ISSN 2085-4552

Implementation of YOLOv8 in Object Recognition Systems for Public Area Security in Kebun Raya Bogor

Prihandoko[1], Sri Agustina Rumapea[2], Muhamad Faishal Fawwaz[3]
[1,3]Departemen of Informatics, Gunadarma University, Depok, Indonesia
[2] Faculty of Computer Science, Universitas Methodist Indonesia, Medan, Indonesia
[1]pri@staff.gunadarma.ac.id, [2]sriagustina78@gmail.com, [3]muhamadfaishalfawwaz44@gmail.com

Accepted 29 April 2025
Approved 29 April 2025

***Abstract*— Pedestrian areas often serve as centers of high public activity, requiring intelligent monitoring systems to ensure user safety and comfort. Computer vision technology, especially object detection, provides a promising method for identifying and counting individuals in public spaces. This study implements YOLOv8 to develop a human detection and crowd counting model in the pedestrian zones of Bogor Botanical Garden. Researchers collected images and videos from three strategic locations and annotated them using Roboflow with a single class labeled "person." They trained the model on Google Colab using a Region of Interest (ROI)-based method and evaluated it through metrics like precision, recall, F1-score, confusion matrix, and mean Average Precision (mAP). The model achieved a precision of 0.846, recall of 0.858, F1-score of 0.85, and mAP@50 of 0.951, though mAP@50-95 dropped to 0.586. These findings show YOLOv8 provides strong real-time pedestrian detection, though enhancing precision in complex environments remains challenging.**

***Index Terms*— *crowd counting; deep learning; object detection; pedestrian surveillance; YOLOv8*.**

I. INTRODUCTION

Crowds in public spaces, including pedestrian areas, town squares, open markets, and sports venues, present significant challenges related to security and safety. High-density human gatherings not only impact comfort levels but also elevate the risk of incidents such as accidents and criminal activities. Poorly managed large crowds have occasionally led to tragic outcomes, exemplified by the crowd crush disaster in Itaewon, Seoul, in 2022, which resulted in over 150 fatalities [1]. With the continuous increase in public events and activities, effective crowd management and monitoring have become imperative. In response to these challenges, artificial intelligence (AI) technologies, particularly object detection and crowd counting methods, offer promising avenues to enhance safety and security.

Numerous prior studies have investigated various deep learning approaches, including CNN, SSD, YOLO, and R-CNN, for human detection and counting tasks. Among these methods, the YOLO (You Only Look Once) algorithm, specifically its latest iteration YOLOv8, has gained recognition due to its superior real-time detection capabilities and high accuracy under complex environmental conditions [2]. YOLOv8 significantly improves human behavior detection accuracy by approximately 4.2% compared to previous versions [3]. Furthermore, YOLOv8 achieved high precision (94.32%) and recall (91.17%) in complex scenarios such as passenger detection in elevators [4]. Despite these advancements, existing research identifies persistent limitations in accurately detecting individuals within densely crowded and complex environments. Architectural modifications to the YOLOv8 model are necessary to improve its mean Average Precision (mAP) [5]. Additionally, although YOLOv8 offers promising performance, it still struggles with distant or partially obstructed objects [6]. Therefore, there remains a critical need to further evaluate and refine the YOLOv8 algorithm under real-world, challenging conditions.

The YOLOv8 method is highly suitable for implementation in research related to object recognition systems for public area security at Kebun Raya Bogor, as this model achieves an optimal balance between detection speed and accuracy—both critical aspects for real-time scenarios. Compared to YOLOv9, which, despite offering slightly better accuracy, introduces higher complexity resulting in increased inference times and computational demands, YOLOv8 provides a more practical and resource-efficient solution for real-world deployments. Meanwhile, Faster R-CNN, although known for high detection accuracy, employs a two-stage detection process that significantly increases inference latency, making it less effective for real-time monitoring in public spaces. Thus, YOLOv8 emerges as the most appropriate choice in this study, effectively

addressing the need for rapid, efficient, and sufficiently accurate object detection in real-time public surveillance applications.

Addressing the existing challenges and building upon previous research findings, the present study aims to develop and evaluate a deep learning-based model employing the YOLOv8 algorithm specifically for the detection and counting of individuals within crowds at the pedestrian areas of Bogor Botanical Gardens. The primary objective of this study is to create a reliable and real-time monitoring system capable of enhancing public safety by promptly identifying human density levels.

To achieve this objective, the research employs an experimental approach involving the collection of image and video datasets from pedestrian areas within Bogor Botanical Gardens. The collected datasets will undergo preprocessing stages, including image resizing and annotation with bounding boxes, facilitated through the Roboflow platform. Subsequently, data will be partitioned into training, validation, and testing subsets to provide comprehensive model evaluation.

II. LITERATURE REVIEW

2.1 Pedestrian Zones and Urban Life

Pedestrian zones are a key part of modern urban planning. These spaces are designed to encourage people to walk, interact, and feel safe and comfortable in the city. When done right, pedestrian zones don't just improve the walkability of a place—they also reduce traffic congestion and lower air pollution. To make pedestrian zones work effectively, planners need to consider how people move through the area, how safe and accessible it is, and how well it connects with other parts of the city [7-9].

But as more people gather in public spaces—especially during events or busy times—it becomes more difficult to manage and monitor those areas. That's where technology, like crowd monitoring and computer vision, starts to play an important role.

2.2 Counting Crowds

Crowd counting is all about estimating how many people are in a space and understanding how they're distributed. It's used in many settings—like public safety, traffic control, and event planning. But this task is far from simple. When crowds are dense or people are partially blocked from view (what we call occlusion), it becomes much harder to accurately count them. The problem gets even trickier when people are moving or spread out unevenly [10].

Crowd counting can be approached in two main ways. The first is supervised learning, where the system is trained using data that's already labeled—so it knows, for example, what a "person" looks like. The second is unsupervised learning, where the system has to figure out for itself how to group and interpret the data without any prior labeling [11].

2.3 Computer Vision

Computer vision is a branch of artificial intelligence that teaches machines to "see" and make sense of visual data, just like humans do. Thanks to powerful models called convolutional neural networks (CNNs), machines can now detect patterns, shapes, and movements in images and videos with impressive accuracy. These systems can recognize what's in a scene, understand how things relate to each other, and even spot unusual activity [12-14].

For crowded areas, computer vision allows systems to automatically scan camera footage and detect how many people are present—without needing a human operator to watch every frame.

2.4 Object Detection

At the heart of many computer vision systems is object detection. This technology lets a computer recognize and locate multiple objects—like people, cars, or bicycles—in a single image. It does this by drawing bounding boxes around the objects and labeling them. There are two types of object detectors:

- Two-stage detectors, such as Faster R-CNN, first identify possible object locations, then analyze each one to decide what's inside.
- One-stage detectors, like YOLO and SSD, skip the proposal step and predict everything at once, making them faster and more efficient for real-time use.

To measure how well these models perform, we use metrics like precision (how many detections were correct), recall (how many real objects were found), and IoU (Intersection over Union)—a measure of how closely the predicted box matches the actual object.

2.5 YOLOv8

One of the most impressive object detection models today is YOLOv8. It's the latest version of the popular "You Only Look Once" family of models, and it's packed with upgrades. Unlike older versions that rely on predefined "anchor boxes," YOLOv8 is anchor-free, which makes it simpler and more flexible when detecting objects of different sizes.

It also uses something called decoupled heads—basically, separate parts of the model for deciding what an object is and where it is. This helps it make better predictions. YOLOv8 also adds smarter loss functions like CIoU and DFL, which help the model learn more efficiently and predict bounding boxes more precisely.

What's more, YOLOv8 isn't limited to object detection. It can also do instance segmentation, pose estimation, and image classification, making it a versatile choice for many real-world applications. It runs fast—up to 60 frames per second—and handles

cluttered, crowded scenes better than previous models [2, 4].

III. METHODOLOGY

This research utilizes the YOLO version 8 or YOLOv8 method. The use of YOLO in machine learning aims to detect and classify specific objects in images or videos. The results of detection and classification using YOLO can be used to count the number of objects that pass through a specific area or that have been defined in an image or a video frame. This research aims to detect people in images or videos at pedestrian areas in Kebun Raya Bogor.

This study employs an experimental approach to evaluate the performance of the YOLOv8 algorithm for detecting pedestrians in crowded urban areas. The overall stages of the research are presented in Fig. 1.

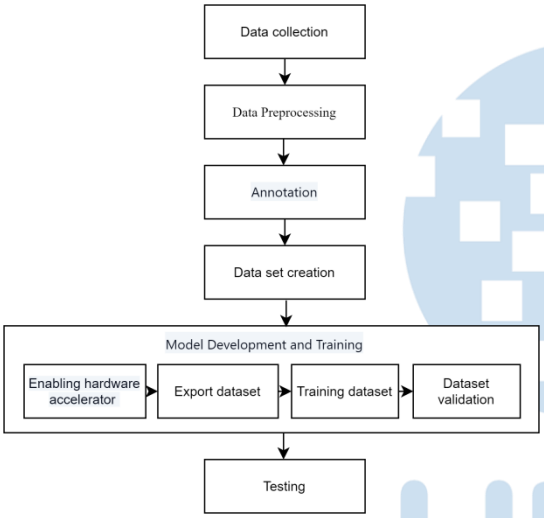

Fig. 1. Research Framework

a) Data collection

The research begins with the collection of data in the form of images and videos. After data collection, pre-processing is carried out to prepare the data for further analysis. Once the pre-processing stage is completed, the data are annotated and divided into three categories: training data, validation data, and test data. An example of the data collection location can be seen in Fig. 2. After the annotation and data division processes, the design and coding of the detection model are conducted. The final stage involves testing, where the developed model is evaluated to assess its accuracy in detecting people within Kebun Raya Bogor.

Fig. 2. Image of Pedestrian Area near Gate 3

b) Data Pre-processing

After collecting images and videos at three designated points, a project was established within the Roboflow Workspace, a software tool designed for data storage and annotation to facilitate the creation of a dataset. The initial step in the workflow involved defining a Class in Roboflow, specifically designated for "person," which serves as the identifier for the object to be detected in the study. Table 1 presents the details of the images per frame that were successfully collected from the three capture points."

Subsequently, the collected image and video files were uploaded to the platform. A crucial step in processing the video data involved extracting frames at a rate of one frame per second from each video to transform these into a uniform image format for further analysis. This frame rate was chosen to balance the need for detailed temporal resolution while managing the volume of data processed and stored, ensuring efficient handling during the subsequent stages of model training and validation.

TABLE 1. NUMBER OF IMAGES FROM CAPTURE POINTS

No	Collection poin	Number of images
1	Pedestrian near Gate 3	416
2	Pedestrian near Gate 4	418
3	Sempur Park	367
Total		**1201**

c) Annotation

This stage begins after the collection of image-formatted data and focuses primarily on identifying the "person" object within the images by using the annotation feature on Roboflow. During the annotation process, each image is carefully marked with a bounding box around each detected person. This labeling is performed manually or under supervision to ensure accuracy and consistency in object identification. The annotation process can be seen in Fig. 3.

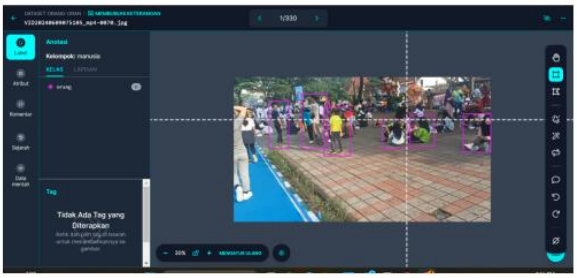

Fig. 3. Image Annotation

After labeling all images, the dataset is divided into three distinct subsets: training data, validation data, and test data. The training data is used to adjust the model's parameters and optimize its performance during the learning process. The validation data serves to periodically evaluate the model throughout training, helping to detect signs of overfitting and ensure generalization. Meanwhile, the test data is reserved for assessing the model's final performance after training is complete. The detailed distribution of the dataset across these categories is presented in Table 2, ensuring clarity and reproducibility for further analysis.

TABLE 2. IMAGE DATA DIVISION

No	Data type	Percentage	Amount of data
1	Training Data	70%	840
2	Validation Data	20%	241
3	Test Data	10%	120
Total		100%	1201

After specifying the percentages for data division, Roboflow automatically partitions the image data into three categories at random. The results of this data segmentation can be observed in Fig. 4, where each image is clearly marked to indicate its classification as training, validation, or test data, noted at the bottom left corner of each image.

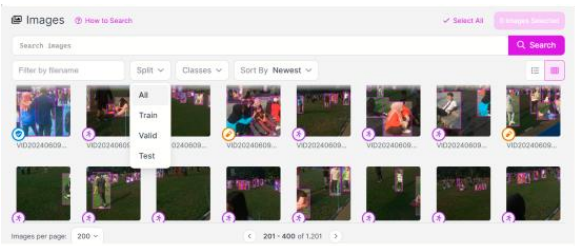

Fig. 4. Results of the Image Data Division

d) Dataset Formation

After dividing the data into three parts, the next step is to create a dataset for research purposes. However, before constructing the dataset, a preprocessing stage is required. During this stage, the images are resized to ensure that they all have the same resolution, facilitating consistent data handling during the subsequent analysis. In the preprocessing stage, all images are resized to a uniform resolution of 1280 x 720 pixels. The fitting format 'Fit (black edges) in' is selected to ensure that images with different original resolutions do not distort—either by stretching or compressing—when compiled into the dataset.

e) Model Development and Testing

In the Model Development and Testing phase, a systematic approach was undertaken to build an effective person detection system tailored for the pedestrian zones of Bogor Botanical Gardens. This phase began with the preparation of the dataset, which included image annotation and segmentation into training, validation, and test sets. Following this, the focus shifted to the design and coding stage, where the YOLOv8 algorithm—renowned for its high performance in real-time object detection—was used to train the model. The model was trained using annotated data to recognize and count individuals, addressing a key challenge in maintaining public safety in densely populated areas.

After training, the model was tested to evaluate its accuracy and reliability under real-world conditions, such as varying lighting and crowd density. This step is critical to ensure that the model performs effectively in dynamic urban environments. The development phase highlights two essential components: training the YOLOv8 model and testing its performance using the annotated dataset. The detailed steps involved in this stage are described in the following sections

1. Enabling hardware accelerator GPUT4

This research was conducted using Google Colab as the computational platform. By default, Google Colab operates on a Central Processing Unit (CPU), which tends to be slower in executing machine learning processes. To address this limitation, a T4 GPU was utilized, which is more effective in accelerating the training and inference of models in machine learning and deep learning. GPUs, particularly the T4 model provided for free by Google Colab, offer better energy efficiency and superior optimization for matrix and vector operations compared to CPUs.

2. Export Dataset

To export the dataset that has been created, it is necessary to copy the API code provided at the dataset formation stage. This code contains the necessary information to access and manage the organized dataset.

3. Training Dataset

The training dataset plays a crucial role in the development of machine learning models, serving as the primary resource for teaching the model to accurately recognize and predict outcomes. This dataset consists of pre-processed and annotated images used to adjust the model's parameters through supervised learning techniques. During training, the

model iteratively learns from this dataset by comparing its predictions against the actual outcomes, continuously improving its accuracy. In the code shown below, the dataset is trained using 100 epochs with an image size of 800 pixels. One epoch means the model has learned or recognized each data sample in the training dataset once. Increasing the number of epochs allows the model to become more familiar with the form of the object being trained. The adequacy of the training dataset directly influences the model's ability to generalize to new, unseen data, making the quality and diversity of the training examples paramount. Consequently, ensuring a comprehensive and representative training dataset is essential for the successful application of the model in real-world scenarios.

```
!yolo task=detect mode=train model={model} data={dataset_location}/data.yaml epoch=100 imgsz=800 plots=True
```

4. Dataset Validation

The validation dataset is a collection of data used for an objective evaluation of a model's performance during the training process. Not involved in adjusting model parameters, the validation dataset plays a critical role in identifying issues such as overfitting, where the model perfectly fits the training data but fails to generalize to new data. The code shown below is used to validate data, ensuring that the trained model is tested using this validation dataset. This validation process is crucial for assessing how effectively the model can predict new data and for adjusting the model to achieve an optimal balance between learning and generalization capabilities. Using this validation dataset allows developers to optimize the model, ensuring that the best-trained model performs well on the same data used during training.

```
!yolo task=detect mode=val model={HOME}/runs/detect/train/weight/best.pt data={dataset.location}/data.yaml
```

f) Testing

In the testing phase, the trained system is evaluated to assess its object detection and classification capabilities under real-world conditions. This process involves the use of video files that were previously uploaded or stored in the Google Colab directory. The test results are presented in video outputs that display bounding boxes around "person" objects, accompanied by confidence scores ranging from 0.25 to 1, indicating the level of prediction accuracy. Evaluation is conducted not only through visual analysis of the output videos but also by measuring performance metrics such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive overview of the model's effectiveness in generalizing and detecting new, unseen objects beyond the training process. Therefore, the testing phase is a critical step in objectively validating the model's performance across various real-world conditions and scenarios.

The following outputs are used to evaluate the accuracy of the person detection model in this study:

- Confusion Matrix
  A performance evaluation table that shows the number of correct and incorrect predictions made by the model for each class. This matrix allows for detailed identification of classification errors.
- Normalized Confusion Matrix
  A normalized version of the confusion matrix, where each value is divided by the total number of predictions for the corresponding class. It provides proportions that are easier to compare across classes.
- F1-Score
  An evaluation metric that combines precision and recall into a single harmonic mean value. The F1-score is particularly useful when there is an imbalance between the number of positive and negative classes.
- Precision
  Indicates the proportion of positive predictions that are truly relevant (correct). In other words, precision measures how accurate the model is when making positive predictions.
- Recall
  Measures the model's ability to correctly identify all actual positive cases. Recall shows how many of the total positive cases were successfully detected by the model.
- Precision-Recall Curve
  A graph that illustrates the trade-off between precision and recall at various threshold levels. This curve is useful for evaluating model performance, especially in cases with imbalanced data.
- Train Batch
  A subset of the training dataset used to update the model's weights during one training iteration. In object detection, train batch images are often visualized to observe interim detection results during training.
- MAP (Mean Average Precision)
  A widely used metric for evaluating object detection models. mAP is the average of the Average Precision (AP) values across all classes, where AP is calculated as the area under the Precision-Recall curve for a given class. The mAP score provides a comprehensive overview of the model's ability to consistently detect objects across multiple categories.

To evaluate the performance of the detection model, four key parameters are used based on the model's classification results on the test data:

1. True Positives (TP): Cases where the model correctly classifies a positive object.
2. True Negatives (TN): Cases where the model correctly identifies a negative (non-target) object.
3. False Positives (FP): Errors where the model incorrectly classifies a negative object as positive.
4. False Negatives (FN): Errors where the model fails to detect a positive object and classifies it as negative.

These four parameters form the basis for calculating evaluation metrics such as precision, recall, and F1-score, which provide a comprehensive overview of the model's accuracy and reliability in object detection.

IV. RESULT

The model evaluation stage aims to measure how well the YOLOv8 model performs during training. In this study, the model was trained for 100 epochs, taking approximately one hour, using 241 validation images. The training outcomes include several key metrics:

- Precision (P): The model achieved a precision of 0.846, which indicates a high proportion of correctly predicted bounding boxes that actually contain a person. This value, being close to 1, reflects good model accuracy.
- Recall (R): With a recall of 0.858, the model successfully detected a large portion of actual people present in the images.
- mAP50: The model scored 0.951 at an 0.01 threshold of 0.5, demonstrating excellent performance in identifying objects at lower threshold levels.
- mAP50-95: The score dropped to 0.586 across stricter IoU thresholds (0.50 to 0.95), indicating reduced performance under more precise detection requirements.

These metrics illustrate that while the model performs exceptionally well under standard threshold conditions, its precision decreases as the required overlap for correct predictions increases.

4.1. Deeper Analysis of Performance Drop

A deeper analysis is necessary to identify the factors contributing to the model's performance drop, particularly the decline in mAP across stricter IoU thresholds (mAP@0.5:0.95). This decrease indicates that although the model performs well at a standard threshold, its localization precision diminishes when tighter bounding box overlaps are required.

One major cause of this performance decline is the occurrence of false positives, where the model incorrectly classifies background elements as humans. For example, objects such as leaves, tree branches, shadows, or other background features with human-like visual patterns were often misclassified, especially under challenging lighting conditions, low contrast, or dynamic backgrounds caused by wind. This issue emerges because the YOLOv8 model, despite its robust capabilities, sometimes learns features that also exist in non-human objects. Additionally, limitations in the training dataset—such as a lack of diversity in human poses, environmental conditions, and movement variations—exacerbate the risk of misclassification.

From a practical standpoint, this degradation in precision significantly affects the deployment of the model for public area surveillance. A high rate of false positives could trigger excessive false alarms, reducing the operational efficiency and trustworthiness of the surveillance system. In public safety scenarios, this could lead to confusion among security personnel and delayed responses to actual threats. Therefore, to enhance the model's practical effectiveness, further improvements are needed. These may include enriching the dataset with more varied real-world examples, adjusting the model architecture for better feature discrimination, and applying additional post-processing techniques to filter out background-induced false positives.

4.2 Evaluation of Confidence Scores

The training results in this study also utilized four types of graphs or curves as tools to compare the model's accuracy in relation to its confidence levels. These curves include the Precision-Confidence Curve, Recall-Confidence Curve, and the Precision-Recall Curve. These visualizations are used to analyze how variations in confidence thresholds affect the model's predictive performance.

Recall-Confidence Curve

As shown in Fig. 5, the x-axis represents the confidence threshold ranging from 0 to 1, while the y-axis shows precision. The curve demonstrates that as confidence increases, so does the precision, peaking at 1.00 when the confidence threshold reaches 0.956. This indicates that the model's predictions become more accurate at higher confidence levels, especially in detecting the "person" class.

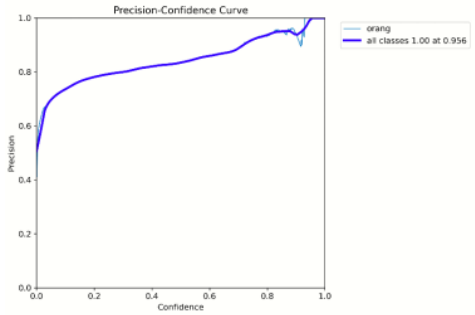

Fig. 5. Recall-Confidence Curve

Precision-Recall Curve

Fig. 6 illustrates the relationship between recall and confidence. Initially, the model maintains a high recall of 0.97 at a confidence level of 0.00. However, recall drops significantly as the confidence threshold increases. This trend suggests that while the model detects more objects at lower confidence, it becomes conservative at higher thresholds, resulting in missed detections.

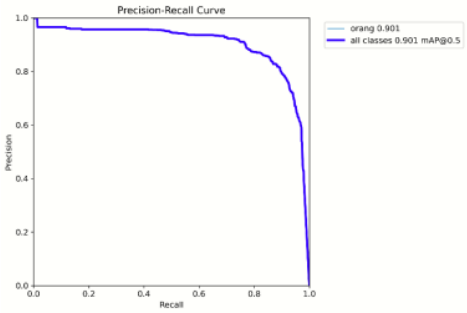

Fig. 6. Precision-Recall Curve

F1-Confidence Curve

Fig. 7 shows that F1-score, which harmonizes precision and recall, is stable at moderate confidence levels and begins to decline as confidence approaches 1. At a confidence level of 0.561, the model achieves an F1-score of 0.85, suggesting a well-balanced performance at this threshold. However, overconfidence may lead to reduced effectiveness due to increased false predictions.

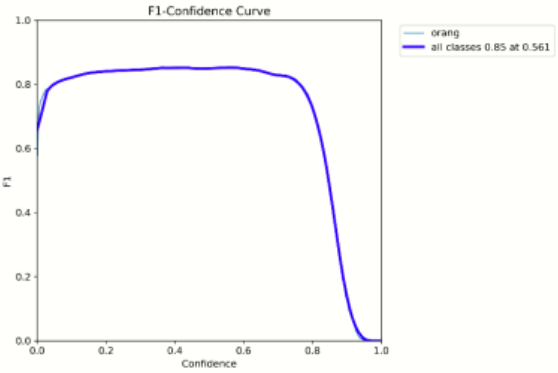

Fig. 7. F1-Confidence Curve

4.3 Confusion Matrix Evaluation

The confusion matrix analysis, presented in Fig. 8, provides a detailed breakdown of the model's classification results:

a. True Positives (TP): 0.90
b. True Negatives (TN): 0.00
c. False Positives (FP): 1.00
d. False Negatives (FN): 0.10

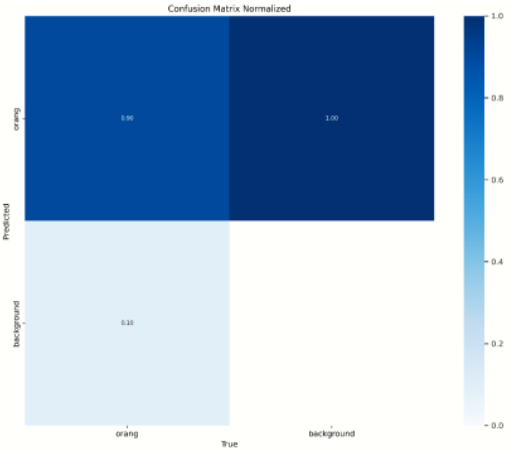

Fig. 8. Confusion Matrix Model

Based on these values, the following metrics are calculated:

a. Precision

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Precision = \frac{0.90}{0.90+1.00} = 0.47$$

b. Recall

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$Recall = \frac{0.90}{0.90+0.10} = 0.90$$

C. F1-score

$$F1score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (3)$$

$$F1score = 2 \times \frac{0.47 \times 0.90}{0.47+0.90} = 0.62$$

D. Accuracy

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

$$Accuracy = \frac{0.90+0}{0.90+1.00+0.10+0} = 0.45$$

Although the recall is high, the model's high false positive rate negatively affects precision, leading to a moderate F1-score. This result suggests that while the model is effective at detecting people, it struggles to distinguish them from background elements.

4.4. Evaluation Using Scatter Plot and Box Plot

Scatter plots are employed to visualize the distribution of bounding box coordinates and dimensions. These visual tools help in understanding the spatial and size characteristics of detected objects.

The (x, y) scatter plot in Fig. 9 shows that most bounding box centers are clustered between x = 0.1–0.4 and y = 0.2–0.6.

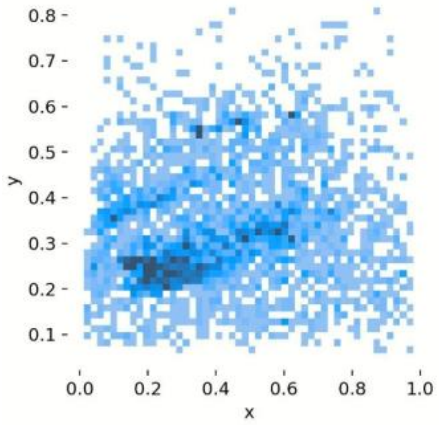

Fig. 9. Scatter Plot x,y

The (width, height) plot in Fig. 10 indicates that bounding boxes are generally taller than wide, with heights concentrated between 0.1–0.2 and widths between 0.0–0.1.

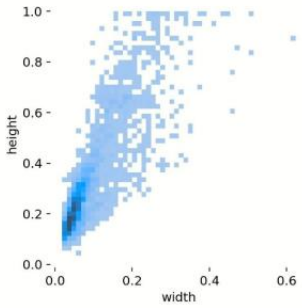

Fig. 10. Scatter plot (width, height)

Fig. 11 includes a bar chart showing fewer than 3500 bounding boxes labeled as "person" in the dataset, alongside an overlay of trained bounding boxes, demonstrating the model's coverage.

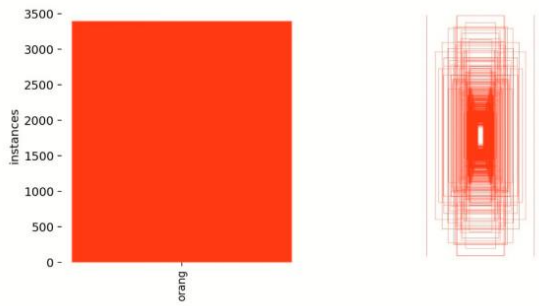

Fig. 11. Bar chart & Bounding box overlays

4.5 Training and Validation Metrics

Training and validation metrics serve as essential indicators of the model's learning progress and generalization ability. These metrics, which include loss functions, precision, recall, and mean average precision (mAP), provide insights into how well the model performs on both seen (training) and unseen (validation) data during each epoch of training. Fig. 12 provides an overall indication that the model is undergoing an effective learning process and progressively improving its performance during both training and validation phases.

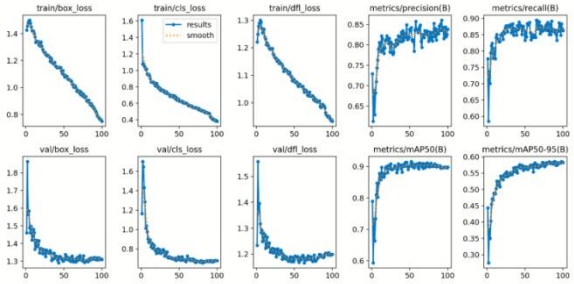

Fig. 12. Matrix & Data Validation

- The loss values (train/box_loss, train/cls_loss, train/dfl_loss) steadily decrease throughout training, indicating that the model is learning effectively in terms of localization, classification, and feature extraction.
- The validation losses (val/box_loss, val/cls_loss, val/dfl_loss) follow a similar downward trend, suggesting good generalization to unseen data.
- Metrics such as precision and recall consistently improve across epochs, showing progressive enhancement in detection performance.
- The increasing trends in mAP@0.5 and mAP@0.5:0.95 further confirm that the model's accuracy level continues to improve throughout the training and validation phases.

These graphical metrics collectively indicate that the model undergoes a positive learning process and progressively enhances its performance over time.

4.6 Train Batch

The concept of train batch refers to the process of training the model using grouped subsets of data, allowing for more stable and efficient learning. Train batching also helps minimize fluctuations in the object representations being learned, ensuring the model can generalize better and produce more consistent results. In this study, the training batch was divided into six parts derived from a total of 840 training images.

Fig. 13 presents the initial training batch, consisting of the first subset of training images introduced to the model at the beginning of the learning process. This batch plays a crucial role in setting the initial learning direction by introducing the model to the fundamental features of the target object.

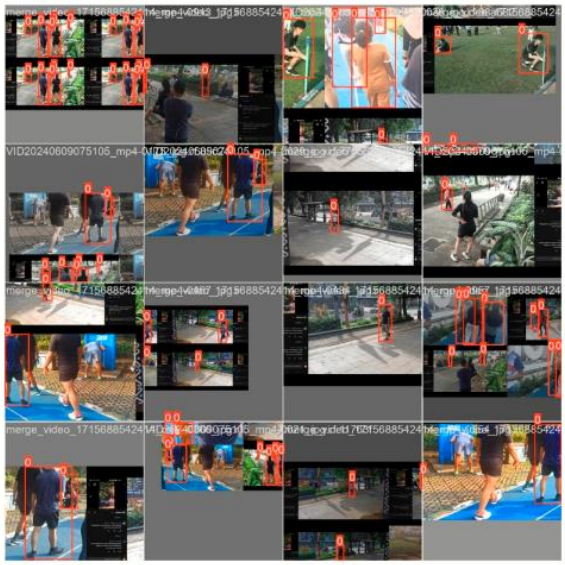

Fig. 13. Initial training batch

In contrast, Fig. 14 illustrates the final training batch, which comprises the last subset of images used during the concluding stage of training. This batch reinforces the model's learned patterns and helps stabilize its performance prior to final evaluation It helps reinforce the patterns and features the model has learned, ensuring the consistency and stability of detection performance by the end of the training cycle.

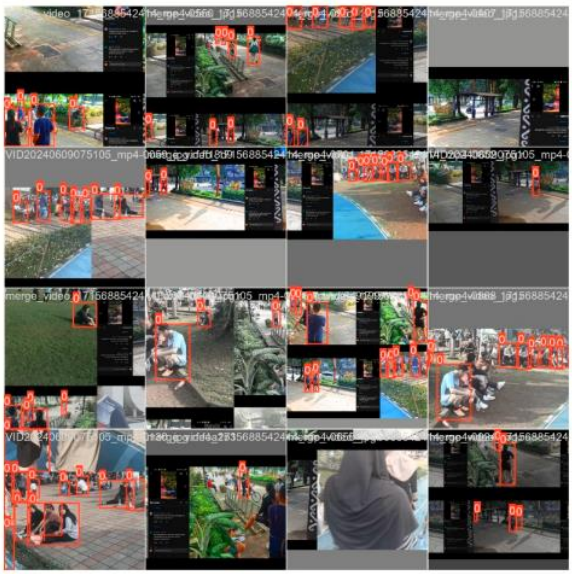

Fig. 14. Final training batch

4.7 Model Output Evaluation

The model's output was evaluated using external video samples from YouTube. Each video was analyzed for detection accuracy in various scenarios, such as crowd density, object distance, and occlusion.

- Accurately detects individuals who are visible and unobstructed.
- Struggles to detect individuals who are distant or partially blocked.
- Occasionally misclassifies non-human objects as "person" (e.g., leaves or fruit).

These findings confirm that while the model performs well under ideal conditions, its accuracy declines in complex environments, suggesting areas for further improvement.

V. CONCLUSIONS

This study successfully developed a YOLOv8-based pedestrian detection system at Bogor Botanical Garden by training it on 1,201 annotated images. Overall, the model demonstrated strong performance at moderate confidence thresholds, although its accuracy declined when stricter precision requirements were applied, particularly under more demanding IoU conditions.

The model, trained over 100 epochs with confidence thresholds ranging from 0.25 to 1.00, achieved a high precision of 0.846, a recall of 0.858, and a strong mAP@0.5 of 0.951. However, its mAP@0.5:0.95 dropped to 0.586, suggesting that while the model could detect people effectively under standard settings, it struggled with finer, more precise object localization.

Confidence-based analysis revealed that the model performed best at specific thresholds: precision peaked at a confidence of 0.956, recall reached 0.97 at a confidence of 0.00, and the highest F1-score of 0.85 was achieved at a confidence of 0.561. The Precision-Recall Curve also showed a high mAP of 0.901, indicating a good balance between precision and recall across different thresholds.

Further evaluation through the confusion matrix highlighted challenges in differentiating humans from similar-looking background elements. Although the model achieved a high recall (0.90), the precision dropped to 0.47 due to frequent false positives, resulting in a moderate F1-score of 0.62 and an overall accuracy of 0.45. These findings suggest that while the model was effective at detecting people, it sometimes misclassified objects such as leaves, shadows, or tree branches as human figures.

Scatter plot visualizations showed that most bounding box centers clustered between x = 0.1–0.4 and y = 0.2–0.6, consistent with the upright posture of standing humans. A width-to-height ratio of roughly 6:10 further supported that the model mainly detected vertical, human-like shapes. Analysis of training batches also confirmed that the model's learning process was stable and systematic throughout.

When tested on real-world videos, the model performed well in detecting clearly visible, unobstructed individuals but struggled with detecting people who were distant or partially blocked from view. Some false positives also occurred, especially when background objects resembled human features.

From a practical standpoint, the YOLOv8-based system shows real potential for integration into real-time CCTV surveillance platforms, particularly in public spaces like parks, squares, or transit areas. Thanks to its relatively lightweight design and efficient inference speed, the model can operate on moderately powered GPUs or even edge computing devices, making it a practical solution for real-world deployments. However, further fine-tuning and optimizations would be necessary to ensure reliable performance in dense or complex environments before full-scale adoption.

Future research is recommended to explore the use of keypoint detection for more precise recognition of human posture, replacing the current bounding box approach. Expanding the dataset with more diverse human movement patterns could further improve model accuracy. Additionally, optimizing the model for real-time detection and integrating it into CCTV-based surveillance systems would enhance its practical use in public safety applications

REFERENCES

[1] BBC News. (2022, October 30). Itaewon crush: At least 153 killed in South Korea Halloween crowd surge. https://www.bbc.com/news/world-asia-63440858

[2] Liu, Q., Zhang, Y., Zhao, X., & Wu, L. (2023). YOLOv8: Real-time object detection for complex scenes. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1032–1041.

[3] Chen, L., Zhang, Y., & Zhao, H. (2023). Enhancing human behavior recognition using YOLOv8 in surveillance videos. Computer Vision and Image Understanding, 225, 103553.

[4] Wang, X., Chen, Y., & Li, T. (2024). Improved YOLOv8 for small object detection in urban surveillance. Sensors and Systems, 48(2), 204–219.

[5] Setiyadi, R., Santosa, P. I., & Nurhasanah, N. (2023). Improving object detection precision through architectural adjustment of YOLOv8 in dense crowd environments. Journal of Applied AI Research, 9(1), 89–98.

[6] Drantantiyas, D., Nugroho, Y., & Widodo, B. (2023). Evaluating YOLOv8 performance for occluded object detection in surveillance systems. Journal of Computer Science and Applications, 15(4), 321–329.

[7] Gehl, J. (2019). Cities for People. Washington, DC: Island Press.

[8] Muchlisin, Z. A. (2020). Desain kawasan pejalan kaki berbasis kenyamanan dan keamanan. Jurnal Ilmiah Teknik Sipil, 24(1), 17–25.

[9] Setyowati, A. (2017). Pedestrian friendly city: Kajian ruang pejalan kaki di kawasan pusat kota. Jurnal Teknik ITS, 6(2), A117–A122.

[10] Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2021). Crowd counting via scale-adaptive convolutional neural network. IEEE Transactions on Circuits and Systems for Video Technology, 30(1), 20–32.

[11] Ilyas, N., Mahmood, A., Mahmood, T., & Rho, S. (2020). A comprehensive survey of crowd counting and density estimation techniques. ACM Computing Surveys, 53(6), 1–37. https://doi.org/10.1145/3417982

[12] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. Cambridge, MA: MIT Press.

[13] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386

[14] Ashtari, S. (2022). Computer vision and its real-world applications. Journal of Artificial Intelligence Research, 78(3), 456–470.