blur to the image with a random kernel size and direction.
Median Blur	Applies median blur to the image with a random kernel size.
Blur	Applies blur to the image with a random kernel size.
Shift Scale Rotate	Randomly shifts, scales, and rotates the image.
Optical Distortion	Distorts the image using random optical effects.
Grid Distortion	Distorts the image using random grid effects.
Hue Saturation Value	Adjusts hue, saturation, and value levels of the image.
Random Brightness Contrast	Adjusts brightness and contrast of the image randomly.
Resize	Resizes the image to a specified height and width.
Normalize	Normalizes the image pixel values.
ToTensorV2	Converts the image to PyTorch tensor format.

The described augmentation techniques play a crucial role in preparing a diverse and robust training dataset for object detection models. Random Rotate 90°, Flip, and Transpose operations introduce variations in image orientation, essential for teaching models to detect objects from different perspectives. Blur and Motion Blur simulate real-world motion and environmental conditions, improving the model's ability to discern objects under varying clarity levels [28]. Shift Scale Rotate diversifies the dataset by randomly adjusting image positions and sizes, enhancing the model's adaptability to different scales and perspectives [29]. Optical and Grid Distortion techniques mimic complex real-world distortions, preparing models to handle non-linear image deformations effectively [30]. Adjusting Hue, Saturation, and Brightness ensures the model can recognize objects under diverse lighting conditions. Finally, resizing, normalizing pixel values, and converting to tensor format ensures compatibility with deep learning frameworks like PyTorch, enabling efficient model training. Together, these augmentations enhance the dataset's richness and prepare both images and bounding box annotations systematically for improved model accuracy and reliability in practical object detection applications [31].

Figure 3. Original Image

The original images collected from the dataset reflect the objects in their standard conditions without modification. For example, the original Figure 3 might show the object in normal lighting, fixed orientation, and without distortion or color changes.

The augmented image in Figure 4 is the result of various transformations applied to the original image. These transformations can include rotations, shifts, brightness changes, distortions, and more.

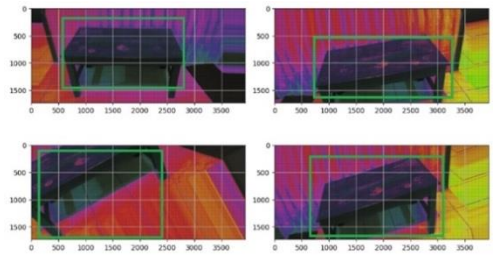

Figure 4. Augmented Image

This data augmentation process collectively enriches the training dataset, providing a variety of images to improve the accuracy and robustness of the model in detecting objects with proper bounding box annotations. By applying this augmentation, the YOLOv8 model can be more effective in recognizing objects in various real-world conditions, improving object detection performance and reliability in navigation applications for the blind.

To execute the augmented image processing tasks effectively, a common approach is to use Jupyter Notebook in conjunction with powerful libraries like OpenCV, Albumentations, and PyTorch. Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and explanatory text. It's ideal for interactive data science and machine learning projects, enabling easy experimentation with different code snippets and visualizations.

OpenCV (Open Source Computer Vision Library) is widely used for real-time computer vision tasks. It provides a comprehensive set of tools for image processing, including functions for reading, writing, and manipulating images, which are crucial for tasks such as object detection and augmentation.

Albumentations is a Python library that specializes in fast and flexible image augmentations. It provides a rich set of operations to enhance the diversity of training datasets by applying transformations like rotation, flipping, scaling, and color adjustments. These augmentations are vital for improving the generalization capability of machine learning models, especially in object detection tasks.

PyTorch is a deep learning framework that offers robust support for tensor computation and GPU acceleration. It is highly regarded for its flexibility and efficiency in building and training neural networks. PyTorch integrates seamlessly with augmentation libraries like Albumentations and can be used to develop and train object detection models like YOLOv8, which are optimized to assist visually impaired individuals by accurately identifying and tracking objects in real-time.

2.4 Model Optimization

Model optimization in object detection in Figure 3, such as in the case of integrating Kalman Filters and fine-tuning parameters, plays a critical role in enhancing the accuracy and efficiency of detection systems. Kalman Filters are pivotal for improving object tracking and distance estimation by predicting and correcting the state of detected objects across frames. This recursive filter is particularly effective in handling noisy data, ensuring smoother trajectories and more accurate predictions over time. By continuously updating the positions and velocities of objects, Kalman Filters enhance the reliability of object tracking in dynamic environments, crucial for applications like autonomous vehicles and surveillance systems [32] as shown in Figure 5.

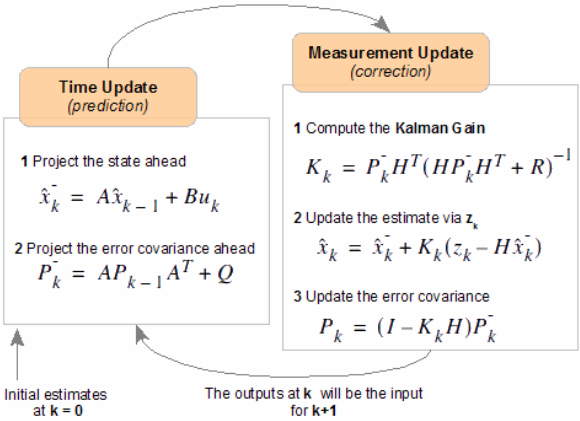

Figure 5. Kalman Filter

The optimization of the YOLOv8 model's object tracking and distance estimation capabilities is significantly enhanced by integrating the Kalman Filter. The Kalman Filter is a recursive algorithm that provides estimates of unknown variables by predicting a system's future state based on its current state and correcting it with observed data. This process is essential for improving the accuracy and reliability of the model, particularly in dynamic environments where real-time tracking is crucial.

Kalman Filter Process in Time Update (Prediction) is State Prediction. State Prediction is the state of the system (e.g., position and velocity of an object) is projected ahead using the state transition model is shown in Formula 1.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (1)$$

$\hat{x}_k^-$ is the predicted state, A is the state transition matrix, $\hat{x}_{k-1}$ is the previous state estimate, B is the control input model, and $u_k$is the cotrol input.

Kalman Filter Process in Covariance Prediction. Covariance Prediction is the error covariance is projected ahead to estimate the uncertainty of the state prediction is shown in Formula 2

$$P_k^- = AP_{k-1}A^T + Q \quad (2)$$

P_k^- is the predicted error covariance, P_{k-1} is the previous error covariance, A is the state transition matrix, and Q is the process noise covariance.

Kalman Filter Process in Measurement Update (Correction) stages are:

Kalman Gain Calculation: The Kalman Gain is computed, which determines the weight given to the measurement update is shown in Formula 3.

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3)$$

K_k is the Kalman Gain, P_k^- is the predicted error covariance, H is the observation model, and R is the measurement noise covariance.

State Update: The predicted state is corrected using the new measurement is shown in Formula 4.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (4)$$

$\hat{x}_k$ is the updated state estimate, z_k is the measurement, and $H\hat{x}_k^-$ is the predicted measurement.

Covariance Update: The error covariance is updated to reflect the improved state estimate is shown in Formula 5.

$$P_k = (I - K_k H)P_k^- \quad (5)$$

P_k is the updated error covariance, and I is the identity matrix.

In this research, the Kalman Filter is integrated into the YOLOv8 model to improve its object tracking and distance estimation performance. By tuning the filter's parameters (such as the process noise covariance Q and measurement noise covariance R), the model becomes more adept at predicting and correcting the state of detected objects, even in the presence of noise and

uncertainty. This optimization is particularly beneficial in visually impaired navigation systems, where accurate and reliable tracking of objects is essential for safe and effective guidance.

3. Results and Discussions

In using this augmentation technique, it should be noted that this approach must be balanced with experimental setup, accurate data collection, to obtain optimal results.

3.1 Experimental Setup

Detailed requirements used in this study to optimize the YOLOv8 model are summarized in Table 2. The hardware components used include NVIDIA RTX 3060 for graphics processing, 16GB RAM, and Intel i7-12700K CPU, which allow fast and efficient data processing. The software used is Python 3.8 as the main programming language, TensorFlow 2.4 for the deep learning framework, and OpenCV 4.5 for image processing. The dataset used consists of 5000 specific images that provide data diversity for model training. Model performance evaluation is carried out using the mean Average Precision (mAP), Precision, and Recall metrics to ensure the model can detect and classify objects with high accuracy after the optimization and data augmentation processes are applied.[33], [34], [35].

Table 2 Detail Requirements

Components	Description
Hardware	NVIDIA RTX 3060, 16GB RAM, Intel i7-12700K CPU
Software	Python 3.8, TensorFlow 2.4, OpenCV 4.5
Dataset	Custom dataset with 5000 images
Evaluation Metrics	mAP, Precision, Recall

The dataset utilized in the study consisted of 5000 custom images specifically curated to represent diverse real-world scenarios encountered in visually impaired navigation. This dataset was crucial for training and evaluating the YOLOv8 model's performance across different environmental conditions and object types. Evaluation metrics focused on mean Average Precision (mAP), Precision, and Recall, providing quantitative measures of the model's detection accuracy, reliability, and ability to minimize false positives and negatives [36], [37], [38].

3.2 Evaluation Metrics

Table 3 compares the performance metrics of the YOLOv8 model before and after optimization efforts. Before optimization, the model achieved an initial mean Average Precision (mAP) of 0.75, Precision of 0.80, and Recall of 0.78. These metrics represent the baseline performance of the model, indicating its initial capability in object detection and classification tasks for visually impaired navigation systems.

Table 3 Detail Before and After Optimization

Metric	Before Optimization	After Optimization
mAP	0.75	0.85
Precision	0.80	0.88
Recall	0.78	0.86

mAP is a metric that measures the average precision across different levels of recall. For object detection systems, *mAP* measures how well a model can correctly find and classify objects [39].

Mean Average Precision (*mAP*) is shown in Formula 6.

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_I \quad (6)$$

n is the number of object classes detected, and AP_i is the Average Precision for each object class.

Precision is shown in Formula 7.

$$Precision = \frac{True\ Positives}{True\ Positive+False\ Positive} \quad (7)$$

Precision measures the ratio between the number of correct positive predictions to the total number of positive predictions made by the model [40].

Recall is shown in Formula 8.

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (8)$$

Recall (also known as sensitivity) measures the ratio between the number of correct positive predictions to the total number of instances that are actually positive[41].

Following rigorous optimization procedures, including data augmentation, parameter tuning, and the integration of advanced techniques such as Kalman Filters, significant improvements were observed in the model's performance metrics. The mAP increased from 0.75 to 0.85, reflecting a higher precision in localizing objects within images. Precision also improved from 0.80 to 0.88, indicating a reduction in false positives and enhanced accuracy in object detection. Moreover, Recall rose from 0.78 to 0.86, indicating an improved ability to detect objects across various environmental conditions.

3.3 Final Testing Result

Table 4 summarizes the final test results of the optimized YOLOv8 model in various real-world scenarios. The tests were designed to evaluate the model's performance under different environmental conditions, including indoor low light, outdoor bright light, and mixed lighting conditions. The metrics used for evaluation include mean Average Precision (mAP), Precision, and Recall, which are important indicators of the model's object detection and classification performance.

Table 4 Final Testing Result

Scenario	mAP	Precision	Recall
Indoor, low light	0.82	0.87	0.84
Outdoor, daylight	0.86	0.90	0.88
Mixed lighting conditions	0.84	089	0.85

in the final test results show a significant improvement the performance of the YOLOv8 model after the optimization process. In the indoor low-light scenario, the model achieved an mAP of 0.82, a Precision of 0.87, and a Recall of 0.84. This shows that the model can detect objects quite well even in less-than-optimal lighting conditions.

In the outdoor bright lighting scenario, the model performed the best with an mAP of 0.86, a Precision of 0.90, and a Recall of 0.88. These results indicate that the model is very effective in detecting objects in optimal lighting conditions, with a very low error rate.

The mixed lighting scenario refers to environments where lighting conditions vary significantly within the same scene, which can include a combination of bright, dim, and shadowed areas. This could occur, for example, in outdoor environments where some areas are directly illuminated by sunlight while others are shaded by buildings or trees, or in indoor settings where artificial lighting creates both well-lit and poorly lit areas.

In this study, mixed lighting scenarios are particularly important as they simulate the diverse and unpredictable lighting conditions that visually impaired individuals may encounter in real-world navigation. The criteria for a mixed lighting scenario in the evaluation include:

Presence of Both Bright and Dim Areas: The image must contain regions with varying levels of brightness, such as direct sunlight juxtaposed with shadowed areas.

Dynamic Range of Illumination: The scenario should encompass a wide range of illumination intensities, from high brightness to low-light regions.

Varied Light Sources: The scenario may involve different types of light sources, such as natural sunlight combined with artificial indoor lighting, or reflections that create additional variability in illumination.

The optimization of the YOLOv8 model is crucial for ensuring reliable object detection across these varying lighting conditions. The reported performance metrics—an mAP of 0.84, Precision of 0.89, and Recall of 0.85 in the mixed lighting scenario—demonstrate the model's enhanced robustness and ability to maintain accuracy even when lighting conditions are inconsistent. This makes the model more adaptable to real-world applications where such lighting variability is common.

Figure 6 is a graph depicting the evaluation metrics before and after optimization of the YOLOv8 model. Above the line graph in Figure 5, we can see a direct comparison between the evaluation metrics of the YOLOv8 model before and after the optimization process. The comparison of the YOLOv8 model's performance before and after optimization is essential to quantify the impact of the optimization techniques applied, including data augmentation and parameter tuning. Before optimization, the model achieved an mAP of 0.75, a precision of 0.80, and a recall of 0.78. These baseline metrics reflect the model's initial ability to detect objects with a certain level of accuracy, precision, and sensitivity to true positives.

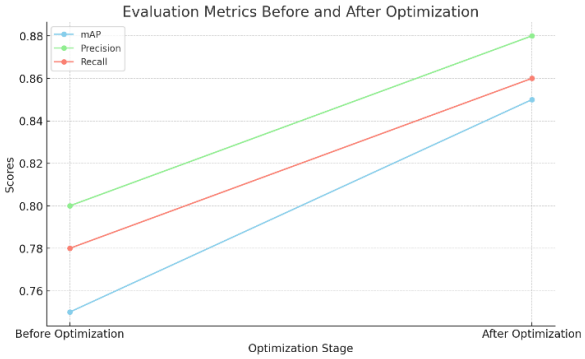

Figure 6 Result Graph Metrics Evaluation

However, these values also highlight areas for improvement, particularly in terms of the model's generalization ability across varied real-world scenarios, which is critical for assisting visually impaired users in navigation tasks. The need for optimization arises from the objective to enhance the model's robustness, ensuring that it can perform reliably across different lighting conditions, object scales, and environmental contexts that visually impaired individuals might encounter.

After implementing the optimization strategies, which included enriching the dataset through data augmentation and fine-tuning model parameters, a significant improvement was observed in all key performance metrics. The mAP increased to 0.85, precision rose to 0.88, and recall improved to 0.86. This comparison clearly demonstrates the effectiveness of the optimizations, validating the necessity of these interventions to achieve a more accurate, precise, and reliable model. The before-and-after comparison is crucial as it provides a measurable validation of the research efforts, showcasing how targeted improvements can lead to substantial gains in the model's performance, making it better suited for real-world deployment in assistive technologies for the visually impaired.

Figure 7 is a bar chart depicting the final test results of the optimized YOLOv8 model in various real-world scenarios. The final test results of the optimized YOLOv8 model showcase its substantial potential in practical applications, particularly in enhancing navigation systems for visually impaired individuals. The model demonstrated its effectiveness across various lighting conditions, achieving an mAP of 0.82, 0.86, and 0.84 in indoor low-light, outdoor daylight, and mixed lighting conditions, respectively. These metrics indicate that the model has been effectively optimized to handle the diverse environments that visually impaired individuals might encounter in their daily navigation, making it a reliable tool for real-world use.

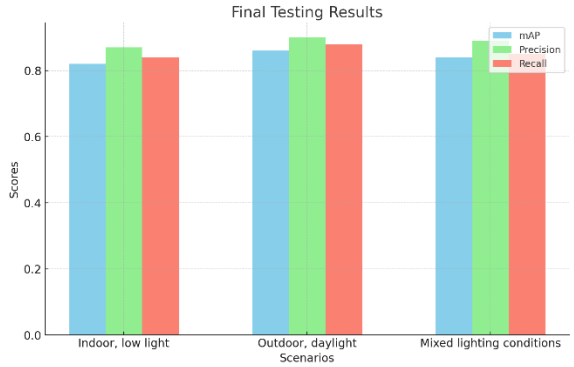

Figure 7. Result Graph of Final Result

The integration of data augmentation techniques significantly contributed to the model's improved accuracy. By enriching the training dataset with variations in lighting, perspectives, and object scales, the model has learned to generalize better across different scenarios, reducing errors and increasing reliability. This adaptability is crucial for the practical application of navigation systems, where environmental conditions are unpredictable and varied.

The practical applications of this optimized model extend to developing assistive navigation systems that can be embedded in wearable devices, mobile applications, or integrated into smart environments. These systems would leverage the well-validated dataset and robust object detection capabilities to provide real-time feedback to visually impaired users, helping them navigate safely and independently. For instance, a mobile application could alert users to obstacles, guide them through pedestrian crossings, or help them identify landmarks, thereby significantly enhancing their autonomy and quality of life. The readiness of the model for such applications is underscored by its validated performance across key metrics, confirming its potential to be a cornerstone in developing advanced assistive technologies for the visually impaired.

4. Conclusions

This study demonstrates that through strategic data augmentation and model optimization, including Kalman Filter integration, the YOLOv8 model significantly enhances its accuracy for blind navigation systems. The evaluation results reveal a notable improvement in mean Average Precision (mAP) from 0.75 to 0.85, Precision from 0.80 to 0.88, and Recall from 0.78 to 0.86 after optimization. Testing in diverse real-world conditions, including low indoor lighting, bright outdoor settings, and mixed environments, confirms the model's robustness and reliability in practical applications. These findings highlight the model's potential to provide dependable navigation assistance for visually impaired individuals, ensuring stable object detection across various scenarios [42]. Future research should focus on further refining the model to adapt to even more complex environments and explore the integration of additional sensory inputs, such as audio feedback, to enhance user experience. The social and technological implications of this research are profound, as it contributes significantly to improving the quality of life for visually impaired individuals by enabling greater independence and safety in navigation. Continued development in this area could lead to more sophisticated assistive technologies, fostering inclusivity and accessibility in society.

Acknowledgements

The authors would like to acknowledge valuable funding provided by Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi in Hibah Penelitian Pascasarjana - Penelitian Tesis Magister of the fiscal year 2024.

References

[1] 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. 63754–63767, 2024, doi: 10.1109/ACCESS.2024.3396224.

[2] M. Safaldin, N. Zaghden, and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," *IEEE Access*, vol. 12, pp. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.

[3] S. Muhamad Itikap, M. Syahid Abdurrahman, E. B. Soewono, and T. Gelar, "Geometry and Color Transformation Data Augmentation for YOLOV8 in Beverage Waste Detection," *Journal of Software Engineering, Information and Communication Technology (SEICT)*, vol. 4, no. 2, pp. 123–138, 2023, doi: 10.17509/seict.

[4] A. Aboah, B. Wang, U. Bagci, and Y. Adu-Gyamfi, "Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.08256

[5] M. I. Thariq Hussan, D. Saidulu, P. T. Anitha, A. Manikandan, and P. Naresh, "Object Detection and Recognition in Real Time Using Deep Learning for Visually Impaired People," *International Journal of Electrical and Electronics Research*, vol. 10, no. 2, pp. 80–86, 2022, doi: 10.37391/IJEER.100205.

[6] 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, Jul. 2024, doi: 10.1016/j.neucom.2024.127685.

[7] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition," *Drones*, vol. 7, no. 5, May 2023, doi: 10.3390/drones7050304.

[8] Y. Lei, S. L. Phung, A. Bouzerdoum, H. Thanh Le, and K. Luu, "Pedestrian Lane Detection for Assistive Navigation of Vision-Impaired People: Survey and Experimental Evaluation," *IEEE Access*, vol. 10, pp. 101071–101089, 2022, doi: 10.1109/ACCESS.2022.3208128.

[9] J. Ye, Z. Yuan, C. Qian, and X. Li, "CAA-YOLO: Combined-Attention-Augmented YOLO for Infrared

Ocean Ships Detection," *Sensors*, vol. 22, no. 10, May 2022, doi: 10.3390/s22103782.

[10] S. X. Tan, J. Y. Ong, K. Ong, M. Goh, and C. Tee, "Boosting Vehicle Classification with Augmentation Techniques across Multiple YOLO Versions," *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION*, vol. 8, no. 1, pp. 45–54, 2024, [Online]. Available: www.joiv.org/index.php/joiv

[11] D. Kumar and N. Muhammad, "Object Detection in Adverse Weather for Autonomous Driving through Data Merging and YOLOv8," *Sensors (Basel)*, vol. 23, no. 20, Oct. 2023, doi: 10.3390/s23208471.

[12] A. Inui *et al.*, "Detection of Elbow OCD in the Ultrasound Image by Artificial Intelligence Using YOLOv8," *Applied Sciences (Switzerland)*, vol. 13, no. 13, Jul. 2023, doi: 10.3390/app13137623.

[13] L. Da Quach, K. N. Quoc, A. N. Quynh, and H. T. Ngoc, "Evaluating the Effectiveness of YOLO Models in Different Sized Object Detection and Feature-Based Classification of Small Objects," *Journal of Advances in Information Technology*, vol. 14, no. 5, pp. 907–917, 2023, doi: 10.12720/jait.14.5.907-917.

[14] E. Casas, L. Ramos, C. Romero, and F. Rivas-Echeverría, "A comparative study of YOLOv5 and YOLOv8 for corrosion segmentation tasks in metal surfaces," *Array*, p. 100351, Jun. 2024, doi: 10.1016/j.array.2024.100351.

[15] B. Zakariya, "System for detecting social distance during COVID-19 using YOLOv3 and OpenCV," 2022, [Online]. Available: https://www.researchgate.net/publication/365098883

[16] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," Jul. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/machines11070677.

[17] 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, Dec. 2021, doi: 10.3390/e23121587.

[18] J. Kaur and W. Singh, "Tools, techniques, datasets and application areas for object detection in an image: a review," *Multimed Tools Appl*, vol. 81, no. 27, pp. 38297–38351, Nov. 2022, doi: 10.1007/s11042-022-13153-y.

[19] Z. Zong, G. Song, and Y. Liu, "DETRs with Collaborative Hybrid Assignments Training," *ArXiv*, vol. 2211, no. 1286v5, pp. 1–13, Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.12860

[20] S. Norkobil Saydirasulovich, A. Abdusalomov, M. K. Jamil, R. Nasimov, D. Kozhamzharova, and Y. I. Cho, "A YOLOv6-Based Improved Fire Detection Approach for Smart City Environments," *Sensors*, vol. 23, no. 6, Mar. 2023, doi: 10.3390/s23063161.

[21] P. Azevedo and V. Santos, "Comparative analysis of multiple YOLO-based target detectors and trackers for ADAS in edge devices," *Rob Auton Syst*, vol. 171, Jan. 2024, doi: 10.1016/j.robot.2023.104558.

[22] T. J. Alahmadi, A. U. Rahman, H. K. Alkahtani, and H. Kholidy, "Enhancing Object Detection for VIPs Using YOLOv4_Resnet101 and Text-to-Speech Conversion Model," *Multimodal Technologies and Interaction*, vol. 7, no. 8, Aug. 2023, doi: 10.3390/mti7080077.

[23] S. Y. Lin and H. Y. Li, "Integrated Circuit Board Object Detection and Image Augmentation Fusion Model Based on YOLO," *Front Neurorobot*, vol. 15, Nov. 2021, doi: 10.3389/fnbot.2021.762702.

[24] S. X. Tan, J. Y. Ong, K. Ong, M. Goh, and C. Tee, "Boosting Vehicle Classification with Augmentation Techniques across Multiple YOLO Versions," *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION*, pp. 45–54, 2024, [Online]. Available: www.joiv.org/index.php/joiv

[25] D. K. Baroroh, C. H. Chu, and L. Wang, "Systematic literature review on augmented reality in smart manufacturing: Collaboration between human and computational intelligence," *J Manuf Syst*, vol. 61, pp. 696–711, Oct. 2021, doi: 10.1016/j.jmsy.2020.10.017.

[26] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information (Switzerland)*, vol. 11, no. 2, Feb. 2020, doi: 10.3390/info11020125.

[27] A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," Sep. 2018, doi: 10.3390/info11020125.

[28] S. E. Ryu and K. Y. Chung, "Detection model of occluded object based on yolo using hard-example mining and augmentation policy optimization," *Applied Sciences (Switzerland)*, vol. 11, no. 15, Aug. 2021, doi: 10.3390/app11157093.

[29] F. J. Du and S. J. Jiao, "Improvement of Lightweight Convolutional Neural Network Model Based on YOLO Algorithm and Its Research in Pavement Defect Detection," *Sensors*, vol. 22, no. 9, May 2022, doi: 10.3390/s22093537.

[30] 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, Sep. 2023, doi: 10.3390/f14091885.

[31] W. Zhao, M. Syafrudin, and N. L. Fitriyani, "CRAS-YOLO: A Novel Multi-Category Vessel Detection and Classification Model Based on YOLOv5s Algorithm," *IEEE Access*, vol. 11, pp. 11463–11478, 2023, doi: 10.1109/ACCESS.2023.3241630.

[32] D. N. Triwibowo, E. Utami, Sukoco, and S. Raharjo, "Analysis of Classification and Calculation of Vehicle Type at APILL Intersection Using YOLO Method and Kalman Filter," in *3rd International Conference on Cybernetics and Intelligent Systems, ICORIS, IEEE*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICORIS52787.2021.9649607.

[33] R. Arifando, S. Eto, and C. Wada, "Improved YOLOv5-Based Lightweight Object Detection Algorithm for People with Visual Impairment to Detect Buses," *Applied Sciences (Switzerland)*, vol. 13, no. 9, May 2023, doi: 10.3390/app13095802.

[34] S. Wang, "Research towards Yolo-Series Algorithms: Comparison and Analysis of Object Detection Models for Real-Time UAV Applications," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jun. 2021. doi: 10.1088/1742-6596/1948/1/012021.

[35] 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. VIII, no. I, pp. 893–897, 2023, [Online]. Available: www.zkginternational.com

[36] Y. Hui, J. Wang, and B. Li, "DSAA-YOLO: UAV remote sensing small target recognition algorithm for YOLOV7 based on dense residual super-resolution and anchor frame adaptive regression strategy," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 1, Jan. 2024, doi: 10.1016/j.jksuci.2023.101863.

[37] Y. Cao, Z. Liu, F. Wang, S. Su, Y. Sun, and W. Wang, "An improved YOLOv7 for the state identification of sliding chairs in railway turnout," *High-speed Railway*, Apr. 2024, doi: 10.1016/j.hspr.2024.04.002.

[38] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2023.

[39] M. Konaite *et al.*, "Smart Hat for the blind with Real-Time Object Detection using Raspberry Pi and TensorFlow Lite," Association for Computing Machinery (ACM), Dec. 2021, pp. 1–6. doi: 10.1145/3487923.3487929.

[40] A. Hendrawan, R. Gernowo, O. D. Nurhayati, and C. Dewi, "A Novel YOLO-ARIA Approach for Real-Time Vehicle Detection and Classification in Urban Traffic," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 1, pp. 428–446, 2024, doi: 10.22266/ijies2024.0229.38.

[41] I. Purwita Sary, E. Ucok Armin, S. Andromeda, E. Engineering, and U. Singaperbangsa Karawang, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection Using Aerial Images," *Ultima Computing : Jurnal Sistem Komputer*, vol. 15, no. 1, 2023.

[42] J. Yan *et al.*, "Enhanced object detection in pediatric bronchoscopy images using YOLO-based algorithms with CBAM attention mechanism," *Heliyon*, vol. 10, no. 12, Jun. 2024, doi: 10.1016/j.heliyon.2024.e32678.

