AITA
ARTIFICIAL INTELLIGENCE THEORY AND APPLICATIONS

RESEARCH ARTICLE

Improving the Performance of the YOLOv8 Algorithm for Air Conditioner Detection in Urban Areas: Data Augmentation Techniques and Model Optimization

Can Aydın [a†]

[a] Department of Management Information Systems, Dokuz Eylül University, İzmir, Türkiye
[†] can.aydin@deu.edu.tr, corresponding author

RECEIVED JUNE 03, 2025
ACCEPTED SEPTEMBER 07, 2025

CITATION Aydın, C. (2025). Improving the performance of the YOLOv8 algorithm for air conditioner detection in urban areas: Data augmentation techniques and model optimization. *Artificial Intelligence Theory and Applications, 5(*2), 49-65.

Abstract

This study aims to evaluate the performance of the YOLOv8 algorithm in the task of climate detection and optimize it for urban environments. Currently, YOLOv8 is a state-of-the-art deep learning model designed with an architecture that gives superior performance results in object detection tasks, combining speed and accuracy. YOLOv8s is adeep learning model that will do well in object detection in complex urban environments. Performance characteristics of a deep learning model will include the influence of background intensity and the effect of different camera illumination while detecting -air conditioner. . The model performs better thanks to several data augmentation techniques, variations in model architecture, and training strategies. Experimental results identified that data augmentation techniques make YOLOv8 perform much better, while model architecture variations present a trade-off between speed and accuracy. In particular, the complicated data augmentation methods, such as Mosaic and perspective switching, significantly improve the model's generalization capability. It would mean that YOLOv8 exceptionally quite well in recognizing air-conditioners and similar objects in an urban context; hence, until now, everything looks very encouraging to support the consideration of YOLOv8 in applications having to do with energy efficiency, heat island effect monitoring, and other environmental analyses.

Keywords: object detection, deep learning, data augmentation, yolo v8, urban areas

1. Introduction

Object detection has emerged as a fundamental research area within computer vision and artificial intelligence, owing to its wide-ranging applications in domains such as urban planning, traffic monitoring, and public safety. However, implementing robust object detection systems in metropolitan environments remains a significant challenge. Urban scenes are characterized by high visual complexity due to the presence of dense arrangements of streets, buildings, vehicles, pedestrians, and other structural elements. This complexity imposes both computational and algorithmic challenges, particularly

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN: 978-605-69730-2-4 © 2025 İzmir Bakırçay University

regarding the need for detection models to operate with high precision and real-time efficiency.

Among various object detection algorithms, the YOLO (You Only Look Once) family continues to be recognized for its superior balance between detection speed and accuracy. Originally introduced by Redmon and Farhadi [1], YOLO reframes object detection as a single regression task, enabling the model to process the entire image in a single forward pass. This holistic approach eliminates the need for region proposal methods, which are commonly used in traditional detection frameworks. As a result, YOLO offers notable advantages, including real-time performance, architectural simplicity, and the capacity to detect multiple objects simultaneously. However, its performance may degrade in highly complex urban scenes where overlapping objects, varying lighting conditions, and dense visual clutter are prevalent. Over time, the YOLO architecture has undergone several refinements—from YOLOv1 through YOLOv8—with each iteration introducing architectural improvements and optimization techniques aimed at enhancing detection accuracy and computational efficiency. The most recent version, YOLOv8, leverages state-of-the-art deep learning advances to achieve significant gains in both precision and speed, making it a strong candidate for urban object detection tasks.

This study enhances the YOLOv8 algorithm to improve object detection performance in urban environments, with a particular focus on the detection of air conditioning units. One of the key motivations is the urban heat island (UHI) effect, a well-documented phenomenon in which urban centers exhibit significantly higher temperatures than their surrounding rural areas. This thermal disparity is primarily attributed to factors such as the high heat retention capacity of construction materials like asphalt and concrete, the scarcity of vegetated areas, and elevated levels of energy consumption.

Air conditioning systems, in particular, play a substantial role in exacerbating the UHI effect—not only due to their intensive energy use but also because of the waste heat expelled into the atmosphere by their external condenser units. Accurately detecting and mapping the spatial distribution of such devices can therefore provide critical insights for climate-responsive urban planning and environmental policy interventions.

This study further conducts a comprehensive performance evaluation of the YOLOv8 algorithm for detecting air conditioning units across a variety of urban environments. Key factors examined in the analysis include object density, lighting variability, background complexity, and diverse mounting configurations observed under different perspectives. The experimental findings indicate that YOLOv8 demonstrates strong performance in air conditioner detection tasks; however, its effectiveness may vary under challenging conditions such as complex illumination or high object density. Moreover, the incorporation of specific data augmentation techniques contributed to greater diversity in the training dataset, thereby enhancing the model's ability to generalize across heterogeneous urban scenarios.

Most state-of-the-art object detection algorithms incorporate data augmentation techniques to improve generalization performance, particularly when confronted with diverse and complex real-world scenarios. However, existing object detection datasets often lack sufficient variability to represent the dynamic and heterogeneous nature of urban environments. This limitation can hinder a model's ability to generalize effectively to varying conditions such as changes in lighting, perspective, and object density.

To address this, the present study integrates a suite of data augmentation strategies specifically designed to simulate such environmental variations. These strategies not only diversify the training set but also contribute to improved model robustness. In addition, this work offers a detailed discussion on the necessity of employing multiple augmentation techniques, especially in the context of complex urban settings where visual diversity is pronounced.

The contributions of this paper are threefold. First, it presents a comparative evaluation of multiple YOLOv8 architectures—including YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), and YOLOv8l (large)—in the context of object detection performance within urban environments. These variants are specifically designed to offer different trade-offs between model complexity, depth, and detection accuracy. This study systematically investigates these trade-offs, with a particular emphasis on balancing inference speed and predictive accuracy. Based on the experimental findings, the work identifies the most suitable YOLOv8 configuration for deployment in dense and visually complex urban settings.

The findings of this study underscore the practical utility of the YOLOv8 framework in a range of urban applications, including energy efficiency auditing, building maintenance, environmental monitoring, and urban planning. In particular, the ability to detect areas with high concentrations of air conditioning units offers actionable insights for designing interventions to mitigate the urban heat island (UHI) effect. This work makes a notable contribution to the existing body of research by demonstrating the applicability of advanced object detection models—specifically YOLOv8—in identifying energy-intensive infrastructure within metropolitan environments. As such, the proposed approach represents a meaningful step toward the development of more sustainable and livable urban ecosystems.

The present study focuses on the detection of air conditioning units in urban environments using the YOLOv8 object detection algorithm. In this context, the research is guided by the following key question:

- How effectively can the YOLOv8 algorithm detect energy-related objects like air conditioners in urban environments, and what impact do factors such as object density, lighting variations, background complexity, and mounting configurations have on its detection performance?

Furthermore, the study investigates the role of data augmentation techniques in improving the generalization capability and overall accuracy of the YOLOv8 model. It systematically examines the contributions of various augmentation strategies to detection robustness under diverse urban conditions.

2. Literature Review

Object detection remains one of the most prominent research areas within computer vision and artificial intelligence due to its wide applicability across domains. Among the various approaches developed, the YOLO (You Only Look Once) algorithm represents a significant advancement, offering superior performance in terms of both detection speed and accuracy. A growing body of literature has evaluated different versions and applications of the YOLO framework, particularly in the context of urban environments, where its performance has been rigorously assessed across diverse detection tasks.

Redmon and Farhadi presented the development and performance of the YOLOv3 algorithm in 2018, and it outperformed other object detection algorithms in both speed and accuracy [1]. This detects objects in three different scales, having Darknet-53 as its backbone, hence finding the small, medium, and big objects with much better precision. This paper will test the performance of YOLOv8 within an urban area; therefore, the overall performance evaluations here can be compared against those of an urban area. Flores-Calero et al. conducted a study in 2019 about real-time traffic sign detection using the YOLO algorithm, where the premise of the study was based on the accuracy and quickness of the detection of the traffic signs [2]. Traffic signs are the most common objects in cities; therefore, from this work, the results can be compared with the results in our climate detection study. Vinh et al 2023 discuss, based on the YOLO algorithm, the performance in the detection of pedestrians concerning Intelligent Transport Systems. Since detecting pedestrians is one of the most important object detection tasks within urban settings, the findings of the present study can be compared to those from the air conditioner detection study [3]. Besides, Cong et al. (2020) compared YOLO with other state-of-the-art object detection algorithms; therefore, the paper of Liu et al. is a very extensive review about deep learning-based object detection methods [4]. This would provide the general assessments and comparisons, which may give the view of how well YOLO is performing in the urban area. Koonce et al. 2017 came up with SqueezeNet; it was a deep-learning model that was very accurate with few parameters and small model sizes compared to the fast and light models like YOLO [5]. It plots the speed-accuracy performance of YOLO in an urban area compared with lightweight models such as SqueezeNet. Geiger et al., 2012, presented the benchmark dataset for autonomous driving-namely, KITTI-and tested YOLO running varied object detection algorithms on that dataset [6]. Since the KITTI dataset comprises an urban area, the performance of YOLO for the same shall be matching with the study. Improvement to YOLOv3 by Ruan and Wang included increased confidence adjustment strategies and dynamic settings of aliasing threshold for better detection in dense object scenes. According to Ruan and Wang (2018), this improvement resulted in better detection with fewer errors in crowded urban settings. Özcan et al. 2023 compared different versions of the YOLO algorithm with regard to object detection in the adverse weather conditions of urban areas. Due to its superior speed and accuracy, YOLOv7 is well-suited for various real-time urban applications, even under extreme conditions [8]. Chen et al. (2021) presented an enhanced version of the YOLO-based urban parking spot detection. The enhancement was done on the original network of YOLOv3 in terms of accuracy for a range of urban light conditions, hence improving efficiency in the parking detection tasks [9]. The lightweight YOLOv3 for military target detection then evolved into YOLO-G proposed by Kong et al. in the year 2022; it provides good performance in complicated urban environmental scenarios in terms of much higher speed and higher accuracy while reducing model size [10]. Cao et al. (2023) developed the MCS-YOLO algorithm, which further improved the result of small object detection and generally performed better in heavy urban traffic conditions [11]. On the other hand, Fang et al. (2019) introduced Tinier-YOLO, a compact version of YOLO tailored for deployment in resource-constrained embedded systems [12]. It reduces the model size significantly with no loss of accuracy, hence very effective for real-time urban sensing tasks. Ling et al. 2024 introduce the YOLOv6 optimized for industrial applications [13]. In practice, the model turns out to be very suitable for an urban environment where there is great variability in object size and position, and it possesses a great trade-off between speed and accuracy. Therefore, Kim et al. 2021 developed a very efficient ECAP-YOLO, which was able to detect small objects in a complex urban aerial imagery [14]. Advanced feature extraction modules were integrated into it for enhancing its performance related to detecting small objects, hence becoming very effective in densely urbanized areas. A comparative study regarding the performance of VOC dataset for YOLOv3, YOLOv4, and YOLOv5, performed by Doshi et al. in 2024, proved that the latter outperformed the previous versions not only in terms of accuracy but also in terms of speed, with predisposition for urban surveillance [15]. Later, in 2023, Haimer et al. assessed some versions of YOLO for bad weather and noticed that the best YOLOv7 provided results related to an urban environment in challenging light and weather conditions[16].

Compared to other object detection algorithms, YOLO employs a single-shot detection approach, making it particularly well-suited for real-time applications. For instance, even though two-step object detection algorithms like Faster R-CNN may eventually yield high accuracy, the speed at which YOLO gives results surpasses such algorithms. Because of its speed, YOLO is especially preferred in those applications where real-time decisions must be taken, such as autonomous driving and security cameras [1]. YOLO can detect multiple objects in an image at the same time and determine the locations of these objects. The overarching object detection capability of YOLO provides an advantage, especially in urban areas with a large number of objects in complex scenes. The single-stage structure of YOLO allows processing an image in one go, which makes it fast and efficient. Thus, YOLO requires lesser computational power and hence could do well on devices operating on low hardware resources-for example, embedded systems or mobile devices. Lighter versions such as the Tinier-YOLO and YOLOv3-tiny, can work effectively even on devices with small hardware capacities and find their applications in object detection in an urban setup [12].

Although YOLO enjoys many good advantages, it has some limitations, too. These limitations make other object detection algorithms more suitable in some cases. The major limitation of YOLO is that it is bad at detecting small objects. Since YOLO divides the image into fixed grids, small objects may not be well-represented in these grids. This might make it hard to find objects that are further away from the camera's view, especially from more complex urban areas. In light of the labeling study conducted within this study, air conditioners on the facades of faraway buildings in the street image were small objects. YOLO might experience degraded performance for either the scenes with complex backgrounds or those with occluded objects. Kim et al. [14] demonstrated that YOLO's performance significantly drops when detecting small objects in low-light conditions. Self-evidently, the comparison of YOLO with other widely used object detection algorithms will reveal its strong and weak points far more clearly. Faster R-CNN performs object detection and classification in two steps. Though this architecture brings more accurate results, it cannot support real-time performance like YOLO. Most of the time, Faster R-CNN is going to outperform YOLO, especially on small object detection. However, YOLO leads all others where speed is required; say in security cameras or autonomous driving systems [18]. SSD is also a single-shot object detection algorithm like YOLO, yielding similarly fast results. However, the SSD outperforms YOLO on small object detection and processes more scales in which objects come into view. The grid-based partitioning method of YOLO is unable to achieve as good performance compared to more flexible configurations of SSD [19].

This literature review contains research works that present various variants and implementations of the YOLO algorithm that have been tested against performance in an urban scenario. Comparing this study to the results in the literature will permit an in-depth look at how YOLOv8 manages to carry out object detection within an urban setting.

3. Methodology

3.1. Data Collection

The dataset curated for this study comprises urban images specifically designed to evaluate the performance of YOLOv8 in detecting air conditioning units. These images include outdoor-mounted air conditioners affixed to building facades along city streets and were collected via the Google Street View API. Google Street View provides high-resolution, geotagged, and regularly updated imagery, making it a valuable resource for training and evaluating object detection models.

The dataset includes images from diverse districts in İzmir, Türkiye—namely Konak, Karabağlar, Güzelbahçe, Karşıyaka, and Menemen—capturing a wide range of building densities and architectural typologies. This diversity enables a robust assessment of model performance under various real-world urban conditions. By encompassing densely built areas as well as varied structural environments, the dataset supports a more comprehensive evaluation of detection accuracy and inference efficiency. This, in turn, enhances the generalization capability of the YOLOv8 algorithm in practical applications.

3.2. Data Preprocessing

In this study, a total of 690 urban street-level images were collected using the Google Street View API, specifically for the task of air conditioner detection. Each image was manually annotated to identify the presence and location of air conditioning units. The labeling process was conducted using Roboflow, a widely adopted annotation tool that facilitates the identification of object classes and bounding box coordinates.
Although the initial annotation was performed by a single expert, quality assurance was ensured through a secondary review of 10% of the labeled data by an independent annotator. This validation step focused on visual consistency and the accuracy of bounding box placement. The dataset includes both near-field and far-field representations of air conditioning units, intentionally capturing a range of object sizes and perspectives to support model robustness. This variability is critical for evaluating detection precision under real-world conditions.

In total, 2,360 air conditioning units were annotated across the 690 images, encompassing a wide variety of models, sizes, and installation configurations to enhance dataset diversity and training effectiveness.

Figure 1. Labeling of air conditioners of different sizes

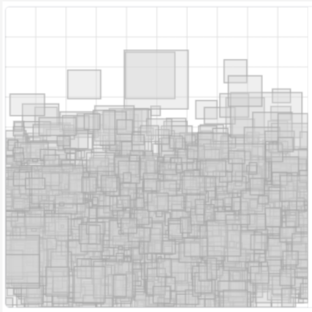


Figure 2. Areas of concentration of labels on the image

As illustrated in Figure 2, the spatial distribution of labeled air conditioning units is predominantly concentrated in the lower and central regions of the images. This pattern reflects the typical real-world installation of outdoor air conditioning units on building facades, often near ground level or mid-height. While object detection algorithms such as YOLO are designed to scan the entire image, the non-uniform spatial distribution of annotated objects may lead to biased learning, where the model disproportionately attends to frequently labeled regions.

In scenarios where the upper portions of images contain few or no labeled instances, the model's detection performance in those regions may be compromised due to limited exposure during training. This imbalance also increases the risk of overfitting, as the model may become overly reliant on localized spatial cues present in the training data.

To address these limitations, a range of data augmentation strategies—including geometric transformations such as rotation, translation, and perspective shift—were employed. These techniques aim to artificially redistribute object positions and diversify spatial contexts, thereby improving the model’s ability to generalize across different regions of the image plane.

Furthermore, analysis of the annotation statistics—summarized in Figure 3—reveals that a single air conditioner appears in 210 images, 2 to 9 units in 440 images, and 10 to 17 units in 31 images. This uneven distribution in object count per image introduces class imbalance, which can affect the training dynamics and detection performance of the model. Specifically, the variation in object density may cause the model to perform suboptimally on underrepresented cases. Preliminary results suggest that detection accuracy is higher for images containing either very sparse or very dense object distributions, while mid-density cases present greater challenges. To mitigate the effects of this imbalance, targeted sampling strategies were adopted to increase the representation of images with fewer air conditioning units. Additionally, data augmentation techniques such as random cropping and image rotation were applied to enhance variability and reduce overfitting within low-frequency classes.

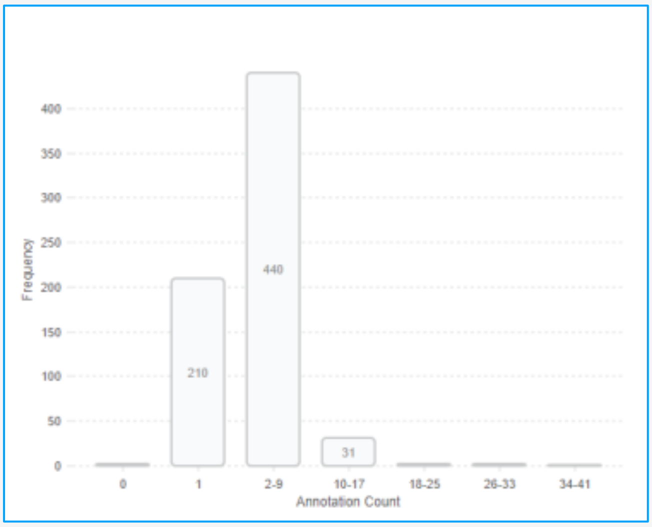


Figure 3. Distribution of the total number of labels in the pictures

Following the annotation process, a series of preprocessing steps were applied to address label distribution imbalances and to standardize the input data. These steps included Auto-Orientation, Resizing the images to a fixed resolution of 640×640 pixels using stretch interpolation, Contrast Stretching for auto-adjustment, and grayscale conversion to reduce color dependence. After preprocessing, data augmentation techniques were employed to further increase variability and simulate real-world visual challenges. These augmentations included horizontal and vertical flipping, application of a 2.5-pixel Gaussian blur, and cutout techniques to introduce partial occlusions. As a result of these steps, the dataset was expanded to include 1,656 images, which were subsequently partitioned into 1,449 for training, 139 for validation, and 68 for testing, adhering to a 70/15/15 split strategy. Since the study focuses on a single-class detection problem (air conditioners), no additional balancing techniques for multi-class scenarios were necessary.

3.3. Modeling

This paper conducts a series of key hyperparameter experiments and training strategies on YOLOv8 to optimize its performance detecting air conditioners in urban environments and enhance its generalization capability. Some experiments over key hyperparameters and some training strategies are performed in this paper to optimize the performance of a deep learning model, YOLOv8, for air conditioner detection within urban areas. The initial hyperparameters the proposed model utilized in the training process are Initial LR = 0.001, which decreases gradually using the Cosine Annealing strategy. Batch size = 16, No. Of epochs = 25, an early stopping strategy is utilized to avoid over-learning of the model. Optimization of the proposed model has been performed using the Adam algorithm with $\beta 1 = 0.9$ and $\beta 2 = 0.999$. Besides, several experiments were applied to avoid lots of problems, such as imbalance in the dataset, variation in the size of air conditioners on the street image, and the complexity of the urban areas, which have negative influences on the model, whereas increasing the generalization capability of the model. The experimental design was selected based on consideration of the requirements of the research question and the fulfillment of existing literature gaps. The experiments involved data augmentation techniques, different variations in model architecture, defining the number of epochs, and setting up the experiment for early stopping. Such experiments were chosen in view of the relevance of the optimization of hyper parameters and the training strategy to the performance of the model, according to the deep learning literature [1][2][3]. It also seeks to present the best performance of the YOLOv8 model for air conditioner object detection tasks in the urban environment and fills some gaps existing today in related literature.

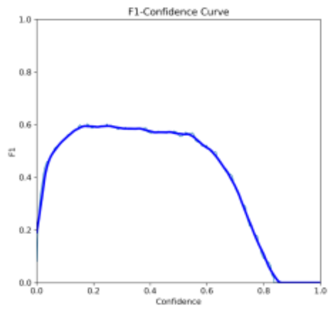
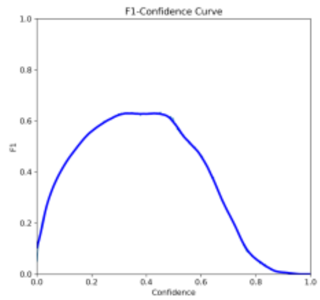
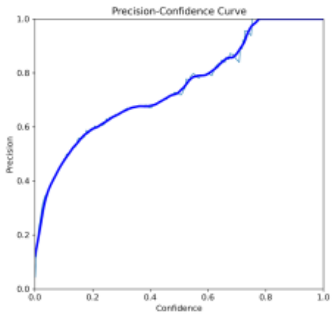
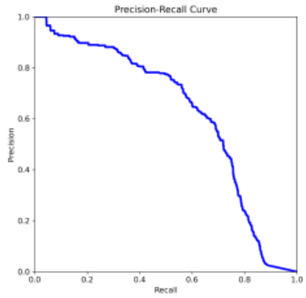
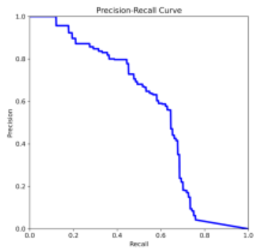
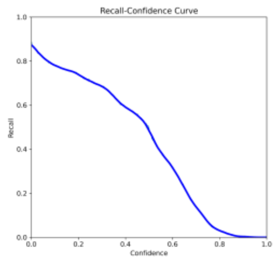
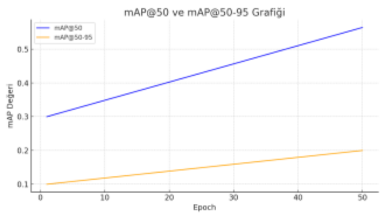
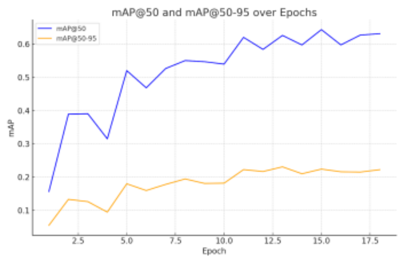
4. Experiments

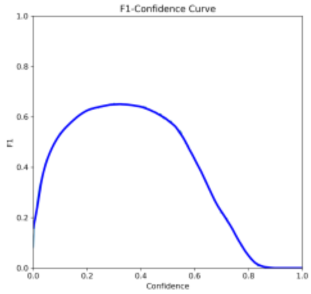
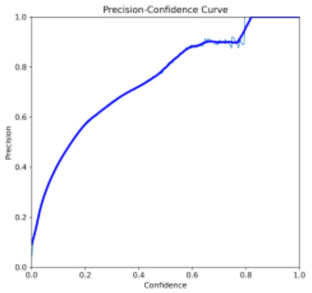
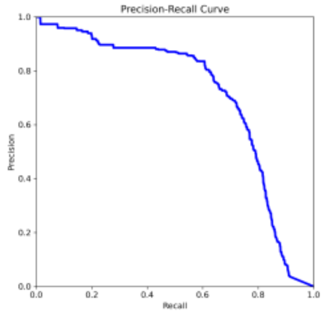
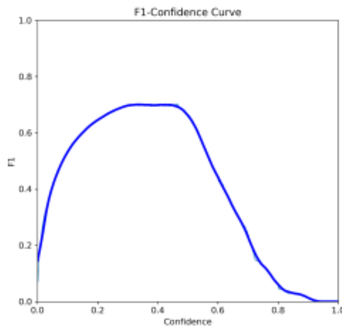
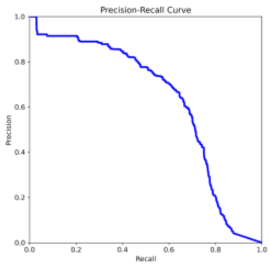
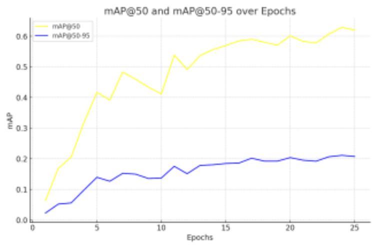
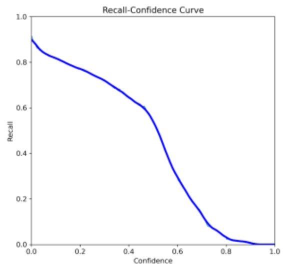
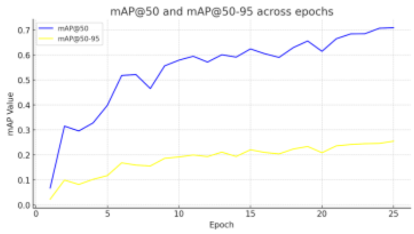
4.1. Experiment with the application of data augmentation techniques

Data augmentation techniques improve the model's generalization ability by generating more variety within the training dataset [2]. Air conditioner units may appear in various mounting configurations and perspectives under diverse environmental conditions. Factors like variations in illumination, changes in background, and the density of objects come into account. Therefore, Different augmentation techniques have been tried to see which one gives better results for the model. With just the proper data augmentation, the model would better detect air conditioning units in complex urban environments and under variable c with just the proper data augmentationonditions. The following parameters were used in the evaluation of the models. The mean Average Precision (mAP) is a standard evaluation metric for object detection, measuring the average precision across all classes. The mAP@50 indicates that a prediction is considered correct if the Intersection over Union (IoU) exceeds 50%, while mAP@50–95 averages the precision across multiple IoU thresholds ranging from 0.50 to 0.95, providing a more stringent evaluation. The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a model's accuracy, particularly when class distributions are imbalanced. It ranges from 0 to 1, where higher values indicate better performance.

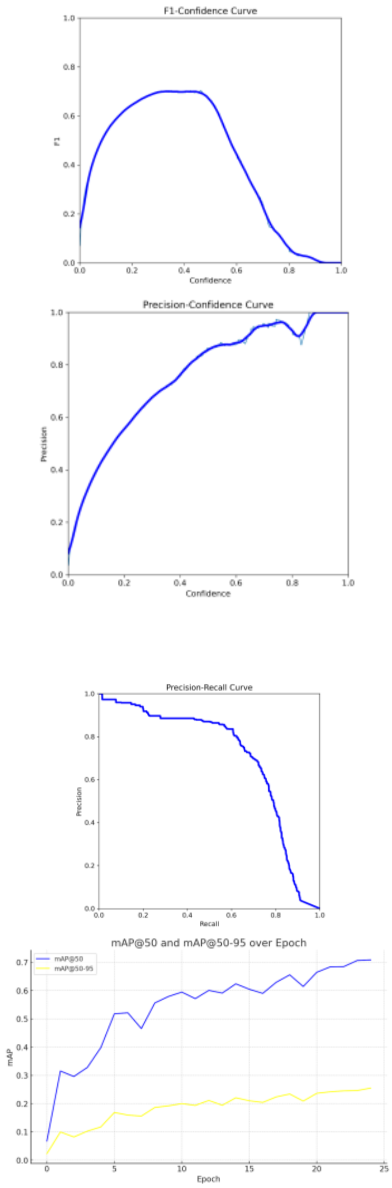
Various data augmentation strategies were applied to improve the generalization capability and performance of the YOLOv8 algorithm in detecting air conditioners in urban environments. Each strategy—outlined in Table 1—is discussed in detail below, including its objective, implementation logic, and observed outcomes.

Table 1. Comparative Analysis of YOLOv8 with Different Data Augmentation Strategies

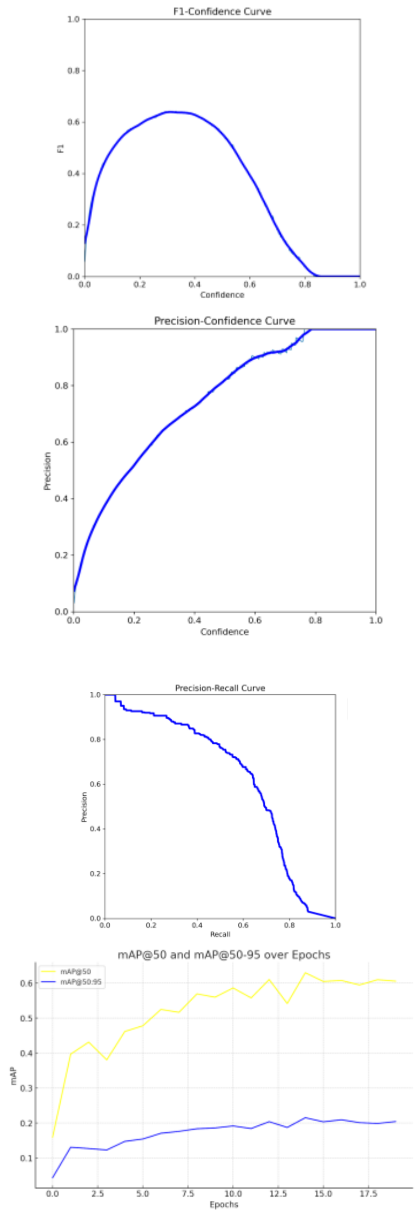
Strategy 1 – Flip, Blur, Cutout	**Strategy 2 –** Noise, Cutout, Blur
	
	
	
	

Strategy 3 – Hue, Cutout, Bounding Box 90 Rotate, Bounding Box, Shear	**Strategy 4 –** Cutout, Mosaic, Bounding Box, 90 Rotate
 	 
 	 

Strategy 5 – Noise, Cutout, Bounding Box, Rotation



Strategy 6 - Gray scale, Hue,Exposure



Strategy 1, involving flip, blur, and cutout transformations, aimed to enhance the model's ability to generalize across diverse scenarios by simulating real-world variations in images. Random flips mimicked different viewpoints, motion blur represented out-of-focus or dynamic images, and cutout obscured portions of images to train the model for partial occlusions. This approach achieved an F1 score of 0.60 at a confidence level of 0.25, with mAP@50 reaching 0.564. However, the performance at stricter Intersection over Union (IoU) thresholds was limited, indicating further room for optimization. Strategy 2 utilized noise, blur, and cutout to bolster the model's robustness against noisy and low-quality data. Noise simulated degraded image conditions, blur represented dynamic image scenarios, and cutout encouraged inference despite missing information. The model's F1 score improved to 0.63 at a confidence level of 0.4, with mAP@50 at 0.60. Although effective against noise, performance at higher IoU thresholds remained a challenge. Strategy 3, which applied hue adjustments, rotation, and shearing, aimed to make the model robust to varying angles and lighting conditions. Hue adjustments simulated lighting variations, rotation diversified object orientations, and shearing created oblique perspectives. The strategy improved the F1 score to 0.65 at a confidence level of 0.32, and mAP@50 increased to 0.62. Despite these gains, stricter IoU thresholds revealed opportunities for further refinement. Strategy 4 introduced mosaic, rotation, and cutout techniques to create a richer and more diverse dataset. Mosaic blended parts of four images to simulate cluttered scenes, rotation diversified orientations, and cutout emphasized learning from incomplete data. This approach achieved the highest F1 score of 0.70 at a confidence level of 0.32, with mAP@50 also reaching 0.70. However, challenges persisted at more stringent IoU thresholds. Strategy 5 combined noise, cutout, and bounding box rotation to address issues of noise and varied object orientations. Noise simulated low-quality data, cutout obscured regions to enhance inference capability, and bounding box rotation diversified object positioning. This strategy also achieved an F1 score of 0.70 at a confidence level of 0.32, matching Strategy 4. Robustness to noise and varied orientations improved significantly, but stricter IoU thresholds required further consideration. Strategy 6 utilized gray scale conversion, hue adjustments, and exposure changes to reduce color dependency and improve performance under varying lighting conditions. Gray scale focused on shape and texture, hue adjustments simulated lighting changes, and exposure modifications represented extreme brightness and contrast scenarios. The model achieved an F1 score of 0.64 at a confidence level of 0.32, with mAP@50 at 0.60. While the strategy proved beneficial, tighter IoU thresholds continued to pose challenges (Table 2).

Table 2. Comparative Analysis of YOLOv8 with Different Data Augmentation Strategies

Strategy	Techniques	F1-Score	Precision @1.0 Conf	Recall	mAP@50	mAP@50-95	Key Advantage
1	Flip, Blur, Cutout	0.60	0.779	~0.56	0.564	~0.20	Basic transformations, low generalization
2	Noise, Blur, Cutout	0.63	0.626	0.87	0.60	0.1–0.2	High recall, noise resistance
3	Hue, Cutout, Perspective	0.65	0.82	0.63	0.62	~0.20	Resistance to light and geometry
4	Mosaic, Rotate, Cutout	0.70	0.708	0.90	0.70	~0.20	Highest success, strong generalization
5	Resize, Noise, Rotation	0.70	0.809	0.71	0.70	~0.20	High success, geometric flexibility
6	Gray Scale, Exposure, Hue	0.64	0.790	0.62	0.60	~0.20	Light and color independence

In summary, these strategies highlight the importance of data augmentation in enhancing the performance and generalization capability of YOLOv8. To assess whether the differences in model performance across augmentation strategies were statistically significant, a one-way ANOVA test was conducted using F1-scores as the dependent variable. The results revealed a significant effect of augmentation strategy on performance ($F(5, 24) = 4.06 \times 10^{28}$, $p < 0.001$), indicating that the observed variations are not due to chance. This supports the claim that advanced augmentation methods such as Mosaic and bounding box rotation (Strategies 4 and 5) lead to significantly better generalization performance than basic transformations (Strategy 1). Furthermore, a Tukey HSD post-hoc analysis revealed that Strategy 4 (Mosaic, Rotate, Cutout) and Strategy 5 (Resize, Noise, Rotation) significantly outperformed Strategy 1 (Flip, Blur, Cutout) ($p < 0.01$). Several other strategy pairs also showed significant differences. These results statistically validate the claim that advanced geometric and compositional augmentations lead to improved model generalization and detection performance in complex urban settings. However, achieving optimal performance at stricter IoU thresholds remains an area for further research and experimentation.

4.2. Variants of Model Architecture Experiment Implementation

The different variants of the YOLOv8 model have varying depths and complexities, such as YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), and YOLOv8l (large) which differ in terms of model complexity, depth, and number of parameters. These variants allow trade-offs between inference speed and detection accuracy. Architectural characteristics of the model directly affect the detection performance, training time, and resource utilization [3]. In the task related to air conditioner detection in urban environments, high precision with efficient resource utilization is very important. Real-time applications may favor lighter models, while deeper models result in higher accuracy. Several model variants are therefore studied in these experiments to vary in performance with resource utilization. These experiments provide evidence to identify the most suitable model architecture for specific application requirements and to guide practical implementations.

Table 3. Variants of Yolo Model Performance Results

Variant	F1 Score	mAP@50	Params (M)	Size (MB)	Inference Time (ms)
YOLOv8n	0.60	0.564	3.2	5.2	3.6
YOLOv8s	0.64	0.60	11.2	21.8	6.0
YOLOv8m	0.63	0.59	25.9	40.2	8.7
YOLOv8l	0.61	0.57	43.7	76.9	11.2

It can also be deduced from the analyzed results of this experiment that the YOLOv8n has the smallest size (5.2 MB) and fastest inference time (3.6 ms per image) model produces more false-negative detections than the other models. While YOLOv8s and YOLOv8m models have similar results, YOLOv8m has slightly more stable results. Of these three, model YOLOv8s gives the highest value of the F1 score, which amounts to 0.64. This means this model has the most balanced output regarding precision and recall rates. Minimum F1 score, given by YOLOv8n, is 0.60. Among them, higher sensitivity, which refers to the model detecting more true positives, comes with the YOLOv8s model. In the case of YOLOv8n, it also has relatively lower performance. Thus, the YOLOv8s model gives the highest mAP value and optimal balance between sensitivity and recall. Although the YOLOv8s and YOLOv8m models exhibit the same recall rate, YOLOv8n demonstrates a comparatively lower value. Among them, the YOLOv8m model has the most stable and regular losses during training. For models YOLOv8n and YOLOv8s, there is more fluctuation in their verification losses. Comparing their mAP@50-95, YOLOv8s outperformed YOLOv8n and YOLOv8m. Overall Best

Performance: YOLOv8s has contributed to the best results in F1 score, mAP value, and balancing precision and recall. It has a very stable training and validation process. Efficiency Although the YOLOv8n model is faster and lighter, this model shows the poorest performance compared to the rest of the models. Most Stable Training: YOLOv8m has a steadier decline in training loss. Some fluctuations still appear in the validation process (Table 3).

4.3. Experiment on the Number of Epochs and Implementation of the Early Stop Strategy

The number of epochs governs the number of cycles the model will see from the training data; hence, it reflects how the model learns. Thus, a suitable number of epochs will give your model enough time to learn while reducing the risk of over-fitting. Early stopping is a technique that automatically halts training at the optimal point, based on performance on the validation set [5]. Training was stopped after 25 epochs based on an early stopping strategy with a patience of 5.

Figure 4. Training Progress of YOLOv8 Variant Across Epochs in Terms of Accuracy and Loss Components

Figure 4 shows the mAP and mAP@50:95 metrics over 180 epochs, indicating a steady increase and stabilization. The lower plots illustrate the evolution of Box Loss, Class Loss, and Objectness Loss, revealing early convergence and stable performance throughout training.

5. Conclusion and Discussion

This study experimentally demonstrated how the YOLOv8 algorithm can be optimized for air conditioner detection tasks in urban areas. The results revealed that data augmentation strategies, model architecture variants, and training process optimizations have a significant impact on the accuracy and generalization capability of YOLOv8.

Among the tested data augmentation strategies, methods involving mosaic blending and bounding box rotation achieved the highest performance, raising the F1 score to 70%. More basic transformations (e.g., flipping, hue adjustment, grayscale conversion) provided limited but meaningful contributions. These findings confirm that advanced augmentation techniques, which increase diversity in complex urban imagery, substantially enhance the overall effectiveness of the model.

Comparative analysis of model architectures indicated that YOLOv8s offers the most optimal balance between accuracy and inference speed. Although YOLOv8m exhibited more stable learning curves, YOLOv8s stood out with its lightweight structure and high performance. In the training phase, an early stopping strategy was applied at 25 epochs to prevent overfitting. This result highlights the importance of careful hyperparameter tuning, particularly when working with limited datasets.

Overall, the findings demonstrate that YOLOv8 can be significantly strengthened for urban object detection tasks through data augmentation, architectural selection, and training strategies. The air conditioner detection outcomes achieved with YOLOv8 can serve as a valuable decision support tool, especially in applications such as energy efficiency auditing, environmental analysis, and urban planning.

This contribution forms part of a broader effort to evaluate air conditioner detector performance in urban settings and aligns with other YOLO-based studies in the literature. A general comparison with related work is provided below. The YOLOv8 algorithm used in this study offers significant improvements over YOLOv3 in terms of both accuracy and speed. Through the application of data augmentation techniques, the training dataset was diversified, enhancing YOLOv8’s generalization ability—particularly in complex urban environments—compared to YOLOv3 as introduced by Ali Farhadi and Redmon (2018). The augmentation techniques used in this study also improved YOLOv8's robustness against complex backgrounds, especially in densely populated and visually cluttered urban areas. Similarly, the accuracy of YOLOv8 was evaluated under urban complexity and varying illumination conditions. Our experiments explored different YOLOv8 variants—nano, small, and medium—and identified their trade-offs between accuracy and speed. As a newer version, YOLOv8 was observed to achieve improved performance metrics consistent with findings from earlier versions [19]. The adopted augmentation methods—primarily including rotation, blurring, and noise addition—substantially enhanced model performance in challenging urban terrains. The study significantly contributes to the literature by demonstrating how data augmentation can improve generalization capability. Our findings align with similar scenarios in which YOLOv8 was applied for air conditioner detection using urban datasets such as KITTI. The performance of YOLOv8 in complex urban environments outperformed earlier YOLO versions with consistent superiority [6].

This paper employed datasets specific to a certain geographic region for air conditioner detection. Therefore, future studies may focus on more generalized models by incorporating diversified urban datasets from multiple geographical areas. Additional variables such as different climatic conditions, building structures, and urban morphological factors should also be considered. It would be of particular interest to examine YOLOv8’s performance under extreme conditions—such as nighttime, low light, rain, or fog. These directions could support further research aimed at improving model robustness through more advanced augmentation strategies under such conditions.

The present study offers substantial contributions through the use of diverse data augmentation techniques that significantly enhanced model performance. Future work will focus on incorporating images from broader geographic regions, introducing various building and climate types, and evaluating model robustness under challenging atmospheric conditions. In addition, the implementation of targeted augmentation techniques for underrepresented classes, along with training on high-resolution imagery, is expected to further improve model accuracy and generalization.

Declarations

Ethical Consideration

This study did not require formal ethical approval because it relied exclusively on publicly accessible, open-source data.

Competing interests

The authors declare that no competing financial interests or personal relationships exist that could have appeared to influence the work reported in this paper.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Farhadi, A., & Redmon, J. (2018, June). Yolov3: An incremental improvement. In Computer vision and pattern recognition (Vol. 1804, pp. 1-6). Berlin/Heidelberg, Germany: Springer.Ref-2

[2] Flores-Calero, M., Astudillo, C. A., Guevara, D., Maza, J., Lita, B. S., Defaz, B., ... & Armingol Moreno, J. M. (2024). Traffic sign detection and recognition using YOLO object detection algorithm: A systematic review. Mathematics, 12(2), 297.Ref-4

[3] Vinh, T. Q., & Long, P. H. (2023, November). Pedestrian Detection Using YOLO with Improved Attention Module. In 2023 International Conference on Advanced Computing and Analytics (ACOMPA) (pp. 93-98). IEEE.

[4] Cong, X., Li, S., Chen, F., Liu, C., & Meng, Y. (2023). A review of YOLO object detection algorithms based on deep learning. Frontiers in Computing and Intelligent Systems, 4(2), 17-20.

[5] Koonce, B., & Koonce, B. (2021). SqueezeNet. Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization, 73-85.

[6] Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3354-3361). IEEE.

[7] Ruan, J., & Wang, Z. (2020). An improved algorithm for dense object detection based on YOLO. In The 8th international conference on computer engineering and networks (CENet2018) (pp. 592-599). Springer International Publishing.

[8] Özcan, İ., Altun, Y., & Parlak, C. (2024). Improving YOLO Detection Performance of Autonomous Vehicles in Adverse Weather Conditions Using Metaheuristic Algorithms. Applied Sciences, 14(13), 5841.

[9] Chen, L. C., Sheu, R. K., Peng, W. Y., Wu, J. H., & Tseng, C. H. (2020). Video-based parking occupancy detection for smart control system. Applied Sciences, 10(3), 1079.

[10] Kong, L., Wang, J., & Zhao, P. (2022). YOLO-G: A lightweight network model for improving the performance of military targets detection. IEEE Access, 10, 55546-55564.

[11] Cao, Y., Li, C., Peng, Y., & Ru, H. (2023). MCS-YOLO: A multiscale object detection method for autonomous driving road environment recognition. IEEE Access, 11, 22342-22354.

[12] Fang, W., Wang, L., & Ren, P. (2019). Tinier-YOLO: A real-time object detection method for constrained environments. Ieee Access, 8, 1935-1944.

[13] Ling, H., Zhao, T., Zhang, Y., & Lei, M. (2024). Engineering Vehicle Detection Based on Improved YOLOv6. Applied Sciences, 14(17), 8054.

[14] Kim, M., Jeong, J., & Kim, S. (2021). ECAP-YOLO: Efficient channel attention pyramid YOLO for small object detection in aerial image. Remote Sensing, 13(23), 4851.

[15] Doshi, Y. (2024). Comparison of YOLO Models for Object Detection from Parking Spot Images. Educational Administration: Theory and Practice, 30 (4), 10401-10411.

[16] Haimer, Z., Mateur, K., Farhan, Y., & Madi, A. A. (2023, May). Yolo algorithms performance comparison for object detection in adverse weather conditions. In 2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET) (pp. 1-7). IEEE.

[17] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149.

[18] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

[19] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.