Matrix: Jurnal Manajemen Teknologi dan Informatika
Volume 15 Issue 2 Year 2025 Pages 60-71
DOI: http://dx.doi.org/10.31940/matrix.v15i2.60-71
URL: http://ojs2.pnb.ac.id/index.php/matrix

Humanoid object detection moving in open space using YOLOv8

Muhammad Kahfi Yansah [1], Rohman Dijaya [2*], Hamzah Setiawan [3], Sumarno [4]

[1,2,3,4] *Informatics Department, Universitas Muhammadiyah Sidoarjo, Indonesia*

*Corresponding Author: *rohman.dijaya@umsida.ac.id*

Abstract: This study explores the application of the YOLOv8 algorithm in detecting humanoid objects in an open space environment, with a special focus on school areas such as parking lots. The main objective is to develop an intelligent system that can accurately identify students based on four uniform classifications: none, grey, batik, and department-specific uniforms. The system is designed to function effectively in real-time by analyzing image and video data. The research methodology begins with data acquisition using CCTV footage, followed by annotation and preprocessing using Roboflow. The dataset consists of 314 images with 1,649 labeled bounding boxes, which are then divided into training and validation sets. A yaml configuration file is created to interact with the YOLOv8 model. Training is performed using YOLOv8s variants, with experimental variations in image size, batch size, and epochs to optimize model performance. The evaluation results show that the model achieves a precision of 0.86, a recall of 0.92, and a mean Average Precision (mAP@0.50) of 0.93. Furthermore, visual testing confirms the system's ability to detect students with a total detection accuracy of 85%. Some minor errors were observed in distinguishing between visually similar classes, such as batik and department uniforms. These results demonstrate the robustness and reliability of YOLOv8 in dynamic real-world environments. This study concludes that YOLOv8 can be effectively applied to educational settings for surveillance or monitoring systems. Future research will focus on improving accuracy by expanding the dataset and incorporating more diverse categories of humanoid objects.

Keywords: deep learning, humanoid object, open space, YOLOv8

History Article: Submitted 12 May 2025 | Revised 13 June 2025 | Accepted 30 June 2025

How to Cite: M. K. Yansah, R. Dijaya, H. Setiawan, and S. Sumarno, “Humanoid Object Detection Appplication Moving in Open Space using YOLOv8: Object Detection”, *Matrix : Jurnal Manajemen Teknologi dan Informatika*, vol. 15, no. 2, pp. 60–71, 2025. doi.org/10.31940/matrix.v15i2.60-71.

Introduction

The advancement of information technology has grown rapidly along with the emergence of various innovations in the field of artificial intelligence (AI) [1]. One of the applications of AI that is currently being developed is the facial recognition and object detection system [2]. This system has an important role in various sectors, such as security, education, and attendance management [3]. Real-time facial recognition technology is considered an effective solution, because it is able to continuously capture and analyze facial data so that it can recognize a person's facial characteristics quickly and accurately [4].

The implementation of a face detection-based attendance system offers many advantages over conventional methods [5]. By utilizing a camera as input data, this system can automatically identify individuals without requiring physical contact, thus supporting the implementation of health protocols and operational efficiency [6]. In such a system, object detection technology plays an important role. Object detection is the process of identifying and determining the presence of an object in a digital image or video [7]. This method has been widely developed with various algorithms that have their own advantages in terms of speed and accuracy [8].

One of the object detection algorithms known to be effective and efficient is YOLO (You Only Look Once) [9]. YOLO is a deep learning-based algorithm that enables real-time object detection by dividing the image into grids and processing object predictions on each part of the grid simultaneously. This approach enables the processing of images or videos in one pass, making it very fast and suitable for real-time applications [10]. Aside from YOLO, several other object detection algorithms have also been widely used, such as Faster R-CNN, which offers high accuracy through a two-stage detection process SSD (Single Shot MultiBox Detector), known for its balance between speed and performance in real-time.

YOLOv8 is the latest iteration of the YOLO algorithm family that brings various improvements in terms of accuracy, efficiency, and detection speed. The YOLO (You Only Look Once) series has evolved significantly since its first release. YOLOv1 introduced real-time object detection by performing predictions in a single pass. YOLOv2 and YOLOv3 improved detection for smaller objects and added better backbone networks like Darknet-19 and Darknet-53. YOLOv4 integrated features like CSPDarknet and PANet, optimizing performance further for complex environments. YOLOv5, although unofficial, became widely adopted for its ease of use, flexibility, and support for deployment on various platforms. YOLOv6 and YOLOv7 continued to refine the trade-off between detection speed and accuracy with updated architectures and training techniques [11]. YOLOv8 uses a more optimal neural network architecture and is supported by more adaptive augmentation and training techniques, enabling object detection in various environmental conditions such as varying lighting, different object sizes, and complex backgrounds. Therefore, YOLOv8 is the main choice in development systems that require real-time response with high accuracy [12].

In various previous studies that have been conducted to create object detection using YOLO v8, such as studies Sholahuddin et al. [13] and Muntiari et al. [14], it is known that these studies only display one object. While in this study, the development is focused on detecting student objects with different classes. So this study can improve the system's ability to understand the details of the specified object.

This study aims to enable the performance of the YOLOv8 algorithm in detecting humanoid objects in school parking areas, both in image and video data. This study also tests the accuracy and algorithm of the algorithm in various dynamic environmental conditions. This study can contribute to the development of AI-based security and monitoring systems, especially in the context of education in the school environment. With the application of this technology, it is hoped that the monitoring system in schools will be more advanced, efficient, and adaptive to the needs of the times.

Methodology

In this section, the author will explain the research flow for detecting moving humanoid objects in open spaces using YOLO v8 starting with preparing the dataset, then the dataset is divided into two groups train and val, and a .yaml file is also created based on the data information, then the model training process is carried out followed by model evaluation and finally model testing. As shown in the Figure 1 below.

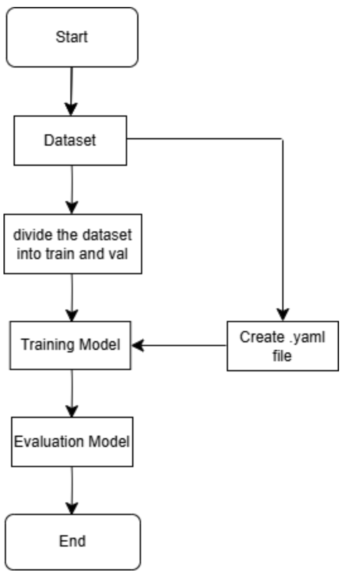

Figure 1. Flow diagram

Dataset

In Figure 1, the first stage begins with data collection taken from CCTV footage at SMK YPM 8 Sidoarjo, and roboflow as data processing. Roboflow is one of the best frameworks to help the labeling, pre-processing, and creating datasets, as well as inputting the dataset into the YOLOv8 algorithm [15].

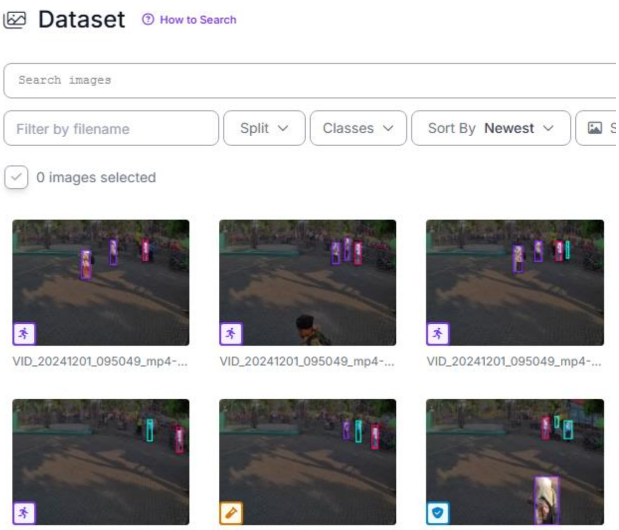

Figure 2. Dataset

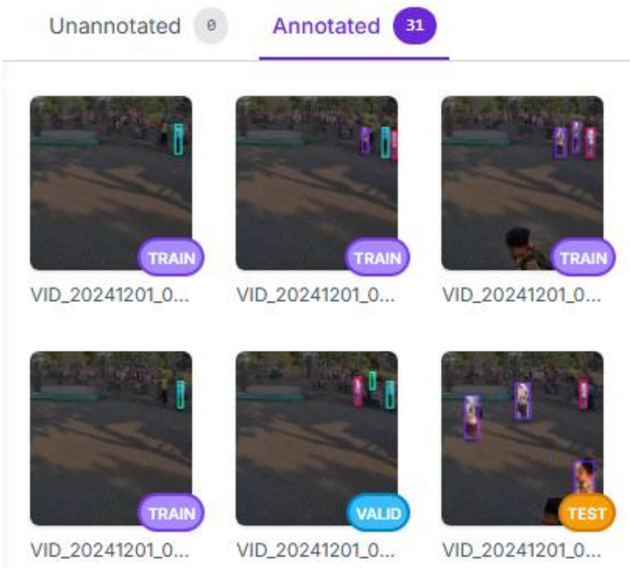

Figure 3. Annotation dataset/labeling

The dataset contains images as seen in Figures 2 and Figure 3 with a total of 314 data each. The dataset is used to contain four data classes, namely none, b_abu students, b_batik students, and b_jurusan students. Then from the annotation it is reprocessed in the form of a table containing data per-bounding box. So that the number of data for the none class is 791 data, b_abu students are 240 data, b_batik students are 460 data, and b_jurusan students are 158 data, then from the four classes it is added up to 1649 data from the initial number of 314, this number is obtained from all labeling or annotations per-body on each image.

The next stage is the division of train and val groups with a ratio of 783;37. the selection of this ratio is considered the best division in training data. Then create a yaml file as a bridge between the YOLO v8 model and the image input in the dataset. The yaml file contains brief information about the dataset, with a writing structure as in Figure 4 below.

File Edit Format View Help

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 4
names: ['none', 'siswa b_abu', 'siswa b_batik', 'siswa b_jurusan']

roboflow:
  workspace: kahfi-xj5tf
  project: kahfi_objek1
  version: 1
  license: CC BY 4.0
  url: https://universe.roboflow.com/kahfi-xj5tf/kahfi_objek1/dataset/1
```

Figure 4. .Yaml file writing structure

Humanoid Object Detection Architecture

In this study, the configuration settings for the img, batch, and epoch variables were determined to obtain optimal results on one previously selected YOLO v8 variant, namely yolov8s[16]. YOLOv8s is a lightweight version of the YOLOv8 family that balances speed and accuracy, making it suitable for real-time applications. The architecture of YOLOv8 features improvements such as a fully convolutional backbone, decoupled head for classification and regression, and a simplified detection head that enhances both inference speed and detection precision. YOLOv8 also utilizes anchor-free detection, enabling it to better adapt to various object shapes and scales. After going through the training process, the model was tested with image input. The result in each image is a prediction bounding box with class divisions labeled as: "none", "siswa b_abu", "siswa b_batik", and "siswa b_jurusan", along with their predicted confidence values. Yolov8 architecture can be seen in Figure 5.

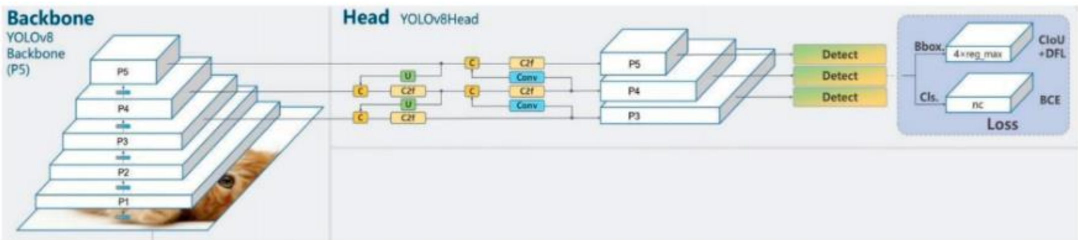

Figure 5. Yolov8 architecture

Humanoid Object Detection Using YOLOv8

In this study, the author uses the YOLO v8 variant, namely yolov8s. The writing structure of the training model command involves several things as follows.

Img

Img interpreted as a regulator of the input image or image size, the larger the size set, the more detailed the object to be detected, but requires more power. The smaller the image size, the faster the model training process which then lightens the computing performance. Several studies have determined the ideal image input size without having to lose much information, namely 416 X 416. in this study, it was determined using the img configuration at values 416 and 640. The selection of 640 was used as a comparison from 416.

Batch

Batch namely the number of images in one group. If the amount of data is 2738 images, with a batch size of 32, then the number of batches needed is 2738: 32 = 86 batches. The more batches used, the greater the memory consumption. In this study, four types of batches will be used starting from the numbers 8, 16, 32, and 64. This is done to determine the differences produced from the multiples of the previous batch.

Epoch

Epoch which is the number of rounds used during the training process. In this study, the epochs used are at 100, these numbers were chosen to minimize training time, and avoid epoch values that are too high because they do not always produce higher accuracy so that appropriate epoch settings are needed to obtain optimal results[17].

Data

Data which is a yaml file containing data collection information.

Weight

Weight refers to the model variant used, namely yolov8s which is available in a set of YOLO v8.

The YOLO v8 training model takes from Ultralytic Github data which then during the training process produces several values, namely the F1 value which is the average value between precision and recall. These results are obtained from the following mathematical statement.

F1-score is an evaluation metric that combines precision and recall to provide a more comprehensive perspective on the performance of a classification model. The goal is to balance precision and recall, especially when the distribution of positive and negative classes is imbalanced.

$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision +\ Recall} \tag{1}$$

Precision value is a measure that shows how many positive category data are correctly classified divided by the total data classified as positive. The equation for calculating precision value is as follows:

$$Precision = \frac{TP}{TP\ x\ FP} \tag{2}$$

The recall value shows the percentage of positive category data that is correctly classified by the system. The equation for calculating the recall value is as follows:

$$Recall = \frac{TP}{TP\ x\ FN} \tag{3}$$

Where TP is *True Positive*, shows the number of positive data that are correctly classified by the system, FP is *False Positive*, shows the number of positive data but is classified incorrectly by the system, FN is *False Negative*, shows the number of negative data but is classified incorrectly by the system.

mAP (mean Average Precision) is an evaluation metric used to measure the performance of object detection algorithms, including YOLO (You Only Look Once). mAP measures how well a model detects and classifies objects in an image. AP (Average Precision) is a measure of detection performance per object class. For example, if there are 4 types of objects (car, human, bicycle, motorcycle), then:

$$mAP = \frac{APmobile + APhuman + APbicycle + APmotorcycle}{4} \tag{4}$$

Where mAP *is average of AP*. AP is *Average Precison*, the average value of precision at various recall levels (usually from 0 to 1).

Accuracy is a measure of how well a system or model performs prediction or classification. In the context of object recognition, accuracy indicates the percentage of correct detections compared to the total number of detections made.

$$Accuracy = \frac{Correct\ Detections}{Number\ of\ Detections}\ x\ 100\% \quad (5)$$

Results and Discussions

Training Model

Based on the results of the model training process test that has been carried out for 100 epochs, the following 5 highest epochs of the results obtained can be presented in Table 1 below.

Table 1. YOLOv8 model test results

Epoch	Precision	Recall	mAP
17	0.86	0.92	0.93
43	0.83	0.88	0.91
49	0.85	0.85	0.89
41	0.85	0.89	0.91
24	0.86	0.89	0.92

Based on Table 1 above, the results at epoch 17 have the highest value with a precision of 0.86, a recall of 0.92, and mAP@0.50 reaching 0.93. These values indicate that the model is able to detect and localize objects with a high level of accuracy at various IoU thresholds.

Epoch 43 has a good balance of recall and mAP values, while epoch 49 shows high precision even though recall decreases slightly. Epochs 41 and 24 also show strong performance, with competitive recall and mAP values. Epoch 24 records the highest precision (0.86) of all the top five, making it the right choice for cases that require high sensitivity to false positives.

Precision, recall, and mAP values are considered good when they exceed 0.80. A precision of 0.86, as seen in epoch 24, indicates that 86% of the model's positive predictions are correct, which reflects a low rate of false positives and is considered strong. Whereas, a recall above 0.80 indicates that the model successfully captures most of the actual objects, minimizing false negatives. Meanwhile, mAP values closer to 1.0 indicate better overall accuracy across all object classes and thresholds. Therefore, the performance observed in epochs 24, 41, 43, and 49 can be categorized as high-quality based on these thresholds.

Figure 6 shows the results of the Confusion Matrix, providing a visual representation of how well the YOLOv8 model can detect Humanoid Objects. Analysis of the Confusion Matrix results shows that the number of correct detections is close to 1 for almost every object class. There are also exceptions in each class where there are 0.01 to 0.10 detection errors.

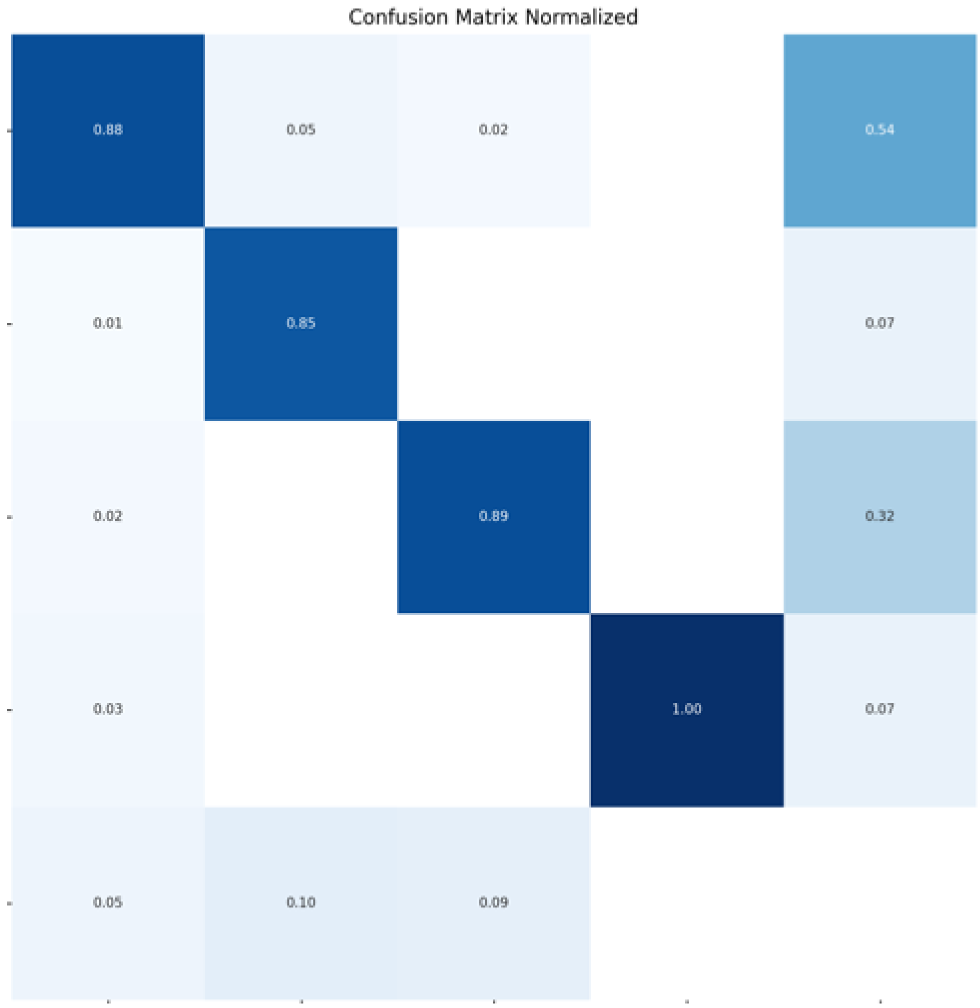

Figure 6. Confusion matrix

Figure 7 shows some graphs of the YOLOv8 training results as follows.

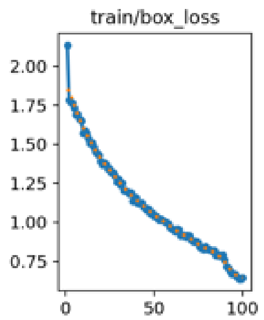

Figure 7. (a) Train/box_loss

Figure 7 (a) shows the decrease in bounding box loss during the training process. A lower loss value indicates an increase in the accuracy of the model in predicting the spatial position of objects.

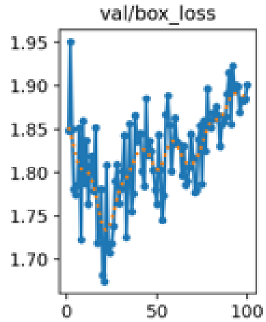

Figure 7. (b) val/box_loss

Figure 7 (b) shows the box loss value on the validation data. Fluctuations in this graph indicate model inconsistency when tested on untrained data, but are still within acceptable limits.

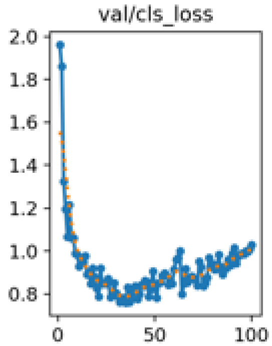

Figure 7. (c) val/cls_loss

Figure 7 (c) shows the classification loss on the validation data. The U-shaped curve indicates that there is an adjustment of the model to the existing object classes until it reaches the minimum loss value.

Figure 8 shows three graphs representing the performance of the YOLOv8 model in detecting humanoid objects. The first graph is the F1-Score curve of 0.88 at a Confidence Level of 0.368. The second graph is the Precision curve against the Confidence Level of the model, where all humanoid object classes achieve a Precision value of 1 at a Confidence Level of 0.812. While the third graph is the Recall curve against the Confidence Level of the model, with all humanoid object classes achieving a Recall value of 0.98 at a Convidence Level of 0.000.

These Confidence Level values reflect the model's prediction certainty thresholds. A Confidence Level of 0.368 indicates the optimal balance between precision and recall, as shown by the highest F1-Score. At a higher threshold (0.812), the model becomes more selective, producing highly accurate predictions (Precision = 1) but potentially missing some objects (lower recall). Conversely, at a Confidence Level of 0.000, the model accepts all predictions, resulting in very high object detection coverage (Recall = 0.98), but possibly including more false positives. Therefore, the choice of confidence level impacts the trade-off between accuracy and completeness of detection.

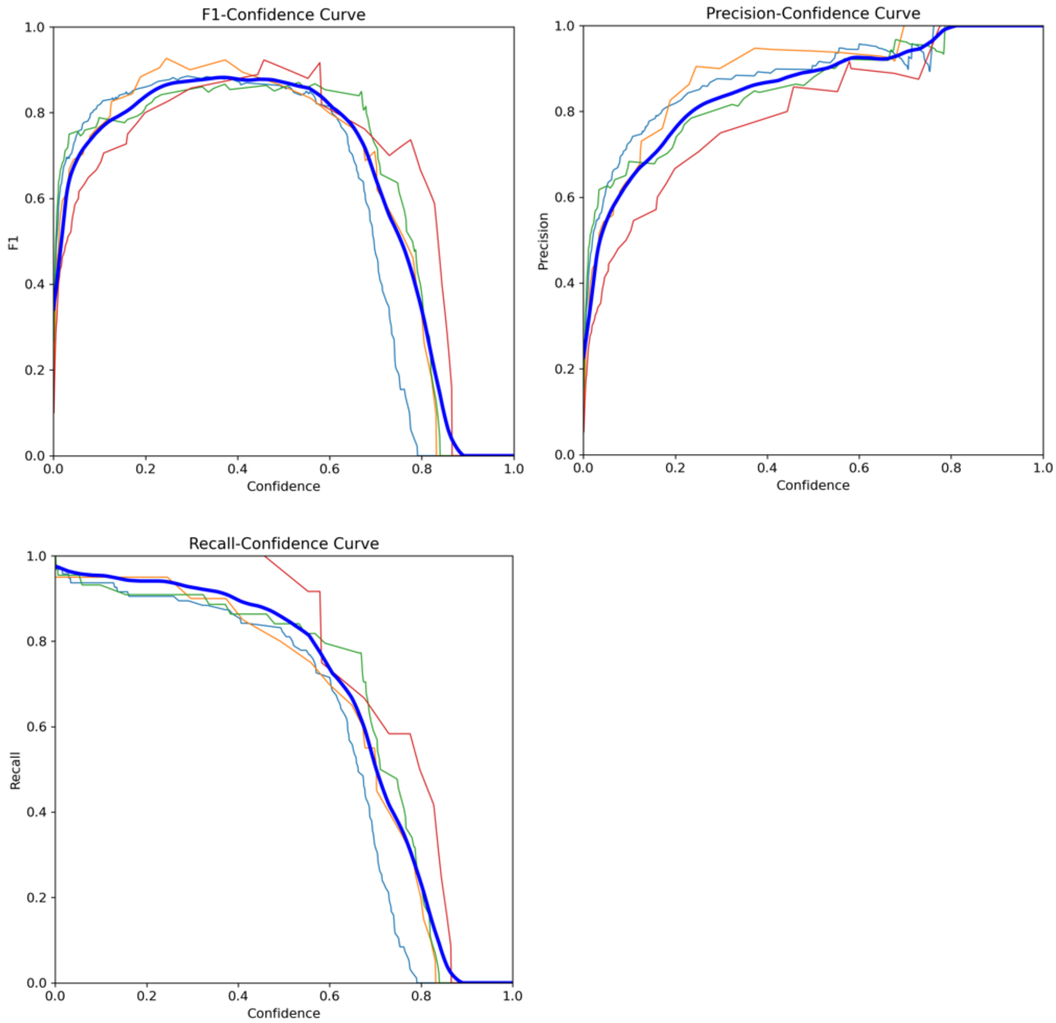

Figure 8. F1-Score Curve, Precision, Recall

Evaluation Model

To validate the accuracy off the research model, an evaluation of the model results in all model training scenarios is required. The evaluation is carried out by testing the image results that have certain conditions to determine the level of correctness of humanoid object detection in the model scenario. The following are the image results needed for the model evaluation process.

Figure 9. Model training results for model evaluation

In Figure 9, the model shows good and stable visual prediction performance according to the metric evaluation results. Important objects *(siswa)* can be recognized with high precision, but some cases of label overlap and low confidence predictions indicate that further fine-tuning can still improve more specific classification accuracy, especially for visually similar classes, the following are the results of a comparative evaluation of the level of correctness and error of each model in detecting objects presented in Table 2 below.

Table 2. Model accuracy evaluation results

Category	Number of detections	Correct detections	False detections	Accuracy
none	29	25	4	86.2%
siswa b_abu	7	6	1	85.7%
siswa b_batik	13	10	3	76.9%
siswa b_jurusan	11	10	1	90.9%

Based on the model accuracy evaluation results, the detection accuracy level reaches 85%, thus showing very good performance in detecting humanoid objects in real conditions.

Conclusion

The conclusion of this study shows that the implementation of the application with the YOLOv8 algorithm in detecting humanoid objects in open spaces, especially in identifying students in uniform, has succeeded in achieving the stated objectives. The best accuracy level reaches a precision value of 0.86, a recall value off 0.92 and a mAP @ 0.93 value during the training process by utilizing 791 images from the dataset. The evaluation results show a good level of accuracy reaching an accuracy value of 85%. These findings contribute to the development of information technology, especially in the field of intelligent surveillance systems, by validating the practical implementation of advanced AI algorithms in dynamic, real-world environments.
Future research will continue to develop the YOLO algorithm in its best version by incorporating a more representative training dataset, which will include a variety of humanoid objects. This will improve the algorithm's ability to accurately recognize and classify object features.

Acknowledgments

Intended to pay gratitude to sponsors, We would like to express our deepest gratitude to Universitas Muhamadiyah Sidoarjo for their support of this research. Their commitment to fostering innovative academic research has been instrumental in allowing us to thoroughly explore the intersection of technology and financial behavior. Our sincere gratitude also goes to the highly dedicated research team who have diligently carried out their responsibilities throughout this research. Their expertise and hard work have been the foundation for the success of this research, and their contributions cannot be overstated. We also thank P3M Politeknik Negeri Bali for providing a high-quality platform for our journal publication. The opportunity to share our findings with the academic community through such a respected publication is greatly appreciated. Together, the support from these institutions has not only made this research possible but also contributed to the advancement of knowledge in the field of financial technology and risk assessment. We are grateful for their continued support and collaboration. Major contributors to the research, such as funders, sources, and other important parties, should be named. However, authors must obtain permission from these individuals or institutions before including them in the acknowledgments. It is not necessary to mention the editor.

References

[1] D. Pongoh, M. Kasenda, S. Kasenda, G. Tondatuon, M. Taroreh, and E. Saleh, "Penerapan kecerdasan buatan (artificial intelligence/ai) dalam dunia pendidikan," *Jurnal Ilmu Komputer dan Sistem Informasi.,* Vol.9, No.4, pp. 164-168, 2023.

[2] A. P. Saputri, A. Taqwa, and S. Soim, "Analisis deteksi objek citra digital menggunakan algoritma Yolo dan CNN dengan arsitektur REPVGG pada sistem pendeteksian dan pengenalan ekspresi wajah," *Syntax Literate: Jurnal Ilmiah Indonesia*, vol. 7, no. 9, Sep. 2022.

[3] F. A. Saputra and J. C. Chandra, "Prototipe Sistem Keamanan Ruang Server Otomatis Menggunakan ESP32CAM dan Algoritma You Only Look Once (YOLO)," *Jurnal TICOM: Technology of Information and Communication*, vol. 11, no. 1, 2022.

[4] N. J. Hayati, D. Singasatia, M. R. Muttaqin, T. Informatika, S. Tinggi, and T. Wastukancana, "Object tracking menggunakan algoritma You Only Look Once (YOLO)v8 untuk menghitung kendaraan," *KOMPUTA : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, 2023, [Online]. Available: https://universe.roboflow.com/

[5] L. Susanti, N. K. Daulay, and B. Intan, "Sistem absensi mahasiswa berbasis pengenalan wajah menggunakan algoritma YOLOv5," *JURIKOM (Jurnal Riset Komputer)*, vol. 10, no. 2, p. 640, Apr. 2023, doi: 10.30865/jurikom.v10i2.6032.

[6] M. F. Nuryasin, C. Machbub, and L. Yulianti, "Kombinasi deteksi objek, pengenalan wajah dan perilaku anomali menggunakan state machine untuk kamera pengawas," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 11, no. 1, p. 86, Jan. 2023, doi: 10.26760/elkomika.v11i1.86.

[7] A. Setiadi, E. Utami, and D. Ariatmanto, "Analisa kemampuan algoritma YOLOv8 dalam deteksi objek manusia dengan metode modifikasi arsitektur," *Jurnal Sains Komputer dan Informatika,* Vol.7, No. 2, 2023.

[8] R. G. Guntara, "Pemanfaatan Google Colab untuk aplikasi pendeteksian masker wajah menggunakan algoritma Deep Learning YOLOv7," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, Feb. 2023, doi: 10.47233/jteksis.v5i1.750.

[9] R. Muwardi, I. P. Nugroho, K. S. Salamah, M. Yunita, R. Rahmatullah, and G. J. Chung, "Optimization of YOLOv4-Tiny Algorithm for vehicle detection and vehicle count detection embedded system," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 10, no. 3, pp. 639–648, Nov. 2024, doi: 10.26555/jiteki.v10i3.29693.

[10] I. H. A. Amin, A. Aprilino, "Implementasi algoritma YOLO dan TESSERACT OCR pada sistem deteksi plat nomor otomatis," *Jurnal TEKNOINFO*, vol. 16, 2022.

[11] R. S. I. Sihombing, W. A. Harahap, and W. K. Rahman, "Implementasi YOLO V8 untuk mendeteksi mata uang rupiah emisi tahun 2022 ber-output audio," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, Aug. 2024.

[12] W. A. Rahmat, S. M. Ladjamuddin, and D. T. Awaludin, "Perbandingan algoritma Decision Tree, Random Forest dan naive bayes pada prediksi penilaian kepuasan penumpang maskapai pesawat menggunakan dataset KAGGLE," *Jurnal Rekayasa Informasi*, vol. 12, no. 2, 2023, [Online]. Available: www.kaggle.com,

[13] M. R. Sholahuddin, M. Harika, I. Awaludin, Y. C. Dewi, F. D. Fauzan, B. P. Sudimulya, and V. P. Widarta, "Optimizing YOLOv8 for Real-Time CCTV Surveillance: A Trade-off Between Speed and Accuracy," *Jurnal Online Informatika*, vol. 8, no. 2, pp. 261–270, Dec. 2023, doi: 10.15575/join.v8i2.1196.
[14] N. R. Muntiari, I. C. Nisa, A. Sriekaningih, A. Y. A. Prasetyo, and M. Yusril, "Penerapan algoritma YOLOv8 dalam indentifikasi wajah secara real-time menggunakan CCTV untuk presensi siswa," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 4, no. 3, pp. 1155–1165, Nov. 2024, doi: 10.51454/decode.v4i3.847.
[15] R. Theofilus and R. Kurniawan, "Deteksi Sampah di Permukaan Sungai menggunakan Convolutional Neural Network dengan Algoritma YOLOv8 Studi Kasus: Sungai Ciliwung (Detection of Floating Wastes on River Surface using Convolutional Neural Network with YOLOv8 Algorithm (Case Study: Ciliwung River))," 2024.
[16] R. Irsabmalfi and R. Rahmadewi, "Deteksi anatomi gigi menggunakan arsitektur YOLOV8 (You Only Look Once) untuk pengenalan jenis-jenis gigi," 2025.
[17] E. Astiadewi, A. Rinaldi Dikananda, and D. Rohman, "Algoritma YOLOv8 untuk meningkatkan analisa gambar dalam mendeteksi jerawat," *Jurnal Informatika Teknologi dan Sains*, vol. 7, no. 1, pp. 346–353, 2025.