informatics

MDPI

Article

Improved YOLOv10 for Visually Impaired: Balancing Model Accuracy and Efficiency in the Case of Public Transportation

Rio Arifando [1], Shinji Eto [1], Tibyani Tibyani [2] and Chikamune Wada [1,*]

[1] Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2–4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan; arifandorio@gmail.com (R.A.); eto.shinji786@mail.kyutech.jp (S.E.)
[2] Department of Information Systems, Faculty of Computer Science, Brawijaya University, Malang 65145, Indonesia; tibyani@ub.ac.id
* Correspondence: wada@brain.kyutech.ac.jp

Abstract: Advancements in automation and artificial intelligence have significantly impacted accessibility for individuals with visual impairments, particularly in the realm of bus public transportation. Effective bus detection and bus point-of-view (POV) classification are crucial for enhancing the independence of visually impaired individuals. This study introduces the Improved-YOLOv10, a novel model designed to tackle challenges in bus identification and pov classification by integrating Coordinate Attention (CA) and Adaptive Kernel Convolution (AKConv) into the YOLOv10 framework. The Improved YOLOv10 advances the YOLOv10 architecture through the incorporation of CA, which enhances long-range dependency modeling and spatial awareness, and AKConv, which dynamically adjusts convolutional kernels for superior feature extraction. These enhancements aim to improve both detection accuracy and efficiency, essential for real-time applications in assistive technologies. Evaluation results demonstrate that the Improved-YOLOv10 offers significant improvements in detection performance, including better Accuracy, Precision and Recall compared to YOLOv10. The model also exhibits reduced computational complexity and storage requirements, highlighting its efficiency. While the classification results show some trade-offs, with slightly decreased overall F1 score, the complexity of Giga Floating Point Operations (GFLOPs), Parameters, and Weight/MB in the Improved-YOLOv10 remains advantageous for classification tasks. The model's architectural improvements contribute to its robustness and efficiency, making it a suitable choice for real-time applications and assistive technologies.

Keywords: YOLOv10; coordinate attention; adaptive Kernel convolution; bus detection; POV classification; assistive technology; visually impaired navigation; deep learning; object detection; model efficiency

check for updates

Academic Editor: Antony Bryant

Received: 7 November 2024
Revised: 27 December 2024
Accepted: 6 January 2025
Published: 16 January 2025

Citation: Arifando, R.; Eto, S.; Tibyani, T.; Wada, C. Improved YOLOv10 for Visually Impaired: Balancing Model Accuracy and Efficiency in the Case of Public Transportation. *Informatics* **2025**, *12*, 7. https://doi.org/10.3390/informatics12010007

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Technological advancements in automation and artificial intelligence are transforming various sectors, including healthcare, security, and transportation. For individuals with visual impairments, these innovations have been particularly impactful, enhancing both accessibility and independence. According to the World Health Organization (WHO), more than 2.2 billion people globally suffer from some form of visual impairment, with around 285 million facing severe mobility challenges in their daily lives [1]. One of the most significant issues is accessing reliable public transportation information—such as bus schedules and routes—which often requires dependence on others and limits personal

autonomy [2]. Assistive technologies, such as those discussed in Assistive Technology for Visually Impaired and Blind People, are crucial for improving mobility for visually impaired individuals. Efficient transportation systems, equipped with such technology, enable accurate bus identification and help users navigate bus stops safely, ensuring they can board the correct bus and position themselves appropriately, thereby enhancing their overall travel experience [3]. Bus detection and bus point-of-view (POV) classification are particularly important for visually impaired individuals for several reasons. Accurate bus detection ensures that individuals can identify and board the correct bus, reducing confusion and enhancing safety. Bus POV classification, on the other hand, helps visually impaired passengers navigate bus stops and determine the optimal position to board the bus, which is crucial in ensuring they are in the right place when the bus arrives.

Technological advancements have significantly improved object detection systems, driving innovations in bus detection methods. Traditional approaches, such as Radio Frequency Identification (RFID) and embedded systems, have been utilized in intelligent transportation systems but are constrained by hardware dependency and environmental factors [4–7]. Recent research has focused on bus POV classification systems that use computer vision algorithms to analyze visual cues and determine optimal viewpoints for individuals at bus stops, thereby aiding effective navigation [8]. Although there has been progress in POV classification, these methods are less developed compared to general bus detection techniques. Building on these advancements, our research employs state-of-the-art algorithms and leverages smartphone technology to create portable, user-friendly solutions for navigating public transportation. By addressing the challenges faced by visually impaired individuals, such as improving their positioning at bus stops and enhancing bus identification accuracy, our work aims to increase independence and mobility for this community.

To achieve this, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have been central to recent innovations, learning hierarchical features from raw pixel data. Models such as Yolov10 [9], EfficientNet [10], DenseNet [11], MobileNet [12], InceptionNet [13], and Xception [14] have set new benchmarks. Recent advancements in attention mechanisms, such as Squeeze-and-Excitation Networks (SENets) [15], Convolutional Block Attention Module (CBAM) [16], and Coordinate Attention [17], have further enhanced feature representation by focusing on important regions of the input data and improving the model's ability to handle complex visual cues. Additionally, advanced convolutional networks like Depthwise Separable Convolutions (DSConv) [18] and Adaptive Kernel Convolution (AKConv) [19] have introduced efficient methods for capturing information across varying scales and details, contributing to improved detection accuracy and speed. Object detection algorithms have similarly benefited from these advancements, with significant improvements in both accuracy and speed. These algorithms are typically categorized into two-stage and one-stage methods. Two-stage algorithms, including R-FCN [20], Mask R-CNN [21], and Cascade R-CNN [22], initially generate region proposals and then apply CNNs for precise detection and classification. In contrast, one-stage algorithms, such as YOLOv4 [23], EfficientDet [24] and CenterNet [25], perform faster processing but generally lower accuracy.

Our research addresses these challenges by integrating advanced deep learning techniques into YOLOv10 [9], a leading model in real-time object detection. YOLOv10 builds upon the YOLO (You Only Look Once) series, known for its end-to-end object detection capabilities with enhanced efficiency and accuracy [yolov10]. It improves upon its predecessors through architectural advancements, including a unified dual-allocation strategy for Non-Maximum Suppression (NMS)-free training and an optimized design for both efficiency and accuracy.

To further enhance YOLOv10's performance, we incorporate the Coordinate Attention (CA) mechanism, which improves feature expression by capturing both channel-wise and spatial dependencies within feature maps [17]. This mechanism extends traditional attention models by addressing spatial information in addition to channel data, which is crucial for detecting small objects and handling complex backgrounds. The CA mechanism's ability to integrate coordinate-wise interactions allows for more precise object localization and detection, particularly in challenging environments. Additionally, we integrate Adaptive Kernel Convolution (AKConv) to overcome the limitations of fixed convolutional kernels. AKConv enables dynamic adjustments in kernel shapes and sizes, allowing the model to better capture information across varying scales and details [19]. This adaptability improves feature extraction by aligning kernel shapes with the input image's characteristics, enhancing detection precision and efficiency in complex and dynamic settings.

The enhancements to YOLOv10, including the substitution of Multi-Head Self-Attention (MHSA) with Coordinate Attention (CA) and the integration of AKConv, significantly improve the model's performance. CA enhances long-range dependency modeling and spatial awareness, while AKConv provides flexibility in feature extraction. These modifications result in a more robust and efficient model, capable of high-accuracy and real-time processing, making it particularly suited for applications that assist visually impaired individuals in navigating public transportation.

The following bullet points summarize the key contributions of the research study, which proposes a lightweight and high-accuracy bus detection model:

- Integration of Coordinate Attention (CA): Replaced Multi-Head Self-Attention (MHSA) with CA in the Partial Self-Attention (PSA) module, improving long-range dependency modeling and spatial awareness for better object localization and detection.
- Incorporation of Adaptive Kernel Convolution (AKConv): Substituted standard convolution layers with AKConv, enabling dynamic kernel adjustments to better capture information across varying scales and details, enhancing detection precision and efficiency.
- Application to Visually Impaired Navigation: Developed a model specifically tailored to enhance accessibility and independence for visually impaired individuals in public transportation systems, leveraging state-of-the-art deep learning techniques for real-time, high-accuracy detection.

The paper is organized into four key sections, each addressing critical aspects of the proposed bus detection model. Section 2 details the YOLOv10 and Improved-YOLOv10 methodologies, offering a thorough analysis of the technical features of the proposed model. Next, Section 3 describes the training process and experimental outcomes of the model, including a discussion of various tests conducted to assess the model's performance. Finally, Section 4 wraps up the paper by summarizing the major findings and contributions of the research, emphasizing the model's strengths such as its efficiency and accuracy.

2. Materials and Methods

2.1. Datasets

In this study, the dataset is designed to fulfill two distinct yet complementary objectives: bus detection and point-of-view (POV) classification. Bus detection involves identifying and locating buses, which is crucial for visually impaired individuals to safely and accurately identify public transportation options. Conversely, POV classification focuses on determining the effectiveness of the camera's perspective, classifying it as either "Good POV" or "Bad POV". This classification assesses whether the viewpoint offers a clear and balanced view, crucial for effective navigation at bus stops.

It consists of a diverse set of images captured under various environmental conditions, including lighting, weather, and background variations. The data was collected using mobile phone cameras and the internet, with the objective of aiding visually impaired individuals by improving their ability to detect nearby buses and navigate bus stops effectively. The bus detection dataset includes images that depict buses in different positions, orientations, and environmental conditions, focusing on three main object classes: buses, and routes. Figure 1 illustrates the diversity of urban environments present in the bus detection dataset.



Figure 1. Diversity of Urban Environments in Bus Detection Dataset.

Figure 2a illustrates the number of annotations per class within the dataset. Figure 2b visualizes the distribution of bounding boxes by showing their locations and sizes, which helps assess the variety in their placement and dimensions to ensure effective object recognition. Figure 2c,d present the statistical distribution of bounding box positions and sizes, depicting how these boxes are spread across the dataset. Figure 2e provides a detailed analysis of label distribution, revealing whether the bounding boxes are evenly distributed or if certain parts of the dataset are denser. This analysis is crucial for verifying that the model can accurately detect objects across varying sizes and positions in the images. The dataset comprised a total of 2936 images, with 70% allocated to the training set (2067 images), 20% to the validation set (575 images), and 10% to the testing set (294 images). The dataset covers a variety of bus shapes, colors, and sizes to enhance the model's adaptability to different real-world environments.

In addition to bus detection, the dataset also supports bus POV classification, which focuses on identifying whether a specific camera viewpoint is optimal for visually impaired users at bus stops. This dataset includes a range of images from various bus stops, with different times of day, weather conditions, and road patterns, which are crucial for detecting buses and recognizing environmental features. Figure 3 showcases the different patterns of road areas with varying illuminations between daytime and nighttime. The dataset addresses one of the key challenges in visual data processing—variations in lighting—by including images captured under different illumination conditions.

For the POV classification task, each image or video is annotated as either "Good POV" or "Bad POV". Good POVs present a clear, balanced view of the bus and its surroundings, aiding bus identification and detection, as demonstrated in Figure 4A. Conversely, Bad POVs show angles that hinder visibility, such as excessive tilt or poor positioning Figure 4B. The dataset includes 2362 Good POV and 2035 Bad POV images for training, with 788 Good POV and 679 Bad POV images set aside for validation. This balanced distribution helps to

ensure that the model can accurately classify both useful and less useful viewpoints for bus detection.

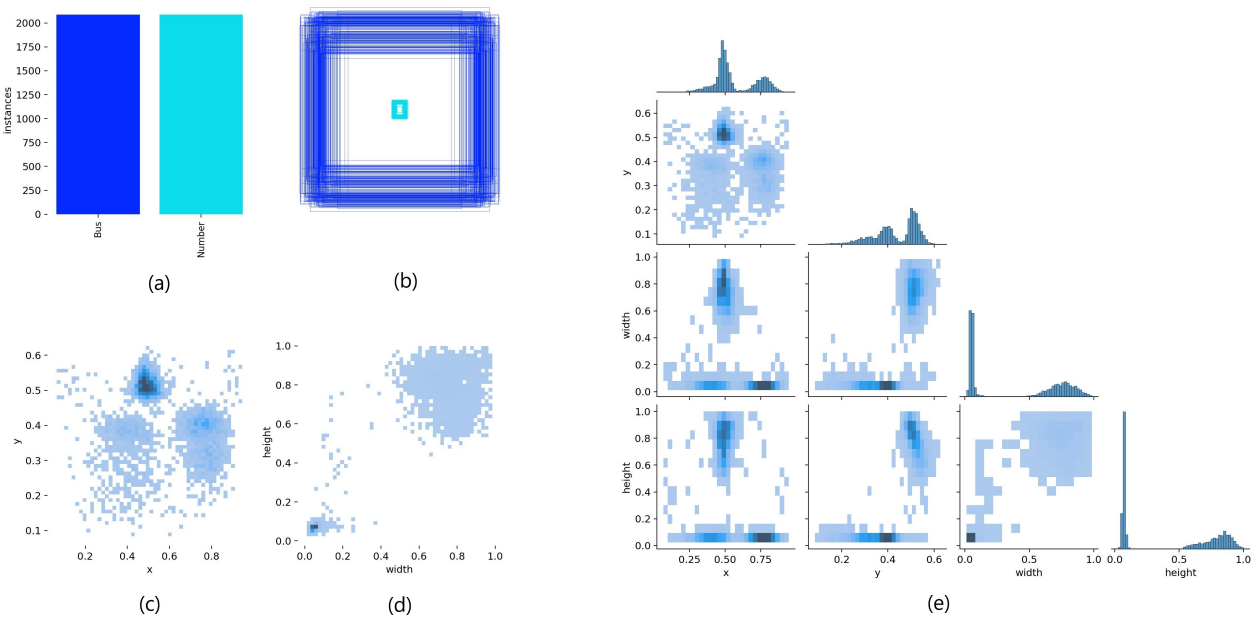


Figure 2. Visualization of the dataset: (**a**) The number of annotations for each class. (**b**) A visual representation of the location and size of the bounding boxes. (**c**) The statistical distribution of bounding box positions. (**d**) The statistical distribution of bounding box sizes. (**e**) Detailed label distribution analysis.



Figure 3. Different Patterns of Road Area with Different Illuminations Between Daytime and Nighttime.

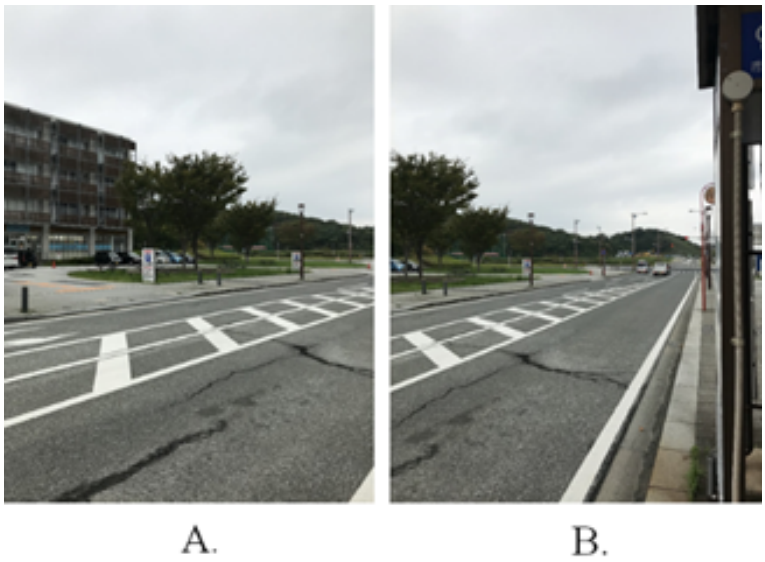


Figure 4. Examples of (**A**) Good and (**B**) Bad Viewpoints.

All images were preprocessed by auto-orienting and resizing them to 640 × 640 pixels to ensure consistent object detection and classification across different scales and positions. This comprehensive dataset design, supported by structured annotation and augmentation, enables the development of a robust system for both bus detection and POV classification, ultimately assisting visually impaired users in identifying buses and navigating bus stops under a wide range of conditions.

2.2. YOLOv10

YOLOv10, released in 2024, represents a significant advancement in the YOLO (You Only Look Once) series, known for its real-time, end-to-end object detection capabilities. As the latest iteration in the YOLO family, YOLOv10 builds upon the success of previous versions (v1–v9) while introducing innovative enhancements that improve both efficiency and accuracy, making it particularly suitable for applications such as enhancing accessibility for visually impaired individuals using public transportation.

The architecture of YOLOv10, shown in Figure 5, incorporates a unified dual-allocation strategy for Non-Maximum Suppression (NMS)-free training, which reduces computational complexity and improves inference speed without compromising detection quality [9]. This is a key improvement over previous YOLO versions, as it eliminates the need for NMS during inference, which can introduce latency. YOLOv10's focus on reducing inference time while maintaining high accuracy makes it highly effective for real-time applications.

The detection pipeline in YOLOv10 begins by processing the input image through an advanced backbone network that extracts feature representations. These features are then aggregated at multiple scales by the neck module, which effectively fuses the information before passing it to the head module. The head module generates multiple bounding boxes and class predictions for each object. Unlike earlier versions of YOLO, YOLOv10 performs object detection without NMS during inference, which helps to reduce processing time and improves real-time performance.

A significant architectural advancement in YOLOv10 is its use of an enhanced variant of the Cross-Stage Partial Network (CSPNet) [26], which optimizes gradient propagation and reduces computational redundancy. This results in superior feature extraction, which helps the model maintain computational efficiency while improving performance. The neck module in YOLOv10 also integrates a Path Aggregation Network (PANet) layer [27], which

facilitates effective multi-scale feature fusion, further improving detection accuracy across objects of various sizes.

The head module of YOLOv10 consists of two configurations: a pair of multi-heads during training and a pair of single-heads during inference. The multi-head configuration generates multiple predictions for each object during training, providing comprehensive supervisory signals and improving learning accuracy. In contrast, the single-head configuration produces a single optimal prediction per object during inference, removing the need for NMS and reducing latency, thus enhancing the overall efficiency of the model.

For classification tasks, YOLOv10 can be adapted to use only the backbone network, which is directly connected to a classification head. This streamlined approach utilizes the backbone's advanced feature extraction capabilities, omitting the neck and detection head, which reduces computational overhead and maintains the real-time performance advantages of YOLOv10. This flexibility in the architecture makes YOLOv10 not only highly efficient for object detection but also versatile for other tasks like classification.

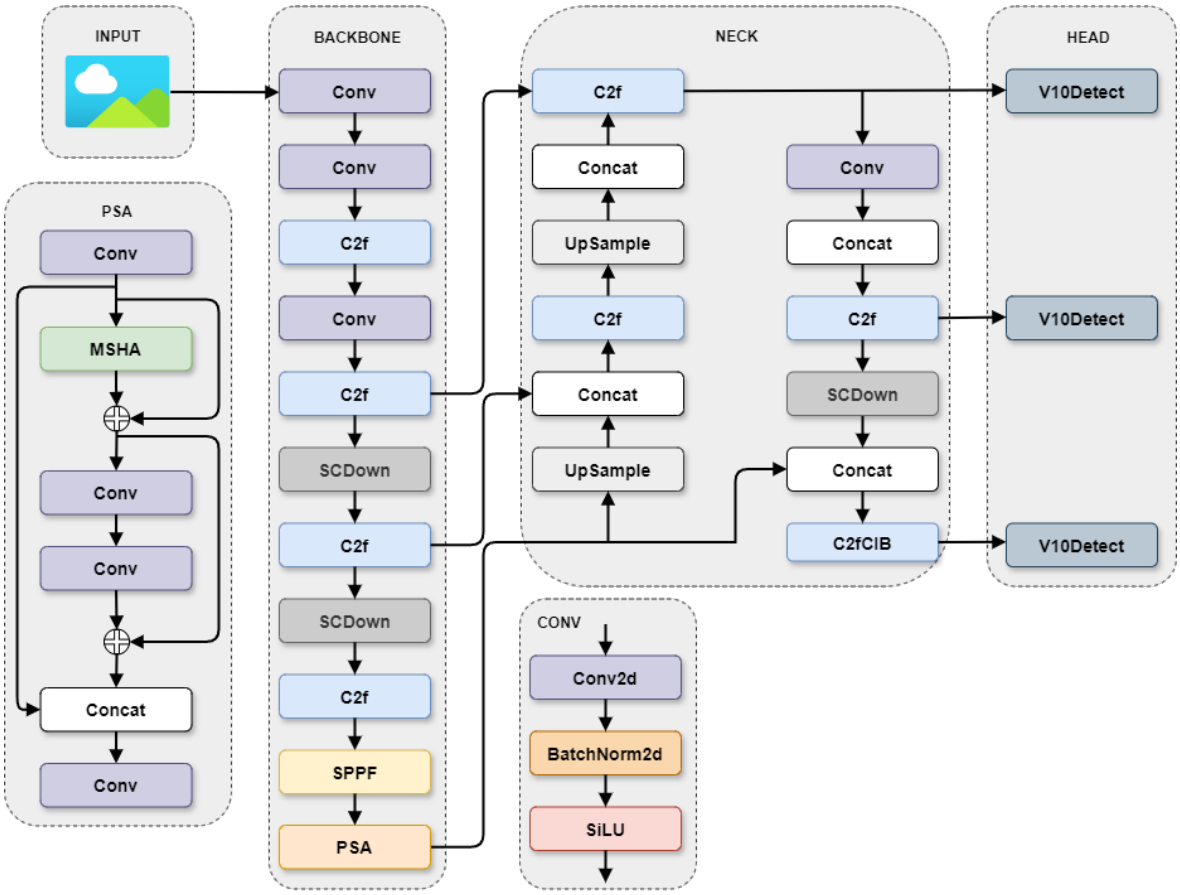


Figure 5. The YOLOv10 model structure.

2.3. Integrating the Coordinate Attention Mechanism

To enhance computational efficiency, integrating advanced features into models like YOLOv10 is crucial. The Coordinated Attention (CA) module, for instance, enhances feature expression by capturing extensive dependencies among image features [17]. This mechanism is acknowledged as a significant advancement in enhancing feature expression capabilities, particularly within mobile networks [28]. As depicted in Figure 6, the CA mechanism is specifically engineered to capture inter-channel information and spatial relationships within the feature map, thereby improving the model's ability to accurately locate and identify object regions.

While attention mechanisms offer valuable benefits, their application in mobile networks can be constrained by computational overhead, especially in models with limited resources. For instance, traditional self-attention mechanisms often introduce impractical computational demands for mobile networks. Consequently, alternative methods such as Squeeze-and-Excitation (SE) [15], BAM, and CBAM are frequently employed. However, SE is limited in its ability to address spatial information, focusing solely on internal channel data. This limitation is critical in computer vision, where the spatial structure of objects is essential.

In contrast, the CA mechanism offers several advantages for detecting small objects and handling complex backgrounds. It effectively integrates both channel and directional positional information, addressing long-range dependency issues with relatively fewer parameters and computational demands. Unlike methods that rely on global pooling for channel position information, which only captures local details and fails to account for long-range dependencies, the CA mechanism provides a more comprehensive approach. It enhances spatial awareness by combining spatial and channel relationships, making it a flexible and lightweight solution suitable for integration into core modules of lightweight networks.

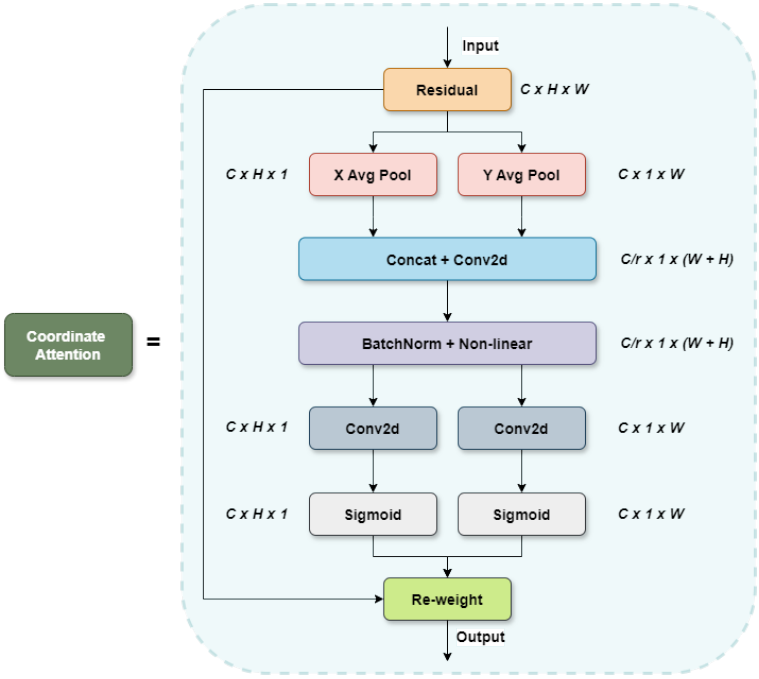


Figure 6. The structure of Coordinate Attention.

Figure 6 illustrates a comparison between two attention mechanisms: (a) SE, and (b) CA. CA extends SE by attending not only to inter-channel correlations but also to spatial locations, thereby enhancing the model's spatial awareness.

As illustrated in Figure 6, the Coordinate Attention (CA) mechanism applies average pooling along both the X and Y axes of the input feature map, which is represented as "Input" with dimensions $C \times H \times W$. This process generates two one-dimensional vectors, resulting in feature maps of dimensions $C \times H \times 1$ and $C \times 1 \times W$, respectively. Unlike conventional global average pooling, which would fully compress spatial information into the channel dimension, this method avoids such compression. Instead, the pooling operation is decomposed to preserve spatial structure, facilitating the capture of precise positional information and enabling the modeling of distant spatial dependencies. Mathematically, the global average pooling for the input feature map is formulated in Equation (1) as:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j) \tag{1}$$

which averages the feature values over both spatial dimensions. Equations (2) and (3) represent the 1D pooling along the horizontal and vertical axes, respectively:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h,i) \tag{2}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j,w) \tag{3}$$

The feature vectors resulting from these two global average pooling operations are concatenated and subsequently subjected to dimensionality reduction through 1×1 convolutional kernels, followed by batch normalization. This process is formalized in Equation (4):

$$f = \delta\left(F_1\left(\left[z^h, z^w\right]\right)\right) \tag{4}$$

where δ represents the non-linear activation function and F_1 denotes the shared 1×1 convolutional transformation. After this operation, the intermediate variable f is obtained, as defined in Equation (5):

$$f \in R^{\frac{C}{r} \times (H+W)} \tag{5}$$

where r is the reduction ratio controlling the block size. The intermediate feature f is then split into two tensors along the spatial dimension, as indicated by Equations (6) and (7):

$$f^h \in R^{\frac{c}{r} \times H} \tag{6}$$

$$f^w \in R^{\frac{c}{r} \times W} \tag{7}$$

These tensors, f^h and f^w, are each transformed using 1×1 convolutional layers, f_h and f_w, respectively, to match the channel dimension of the input X. The transformations are represented in Equations (8) and (9):

$$g^h = \sigma\left(F_h\left(f^h\right)\right) \tag{8}$$

$$g^w = \sigma(F_w(f^w)) \tag{9}$$

where σ denotes the sigmoid activation function. Finally, the outputs g^h and g^w are unfolded and utilized as attention weights. The final output of the Coordinate Attention Block, denoted by Y, is expressed in Equation (10):

$$y_c(\mathrm{i},j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{10}$$

From these derivations, it is clear that CA introduces positional information along with channel attention, allowing for more effective modeling of spatial relationships in the input feature map.

2.4. AKConv

Further improving YOLOv10's capability, AKConv (Adaptive Kernel Convolution) offers a flexible convolutional approach that addresses limitations inherent in standard convolution operations. Current neural networks using standard convolution operations

have driven significant progress in deep learning. However, they face key limitations: fixed sampling shapes confine operations to local windows, limiting broader contextual understanding, and fixed kernel sizes $(k \times k)$ lead to exponential growth in computational parameters, posing challenges in building lightweight models.

In contrast, the kernel convolution (AKConv) [19] offers a flexible approach, allowing variable kernel shapes and parameters to better capture information across wider regions. AKConv dynamically adjusts its sampling shapes through learned offsets, which enables it to adapt to varying image characteristics, improving both model performance and parameter efficiency. This adaptability, illustrated in Figure 7, enhances feature extraction by modifying kernel shapes to align with the input image's characteristics.

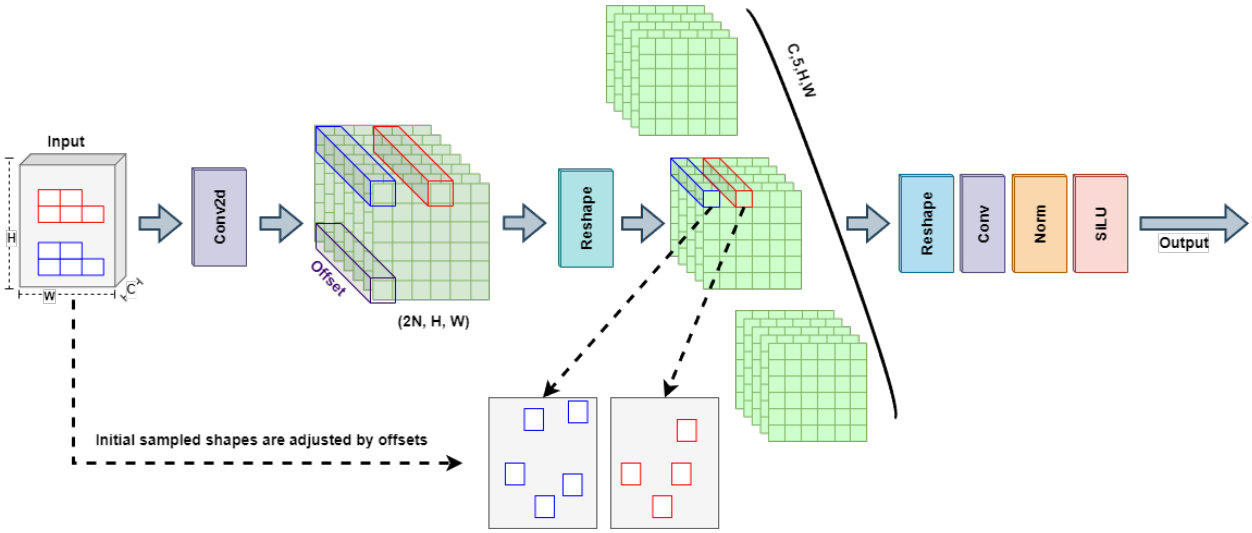


Figure 7. AKConv network structure.

AKConv is particularly effective in medical image analysis, where lesion sizes and distributions vary. The dynamic adjustment of kernel size and shape enables efficient feature extraction across diverse lesions, as shown in Figure 7. By initializing different sampling shapes for a 5×5 grid, AKConv improves coverage and processing accuracy, as depicted in Figure 8. Additionally, it adjusts kernel positions via offsets, enabling better handling of non-rigid deformations, occlusions, and complex backgrounds, enhancing detection accuracy, especially for disease identification, as demonstrated in Figure 9.

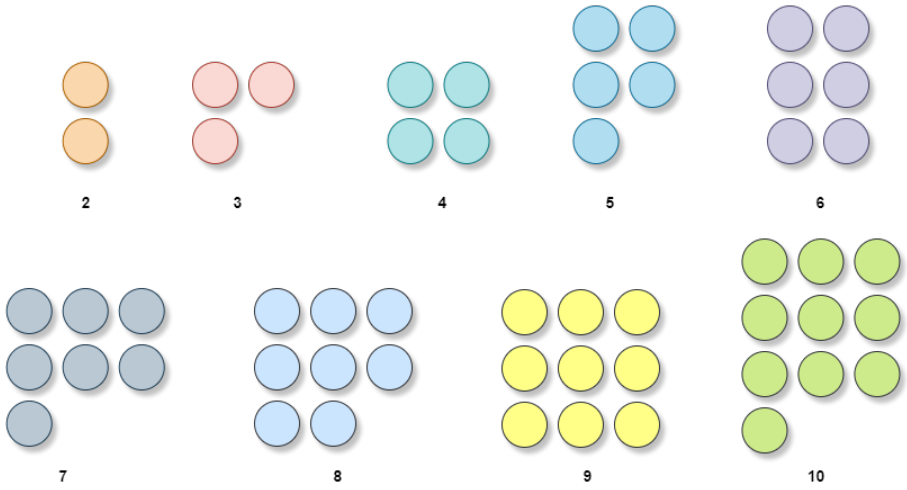


Figure 8. Initial sampling shape.

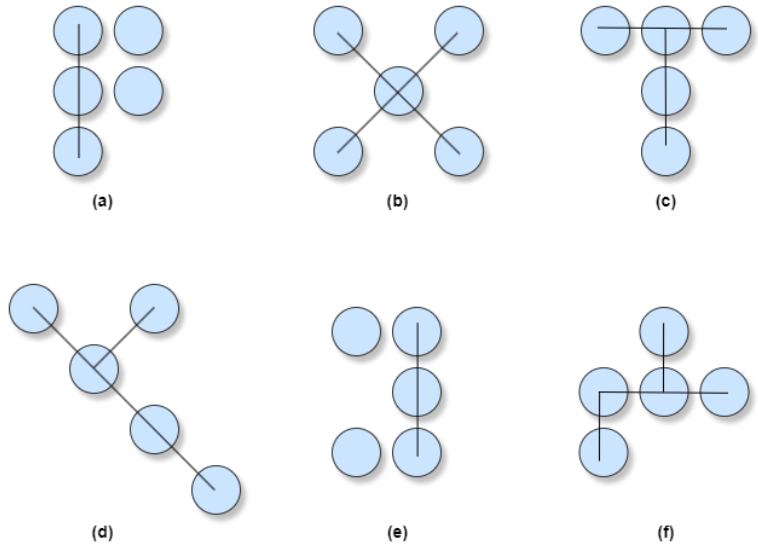


Figure 9. The 5 × 5 different initial sample shapes.

The initial sampling coordinate algorithm, which underpins AKConv's adaptive behavior, localizes features by leveraging convolutional operations with a regular sampling network. This process is expressed in Equation (11):

$$L_x = L_0 + \Delta L_x \tag{11}$$

where L_x represents the adjusted coordinate for the irregular convolution, L_0 is the initial sampling coordinate, and ΔL_x reflects the offset adjustment. Based on these adjusted coordinates, input feature maps are resampled, and the resampled maps are processed through the convolution layer, yielding the final output. The functional expression for AKConv's variable kernel convolution can be formalized in Equation (12):

$$\text{AKConv}(L_0) = \sum \omega(L_0 + L_x) \tag{12}$$

where ω denotes the convolution parameters. This dynamic approach enhances AKConv's ability to adapt to varying image features, improving its capacity for precise feature localization and extraction.

2.5. Improvement of Network Model

The YOLOv10 model has been enhanced through two modifications: the integration of Coordinate Attention (CA) and the substitution of standard convolution layers with Adaptive Kernel Convolution (AKConv). These modifications result in a more robust and adaptable model, improving feature representation and handling varying object scales.

Coordinate Attention (CA): CA replaces Multi-Head Self-Attention (MHSA) within the Partial Self-Attention (PSA) modules of the YOLOv10 backbone. While MHSA manages dependencies effectively, it is computationally intensive and less efficient at capturing long-range dependencies. CA improves this by capturing both channel interdependencies and precise positional information, embedding spatial context into the attention mechanism.

CA achieves this through global average pooling along the height and width of the input feature map $X \in \mathbb{R}^{C \times H \times W}$, yielding context descriptors:

$$z_c^h(h) = \frac{1}{W} \sum_{i=1}^{W} x_c(h, i), \quad z_c^w(w) = \frac{1}{H} \sum_{j=1}^{H} x_c(j, w).$$

These descriptors capture long-range dependencies along each spatial dimension. They are transformed to generate attention weights $g_c^h(i)$ and $g_c^w(j)$, which recalibrate the input feature map:

$$y_c(i,j) = x_c(i,j) \cdot g_c^h(i) \cdot g_c^w(j).$$

By integrating CA into the backbone, the network refines feature maps early on, ensuring improved spatial awareness and object localization.

Adaptive Kernel Convolution (AKConv): The original fixed-size convolutional kernels in YOLOv10 limit adaptability to differing object scales. AKConv addresses this by introducing learnable offsets Δp_k, adjusting kernel sizes dynamically based on input features, thus enhancing nuanced feature extraction.

The modified convolution operation becomes:

$$y(p_0) = \sum_{k=1}^{K} w_k \cdot x(p_0 + p_k + \Delta p_k).$$

AKConv enables the network to focus precisely on salient features across various scales, improving detection and classification of both small and large objects.

The integration of Coordinate Attention and AKConv into YOLOv10 results in a more robust and efficient model. Figure 10 illustrates the updated architecture. The improved YOLOv10 model now benefits from enhanced feature extraction and localization capabilities, making it particularly suitable for applications requiring high accuracy and real-time performance.

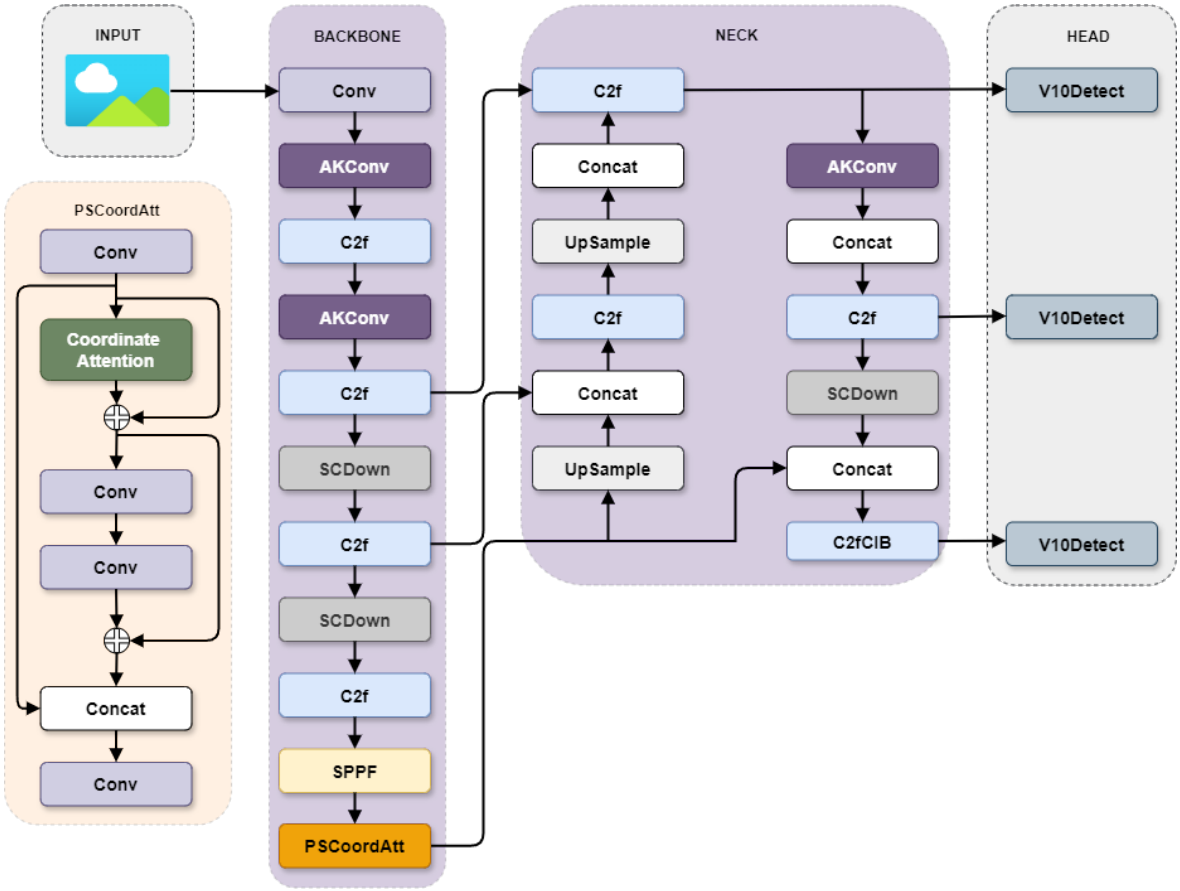


Figure 10. The Improved YOLOv10 model structure.

The integration of Coordinate Attention (CA) and Adaptive Kernel Convolution (AKConv) into the YOLOv10 model significantly enhances its detection precision and efficiency. CA modules are strategically placed within the Cross Stage Partial (CSP) blocks, immediately following the convolutional layers in the backbone. This arrangement allows for the early and efficient enhancement of feature maps, directing the network's focus to critical spatial regions. By improving the model's ability to capture and process essential spatial information, CA contributes to increased accuracy and efficacy in object classification, particularly in complex scenarios.

Meanwhile, AKConv is adopted in place of the standard convolutional layers in the detection head. This modification involves the incorporation of an offset learning layer, which dynamically predicts adjustments to the kernel parameters. As a result, AKConv enables the kernels to align with relevant spatial information, allowing for nuanced and adaptive feature extraction. This dynamic adjustment capability is especially beneficial in handling objects of varying sizes and shapes, expanding the model's effectiveness in dynamic environments. Together, these enhancements make the YOLOv10 model more robust, versatile, and capable of maintaining high performance across diverse and challenging object detection tasks.

3. Experiment and Results

3.1. Experimental Environment

The experimental setup for this study utilized a GPU with CUDA support and the PyTorch framework to facilitate efficient learning and object detection. Parallel computing, enabled by the GPU and CUDA, accelerated the training and inference processes of the YOLOv10 model. The PyTorch framework was employed for model construction and training. The detailed configuration of the experimental environment is provided in Table 1.

Table 1. Experimental setting.

Device	Configuration
System	Windows 11 Pro
CPU	12th Gen Intel(R) Core (TM) i7-12700 2.10 GHz (Manufacturer: Intel Corporation)
GPU	NVIDIA GeForce RTX 3060 Ti (Manufacturer: NVIDIA Corporation)
Framework	PyTorch 1.13.1
IDE	Visual Studio Code
Python version	version 3.7.9

3.2. Evaluation Metrics

Model evaluation is essential to assess performance and alignment with research objectives. In bus object detection, key metrics include accuracy, speed, and efficiency. Accuracy-focused metrics are critical as the model must identify buses with high precision. Common accuracy metrics include precision (P) and recall (R). Precision measures the ratio of true positives (TP) to total positive predictions (TP + false positives (FP)), indicating how accurately the model predicts the positive class. Recall is the ratio of true positives to total actual positives (TP + false negatives (FN)), reflecting the model's ability to identify all relevant positive cases.

Additionally, metrics like average precision (AP) and mean average precision (mAP) are widely used in object detection. AP evaluates how well the model distinguishes relevant from irrelevant objects, calculated from the area under the precision-recall curve. mAP provides a comprehensive performance overview by averaging the APs across all object classes. The equations are shown in Equations (13)–(16).

$$P = \frac{TP}{TP + FP'} \tag{13}$$

$$R = \frac{TP}{TP + FN'} \tag{14}$$

$$AP = \int_0^1 P(r)dr \tag{15}$$

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{k} \tag{16}$$

Speed and efficiency metrics, such as gigaflops per second (GFLOPS) and inference time, are crucial in real-time applications like video-based object detection. GFLOPS measures the number of floating-point operations per second, while inference time quantifies how long the model takes to process a single input.

Other key metrics include the number of model parameters and model size. The number of parameters represents the model's complexity, as it indicates the amount of information the model uses to learn patterns from the data. Model size, measured in bytes or megabytes (MB), reflects both the model's complexity and its suitability for space-limited applications requiring efficient data transfers.

3.3. Model Training and Performance Analysis

3.3.1. Evaluation of Detection Model

In this section, we analyze the performance of YOLOv10 and its improved version, YOLOv10-PSCA-AKConv. YOLOv10-PSCA-AKConv incorporates enhancements such as replacing Multi-Head Self-Attention (MHSA) with Coordinate Attention (CA) and integrating Adaptive Kernel Convolution (AKConv). These modifications aim to refine long-range dependency modeling, improve spatial awareness, and enhance feature extraction flexibility, resulting in a more robust and efficient model that excels in real-time applications, particularly for aiding visually impaired individuals in navigating public transportation. Both models were trained over 1000 epochs to ensure sufficient learning and avoid overfitting. Early stopping was used as a regularization technique, which halts training if there is no improvement in performance on a validation set after a predefined number of epochs. In this case, the patience was set to 100 epochs, meaning that training would stop if validation performance did not improve for 100 consecutive epochs.

The performance of both models was evaluated based on several metrics, including mean Average Precision at 50% overlap (mAP50), Precision, and Recall. Figure 11 provides a summary of these metrics:

YOLOv10-PSCA-AKConv shows a small but important enhancement over YOLOv10, as evidenced by its mAP50 score of 0.9303 compared to the baseline model's 0.9297. Although this difference might seem minor, it reflects a notable improvement in the model's detection capabilities. This advancement is largely due to the refined architectural modifications integrated into YOLOv10-PSCA-AKConv. The increase in Precision from 0.9141 in YOLOv10 to 0.9264 in YOLOv10-PSCA-AKConv indicates a marked improvement in the model's ability to accurately identify true positives while reducing false positives. This improvement can be attributed to the Coordinate Attention (CA) mechanism, which enhances the model's focus on both spatial and channel-specific features. By improving the model's sensitivity to relevant objects, CA effectively increases Precision. Similarly, the increase in Recall from 0.8934 to 0.9036 demonstrates that YOLOv10-PSCA-AKConv is more effective at detecting true positives across various scenarios. This enhancement in Recall is likely due to the Adaptive Kernel Convolution (AKConv), which dynamically

adjusts convolutional kernel shapes to capture features at multiple scales. This adaptability helps the model identify objects that might otherwise be missed, thereby improving its overall detection performance. The combined improvements in Precision and Recall highlight YOLOv10-PSCA-AKConv's ability to offer both accurate and comprehensive object detection. This balance is crucial for applications requiring high accuracy and reliability, such as assistive technologies for visually impaired individuals. The model's enhanced Precision ensures fewer false positives, while its improved Recall ensures that more relevant objects are detected.

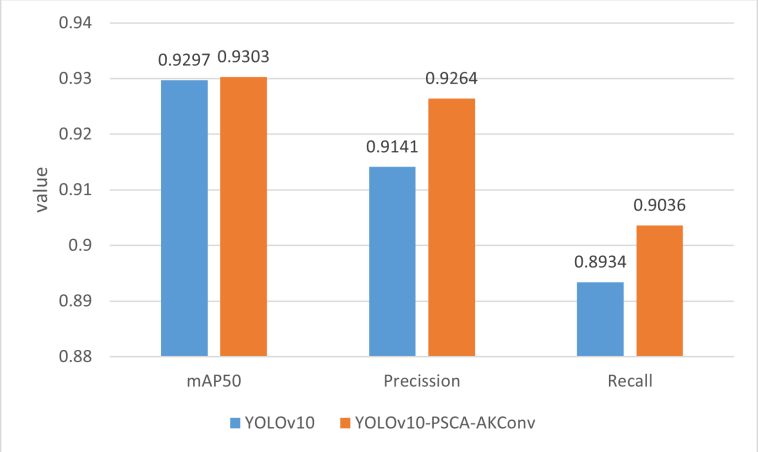


Figure 11. Comparison of YOLOv10 and YOLOv10-PSCA-AKConv Performance Metrics.

Figures 12 and 13 offer detailed insights into the training and evaluation process for both YOLOv10 and YOLOv10-PSCA-AKConv. These figures display training curves for box loss, classification loss (Cls Loss), detection-focused loss (Dfl Loss), Precision, Recall, mAP50, mAP95, and validation losses. The training curves reveal that YOLOv10-PSCA-AKConv generally achieves lower loss values and higher Precision and Recall over the course of training. The model shows more stable performance and improved metrics, reflecting the effectiveness of the enhancements in the architecture.

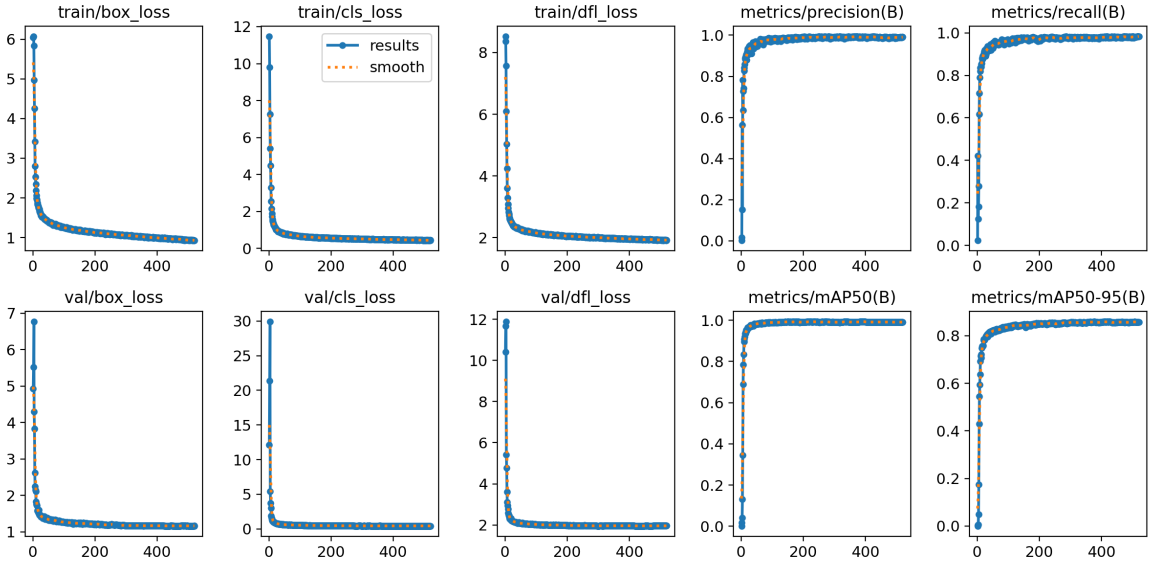


Figure 12. Training and Evaluation Results of YOLOv10 (Baseline Detection Model).

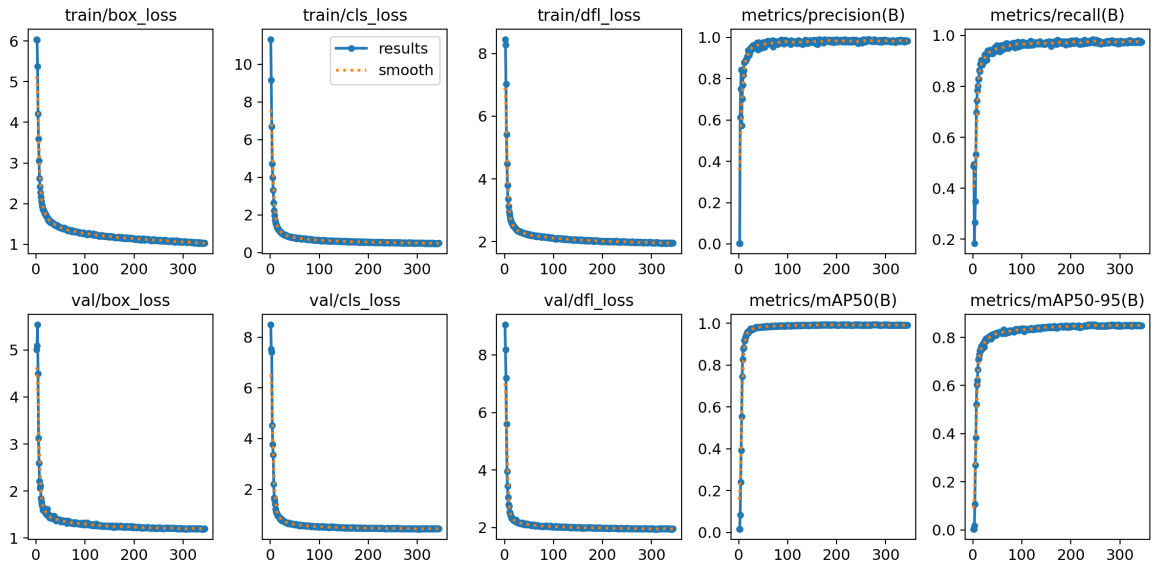


Figure 13. Training and Evaluation Results of YOLOv10-PSCA-AKConv (Improved Detection Model).

Figure 14 highlights a trade-off between different loss metrics in YOLOv10 and YOLOv10-PSCA-AKConv. While YOLOv10-PSCA-AKConv shows a small increase in box loss (1.5823 vs. 1.5764) and Dfl Loss (3.3768 vs. 2.4675), the reduction in classification loss (0.9834 vs. 1.3594) is significant. This suggests that the improved model is slightly less precise in terms of bounding box localization and feature distribution, but it is more accurate in correctly classifying objects.

The reduced classification loss reflects the model's enhanced ability to distinguish between different object classes, which is especially important in real-time detection tasks like identifying buses for visually impaired users. The small increase in box and Dfl loss can be considered an acceptable trade-off, given that the overall goal of the model is to ensure accurate identification rather than perfect bounding box precision. In practical applications, this improvement in classification accuracy could lead to better performance when assisting users in recognizing the correct bus, making the model more suitable for real-world use.

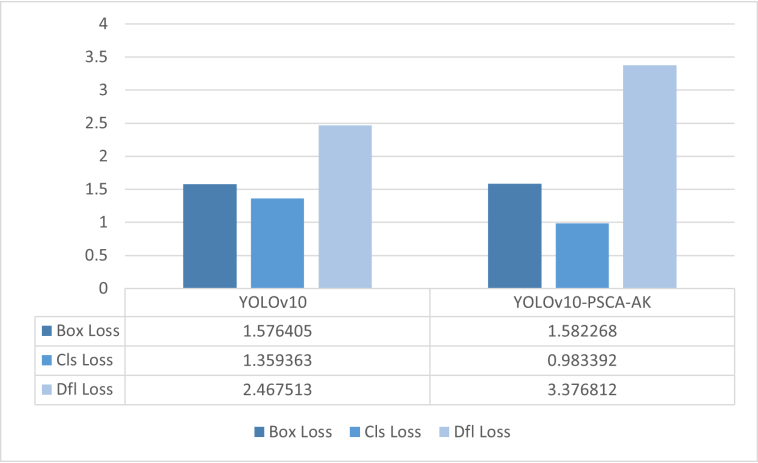


Figure 14. Loss Comparison Between YOLOv10 and YOLOv10-PSCA-AKConv.

3.3.2. Evaluation of Classification Model

In this analysis, we evaluate the classification performance of YOLOv10 and YOLOv10-PSCA-AKConv based on various metrics. The training and evaluation results, depicted in

Figure 15 displays training and evaluation results for both YOLOv10 and YOLOv10-PSCA-AKConv. Panel (a) presents the metrics for YOLOv10, including training and validation losses as well as top-1 accuracy. Panel (b) shows the corresponding metrics for YOLOv10-PSCA-AKConv, highlighting training and validation losses along with top-1 accuracy for the improved model.

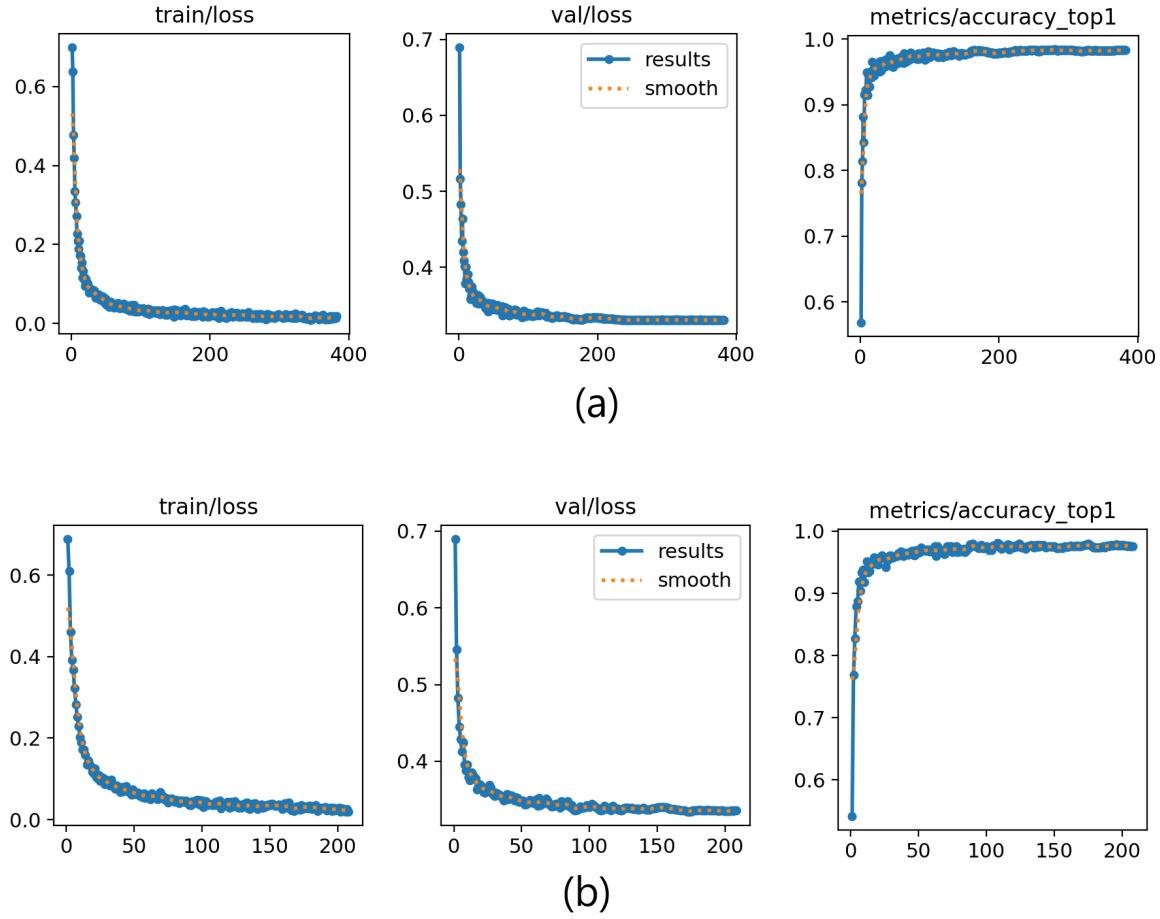


Figure 15. Training and evaluation results of (**a**) YOLOv10 (Baseline Detection Model) and (**b**) YOLOv10-PSCA-AKConv (Improved Detection Model).

Table 2 summarizes the classification metrics for YOLOv10 and YOLOv10-PSCA-AKConv. YOLOv10 slightly outperforms YOLOv10-PSCA-AKConv in accuracy, precision, and F1 score, while both models have similar recall values. YOLOv10-PSCA-AKConv exhibits a small drop in specificity compared to YOLOv10. The comparison between these panels reveals that YOLOv10-PSCA-AKConv generally maintains competitive performance in terms of accuracy but exhibits a slight increase in both training and validation losses compared to YOLOv10. This increase in loss, combined with the observed decreases in precision, specificity, and overall F1 score, suggests that while the improved model excels in detection tasks due to its advanced features, it may experience some trade-offs in classification efficiency. The enhancements in YOLOv10-PSCA-AKConv, such as Coordinate Attention and Adaptive Kernel Convolution, contribute to better detection but may introduce complexities that impact classification performance.

Table 2. Performance Metrics Comparison for YOLOv10 and YOLOv10-PSCA-AKConv. (a) YOLOv10 and (b) YOLOv10-PSCA-AKConv.

Metric	Confusion Matrix (a)	Confusion Matrix (b)
Accuracy	98.36%	98.09%
Precision	98.48%	97.97%
Recall	98.48%	98.47%
F1 Score	98.48%	98.22%
Specificity	98.23%	97.66%

The confusion matrices shown in Figure 16 provide a visual representation of the true positives, false positives, true negatives, and false negatives for both models. Interestingly, despite the advanced features in YOLOv10-PSCA-AKConv showing substantial improvements in detection tasks, such as better handling of long-range dependencies and spatial features, these enhancements do not translate equally well into classification performance. The inclusion of Coordinate Attention (CA) and Adaptive Kernel Convolution (AKConv) significantly boosts the model's object detection capabilities by improving feature extraction and spatial awareness. However, these features might inadvertently complicate the classification process, especially in scenarios with high class variability or where subtle distinctions between classes are crucial. Coordinate Attention, while enhancing spatial feature representation, may introduce additional complexity in classifying objects accurately, particularly when the model needs to focus on nuanced differences between object classes. Adaptive Kernel Convolution, designed to capture varying scales and details, may also affect the model's ability to maintain consistency in classifying objects across different contexts, resulting in a slight decrease in classification metrics such as precision and specificity.

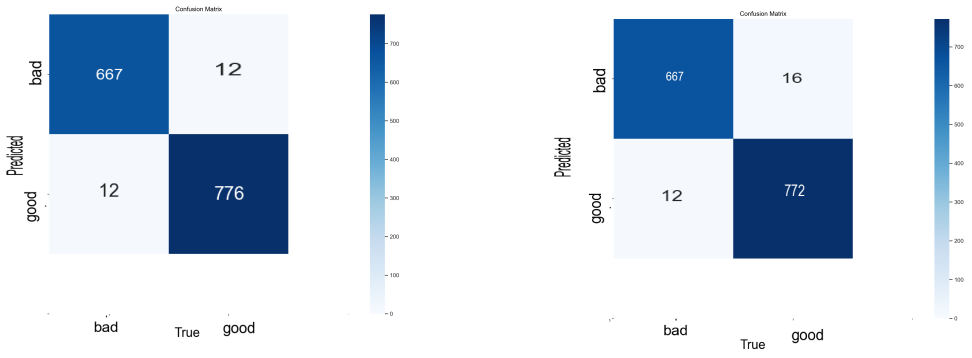


Figure 16. Confusion Matrix Results for YOLOv10 and YOLOv10-PSCA-AKConv. (**a**) YOLOv10 and (**b**) YOLOv10-PSCA-AKConv.

3.4. Models Complexity

In this section, we offer a comprehensive comparison of the computational complexity and storage requirements for the YOLOv10 model and its enhanced variants in both detection and classification tasks.

In the detection domain, as shown in Table 3, YOLOv10 serves as the baseline with a computational complexity of 8.395 GFLOPs and a weight size of 5.54 MB. This benchmark allows us to evaluate the impact of various enhancements on model efficiency. YOLOv10-PSCA, which incorporates Coordinate Attention (CA), shows a slight reduction in computational complexity to 8.357 GFLOPs and a decrease in weight size to 5.43 MB. The CA mechanism improves the model's ability to capture long-range dependencies and spatial relationships in the feature maps. This enhancement is achieved with a minimal increase

in complexity, indicating that CA provides a more refined feature representation without significantly increasing computational overhead. The YOLOv10-PSCA-AKConv model, which integrates both CA and Adaptive Kernel Convolution (AKConv), further optimizes the model. It achieves a computational complexity of 8.041 GFLOPs and a reduced weight size of 5.36 MB. The addition of AKConv introduces flexibility in feature extraction by adapting the convolutional kernel sizes to different input features. This results in a decrease in computational load and storage requirements while enhancing the model's efficiency in processing varying scales and details. The combined application of CA and AKConv not only enhances detection performance but also contributes to a more efficient model, optimizing both computational and storage demands.

Table 3. Comparison of the detection models' complexity.

Detection Model	GFLOPs	Parameters	Weight/MB
YOLOv10	8.395	2.70782	5.54 MB
YOLOv10-PSCA	8.357	2.659844	5.43 MB
YOLOv10-PSCA-AK	8.041	2.625974	5.36 MB

In the classification context, as shown in Table 4, YOLOv10's baseline complexity is 3.45 GFLOPs with a weight size of 3.05 MB. This baseline allows us to measure the efficiency improvements provided by enhancements. YOLOv10-PSCA, incorporating CA, shows a slight reduction in computational complexity to 3.411 GFLOPs and a weight size of 2.95 MB. The decrease in GFLOPs and weight indicates that CA contributes to more efficient classification by enhancing feature attention without a significant increase in computational cost. YOLOv10-PSCA-AKConv further reduces the complexity to 3.163 GFLOPs and a weight size of 2.92 MB. The inclusion of AKConv allows for more adaptable and efficient feature extraction, aligning kernel sizes with input characteristics to optimize computational resources. This results in a more efficient model for classification tasks, with lower computational and storage requirements compared to both YOLOv10 and YOLOv10-PSCA.

Table 4. Comparison of the classification models' complexity.

Classification Model	GFLOPs	Parameters	Weight/MB
YOLOv10	3.45	1.53173	3.05 MB
YOLOv10-PSCA	3.411	1.483754	2.95 MB
YOLOv10-PSCA-AK	3.163	1.470998	2.92 MB

Overall, YOLOv10-PSCA-AKConv demonstrates superior efficiency compared to YOLOv10 and YOLOv10-PSCA in both detection and classification tasks. The model's reduced GFLOPs and weight size highlight the effectiveness of CA and AKConv in optimizing computational and storage demands. YOLOv10-PSCA-AKConv achieves a balance between advanced feature representation and resource efficiency, making it a compelling choice for real-time applications in both detection and classification domains.

3.5. Ablation Experiment

The ablation experiments detailed in Table 5 investigate the impact of different enhancements on model performance. These experiments evaluate the contributions of Coordinate Attention (CoordAtt), Partial Self-Attention (PSCoordAtt), and Adaptive Kernel Convolution (AKConv) to YOLOv10's baseline performance.

Table 5. Ablation experiments.

Model	YOLOv10	CoordAtt	PSCoordAtt	AKConv	mAP50	mAP50-90	Precission	Recall	Cls Acc [1]
1	✓	-	-	-	0.929	0.765	0.914	0.893	0.9557
2	✓	✓	-	-	0.936	0.770	0.924	0.896	0.9470
3 [2]	✓	✓	✓	-	0.938	0.771	0.928	0.908	0.9531
4 [3]	✓	✓	✓	✓	0.930	0.930	0.762	0.903	0.9547

[1] Results are derived from the classification model. [2] YOLOv10 with Partial Self-Attention (PSCA) model. [3] YOLOv10 with Partial Self-Attention (PSCA) and Adaptive Kernel Convolution (AKConv), our improved model.

The baseline YOLOv10 model (Model 1) establishes the reference performance with a mean Average Precision at 50% Intersection over Union (mAP50) of 0.929, an mAP50-90 of 0.765, Precision of 0.914, Recall of 0.893, and a Classification Accuracy of 0.9557. This provides a benchmark for evaluating the effects of subsequent enhancements. Model 2 introduces CoordAtt into the YOLOv10 architecture, resulting in an improved mAP50 of 0.936 and mAP50-90 of 0.770. Precision increases to 0.924, and Recall improves to 0.896. This enhancement demonstrates better object detection performance with CoordAtt, which effectively improves feature attention and spatial awareness. Despite these gains, the Classification Accuracy slightly decreases to 0.9470, indicating a marginal reduction in classification performance. Model 3 incorporates both CoordAtt and PSCoordAtt, which further enhances detection performance. The mAP50 rises to 0.938, and mAP50-90 reaches 0.771. Precision increases to 0.928, and Recall improves to 0.908, with Classification Accuracy at 0.9531. The combination of CoordAtt and PSCoordAtt contributes to significant improvements in detection accuracy and recall, reflecting their effectiveness in enhancing feature extraction and attention mechanisms. Model 4, which integrates CoordAtt, PSCoordAtt, and AKConv, achieves a balanced performance with an mAP50 of 0.930 and mAP50-90 of 0.930. This model shows a Precision of 0.762 and a Recall of 0.903, with Classification Accuracy slightly declining to 0.9547. Although the detection metrics are robust, the inclusion of AKConv appears to negatively impact classification performance. The Classification Accuracy in Model 4 is lower compared to the baseline YOLOv10, suggesting that while AKConv enhances the detection and adaptability of feature extraction, it introduces complexities that adversely affect classification.

The ablation experiments highlight that while each enhancement contributes positively to detection performance, the introduction of AKConv in Model 4 results in a trade-off, particularly affecting classification accuracy. The findings emphasize the need for a balanced approach in incorporating advanced techniques to optimize both detection and classification performance, with Model 4 demonstrating the limitations of combining multiple enhancements in terms of classification accuracy.

4. Conclusions

In this study, we evaluated YOLOv10 and its enhanced version, YOLOv10-PSCA-AKConv, focusing on their performance in object detection and classification tasks. YOLOv10-PSCA-AKConv incorporates advanced architectural improvements, such as Coordinate Attention (CA) and Adaptive Kernel Convolution (AKConv), aimed at better handling long-range dependencies, improving spatial awareness, and enhancing feature extraction flexibility. These modifications were intended to refine real-time object detection, particularly for applications assisting visually impaired individuals in public transportation.

The YOLOv10-PSCA-AKConv model demonstrated a slight yet notable improvement in detection performance compared to YOLOv10. Specifically, YOLOv10-PSCA-AKConv achieved a mean Average Precision at 50% overlap (mAP50) of 0.9303, slightly exceeding

YOLOv10's 0.9297. This improvement reflects better detection capabilities attributed to the model's refined architecture. The enhancement in Precision, from 0.9141 with YOLOv10 to 0.9264 with YOLOv10-PSCA-AKConv, highlights the effectiveness of the CA mechanism in accurately identifying true positives while reducing false positives. Additionally, YOLOv10-PSCA-AKConv's improved Recall, increasing from 0.8934 to 0.9036, signifies better performance in detecting true positives across varied scenarios, a benefit of the AKConv's dynamic feature extraction. Training curves and loss metrics also demonstrate YOLOv10-PSCA-AKConv's superior performance. The model shows lower loss values and improved Precision and Recall throughout training. Despite a minor increase in box loss and detection-focused loss (Dfl Loss), the reduction in classification loss indicates that YOLOv10-PSCA-AKConv achieves more accurate classification, which is crucial for reliable applications like assistive technologies for visually impaired users.

Despite these improvements in detection performance, YOLOv10-PSCA-AKConv shows a slight trade-off in classification tasks. Although the model maintains high accuracy, Precision, and Recall, it experiences a minor decrease in specificity and F1 score compared to YOLOv10. The trade-off is likely due to the added complexity from CA and AKConv, which enhance detection capabilities but may slightly compromise the model's ability to distinguish subtle differences between classes. The integration of CA improves feature extraction for long-range dependencies, while AKConv boosts the flexibility of convolution operations. However, these features may introduce minor inefficiencies in classifying fine-grained categories, affecting the F1 score.

The trade-off in classification performance, particularly in specificity and F1 score, is a direct result of the model's added complexity. The Coordinate Attention (CA) mechanism improves the model's ability to capture long-range dependencies, which benefits object detection but may introduce ambiguity when distinguishing between classes that are visually similar or have subtle differences. The added Adaptive Kernel Convolution (AKConv) layer provides more flexible feature extraction, but this flexibility can increase the model's tendency to misclassify objects that require a more precise, localized focus. As a result, while the model improves in overall detection accuracy and Recall, it may sacrifice the fine-grained classification that is needed for high specificity. This subtle trade-off occurs because the advanced features optimize the model's overall detection capabilities but may blur the distinctions between certain class categories.

In particular, F1 score, which is the harmonic mean of Precision and Recall, can be affected by small reductions in specificity. Although Recall increases due to better detection, the minor increase in false positives—caused by the complexity of CA and AKConv—results in a slight decrease in specificity. This, in turn, lowers the F1 score, reflecting the balance between achieving high Recall (detecting true positives) and minimizing false positives (high Precision).

Thus, while the added features of CA and AKConv are beneficial for improving object detection, the model's ability to classify objects correctly, especially those with subtle differences, is somewhat diminished. The increase in Precision and Recall in object detection tasks is counterbalanced by a small decrease in the model's ability to distinguish fine-grained categories in classification tasks.

YOLOv10-PSCA-AKConv also demonstrates improvements in computational efficiency and storage requirements compared to YOLOv10 and YOLOv10-PSCA. The model achieves reduced GFLOPs and weight size, highlighting the success of the architectural enhancements in optimizing computational and storage demands. YOLOv10-PSCA-AKConv's efficiency in handling varying scales and details, along with its advanced feature representation, makes it a suitable choice for real-time applications requiring both high performance and resource efficiency.

Ablation experiments confirm that CA and AKConv integration benefits YOLOv10's baseline performance. The addition of CA alone improves detection metrics, while the combination of CA with PSCoordAtt further enhances performance. Incorporating AKConv in YOLOv10-PSCA-AKConv results in a balanced improvement across both detection and efficiency, validating the effectiveness of these architectural modifications.

Overall, YOLOv10-PSCA-AKConv shows significant advancements over YOLOv10 in detection accuracy and model efficiency. Although there are minor trade-offs in classification performance, the improved model offers a robust and efficient solution for real-time applications, particularly in assistive technologies for visually impaired individuals.

Author Contributions: Conceptualization, R.A. and C.W.; methodology, R.A. and C.W.; software, R.A.; validation, R.A., T.T. and C.W.; formal analysis, R.A.; investigation, R.A., S.E. and C.W.; resources, R.A., T.T. and S.E.; data curation, R.A.; writing—original draft preparation, R.A.; writing—review and editing, R.A. and C.W.; visualization, R.A.; supervision, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: R.A. is supported by a scholarship from the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT) scholarship.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study cannot be shared due to privacy and confidentiality concerns. Data are stored in a password-protected PC in the Kyushu Institute of Technology.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. World Health Organization. *World Report on Vision*; World Health Organization: Geneva, Switzerland, 2019.
2. Venter, C.; Bogopane, H.; Rickert, T.; Camba, J.; Venkatesh, A.; Mulikita, N.; Maunder, D.; Savill, T.; Stone, J. Improving Accessibility for People with Disabilities in Urban Areas: Améliorer l'accessibilité du milieu urbain aux personnes handicapées. In Proceedings of the International Conference on Urban Transport and the Environment for the 21st Century, Transport Research Laboratory, Pretoria, South Africa, 15–18 July 2002.
3. Hersh, M.; Johnson, M. *Assistive Technology for Visually Impaired and Blind People*; Springer: Berlin/Heidelberg, Germany, 2008. [CrossRef]
4. Anu, V.M.; Sarikha, D.; Keerthy, G.S.; Jabez, J. An RFID based system for bus location tracking and display. In Proceedings of the International Confernce on Innovation Information in Computing Technologies, Chennai, India, 19–20 February 2015; pp. 1–6. [CrossRef]
5. Oudah, A. RFID-based automatic bus ticketing: Features and trends. *IOP Conf. Ser. Mater. Sci. Eng.* **2016**, *114*, 012146. [CrossRef]
6. Vinod, D.; Mohan, A. A Successful Approach to Bus Tracking Using RFID and Low Power Wireless Networks. In Proceedings of the TENCON 2018—2018 IEEE Region 10 Conference, Jeju, Republic of Korea, 28–31 October 2018; pp. 1642–1646. [CrossRef]
7. Own, C.M.; Lee, D.S.; Wang, T.H.; Wang, D.J.; Ting, Y.L. Performance Evaluation of UHF RFID Technologies for Real-Time Bus Recognition in the Taipei Bus Station. *Sensors* **2013**, *13*, 7797–7812. [CrossRef]
8. Tangsuksant, W.; Wada, C. Classification of Viewpoints Related to Bus-Waiting for the Assistance of Blind People. *Int. J. New Technol. Res.* **2018**, *4*, 43–52. [CrossRef]
9. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
10. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946.
11. Huang, G.; Liu, Z.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2016**, arXiv:1608.06993.
12. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861
13. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. *arXiv* **2014**, arXiv:1409.4842.
14. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2016**, arXiv:1610.02357
15. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *arXiv* **2019**, arXiv:1709.01507.

16. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521.
17. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. *arXiv* **2021**, arXiv:2103.02907.
18. Gennari, M.; Fawcett, R.; Prisacariu, V.A. DSConv: Efficient Convolution Operator. *arXiv* **2019**, arXiv:1901.01928.
19. Zhang, X.; Song, Y.; Song, T.; Yang, D.; Ye, Y.; Zhou, J.; Zhang, L. AKConv: Convolutional Kernel with Arbitrary Sampled Shapes and Arbitrary Number of Parameters. *arXiv* **2023**, arXiv:2311.11587.
20. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv* **2016**, arXiv:1605.06409.
21. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. *arXiv* **2017**, arXiv:1703.06870.
22. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. *arXiv* **2017**, arXiv:1712.00726.
23. Bochkovskiy, A.; Wang, C.; Liao, H.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
24. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. *arXiv* **2019**, arXiv:1911.09070.
25. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. *arXiv* **2019**, arXiv:1904.08189.
26. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *arXiv* **2019**, arXiv:1911.11929.
27. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. *arXiv* **2018**, arXiv:1803.01534.
28. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *arXiv* **2019**, arXiv:1910.03151.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.