Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol. 8, No. 3, November 2025, pp. 728 – 736
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Comparative Study: Performance Comparison of You Only Look Once and Convolutional Neural Networks Algorithms in Human Object Detection

[1*]Dewi Permata Sari, [2]M. Akbar Tri Ramadhani, [3]Abdurrahman
[1,2,3]Departement of Electrical Engineering, Politeknik Negeri Sriwijaya, Indonesia
Email: [1]dewi_permatasari@polsri.ac.id, [2]akbar.tramdhan18@gmail.com, [3]abdurrahman@yahoo.com

Article Info
Article history:
Received Jul 12th, 2025
Revised Oct 09th, 2025
Accepted Nov 27th, 2025

Keyword:
Computer Vision
Human Detection
MobileNetSSD
Real-Time Detection
YOLOv8n

ABSTRACT
The evolution of object identification technologies, particularly for person detection applications, has accelerated significantly due to the merger of deep learning, artificial intelligence, and computer vision. This study aims to evaluate the efficacy of two object detection algorithms, YOLOv8n and MobileNetSSD, in identifying human objects in digital images. A dataset of 12,334 human-labeled photos from the Roboflow platform was utilized to train the YOLOv8n model, while performance results for the CNN MobileNetSSD model were acquired from a prior article. The precision, recall, and F1-score of each model were examined. Experimental results reveal that YOLOv8n attains 94% precision, 92% recall, and a 92.9% F1-score, representing a considerable enhancement over MobileNetSSD. Conversely, MobileNetSSD got an F1-score of 85.2%, with a precision of 86.5% and a recall of 84.1%. The findings indicate that CNN MobileNetSSD is more suitable for non-time-sensitive or resource-limited scenarios; however, YOLOv8n is preferable for real-time human identification tasks due to its higher accuracy and faster inference. This comparative analysis is crucial for differentiating object detection models tailored to specific application needs.

Corresponding Author:
Dewi Permata Sari,
Department of Electrical Engineering, Polytechnic State of Sriwijaya,
Jl. Srijaya Negara, Bukit Lama, Ilir Bar. I, Palembang City, South Sumatra, Indonesia - 30128
Email: dewi_permatasari@polsri.ac.id
DOI: http://dx.doi.org/10.24014/ijaidm.v8i3.37676

1. INTRODUCTION

In recent years, the demand for accurate and real-time human detection has increased significantly, particularly in applications such as security systems, smart environments, access control, and public monitoring[1]. Advancements in deep learning and artificial intelligence (AI) have considerably driven the growth of computer vision, notably in object detection. However, conventional detection methods still face several challenges, including varying lighting conditions, complex backgrounds, occlusion, and limited computational efficiency, which often result in unstable detection accuracy and slow processing performance. Therefore, this study aims to analyze and compare the performance of two widely used deep learning-based detection approaches You Only Look Once (YOLO) and Convolutional Neural Networks (CNN) to determine which method offers superior accuracy, speed, and robustness in human detection tasks.

This comparison is crucial because recent YOLO versions, such as YOLOv4, YOLOv5, YOLOv7, and YOLOv8, have demonstrated significant improvements in detection speed and real-time capability[2][3][4][5]. The YOLO algorithm was designed to address these challenges, employing a single-stage object detection approach that prioritizes speed and efficiency. A deep learning methodology, termed YOLO, is capable of identifying various objects, including persons [6][7]. The YOLO approach was created for real-time object recognition. The detection system utilizes a classifier or localizer for object detection tasks[8]. Utilizing a global image detection technology, YOLO eliminates erroneous background object identifications and successfully obtains contextual information[9].

Meanwhile, modern CNN-based detectors such as MobileNet-SSD, which are designed to be lightweight and efficient for embedded applications, generally offer good feature extraction capabilities but still require careful optimization to achieve real-time performance[10]. CNNs are a prominent deep learning technology, particularly useful for object detection and picture categorization applications. These algorithms are among the most used for processing and assessing visual data[11]. CNN algorithms have showed great proficiency in several practical applications, including facial recognition, medical picture interpretation, and autonomous cars[12]. Although traditional CNN algorithms display remarkable accuracy, they are occasionally computationally costly, thereby reducing their efficacy in real-time applications.

Prior research demonstrates that CNN and YOLO can indeed be compared because both are deep-learning-based object detection approaches, but they differ significantly in architecture and performance. Khairunisa et al.[13] and Dewanto et al.[14] show that CNN models generally achieve higher accuracy, yet require longer processing time, making them less suitable for real-time applications. This finding aligns with the study by Lee and Kim[15], which reports that YOLO provides faster detection speed due to its single-stage architecture.

Furthermore, Su[16] compares YOLO with other CNN-based detectors such as Faster R-CNN and SSD, reinforcing that YOLO consistently outperforms in speed while maintaining competitive accuracy. Collectively, these studies confirm that comparing CNN and YOLO is both valid and meaningful, as they present a clear performance trade-off: CNN prioritizes accuracy, whereas YOLO emphasizes real-time processing. Therefore, the choice between the two depends on whether an application requires higher accuracy or faster detection.

This research compares the CNN and YOLO algorithms for human object detection, clearly highlighting the strengths and limitations of each method. The comparison aims to help determine the most suitable detection model for specific application needs. By evaluating both algorithms using the same dataset and testing conditions, this study provides more reliable insights for selecting the algorithm that best supports real-time human detection tasks.

2. RESEARCH METHOD

This study compares the performance of CNN MobileNet-SSD with YOLOv8 in person detection using F1-score, precision, and recall measurements. The study will move through numerous stages to reach this purpose, as described below:

2.1. Literature Study

The initial phase of this endeavor is a literature assessment (see Table 1), which involves the identification and examination of scholarly articles on human detection using CNN MobileNet-SSD-based and YOLOv8-based methodologies.

2.2. Research Design

This study employs a quantitative comparative methodology. The author of this paper developed and trained a YOLO-based human detection algorithm utilizing a publicly accessible dataset. The performance data of a CNN-based model (MobileNet-SSD) were acquired from a peer-reviewed work by Abdul Aziz et al. (2024) entitled "Human Object Detection Testing Using Jetson Nano with the SSD MobileNetV2 Model" for comparative analysis instead of through retraining [18].

The referenced paper conducted analogous tests under the same conditions. This approach focuses the training efforts on the YOLO architecture while enabling the evaluation of the YOLO model's efficacy in comparison to a recognized CNN model. The evaluation emphasizes critical performance metrics such as F1-score, recall, and precision to illustrate the comparative merits and drawbacks of each technique.

2.3. You Only Look Once (YOLO)

YOLOv8, the most recent iteration of the YOLO family, is renowned for its exceptional speed and precision in real-time object detection in photographs[21]Object recognition in real time was accomplished through the development of an algorithm known as YOLO [22]. YOLO outperforms competing algorithms due to its capacity to efficiently process data in real time while maintaining a commendably high level of accuracy[23]. To utilize YOLO, divide an image into an S×S grid, with each cell anticipating m boundary boxes. Bounding boxes contain offset values, confidence scores, and class probabilities. If the picture class probabilities are above a threshold, bounding boxes are used to identify objects[24]. Figure 1 shows the architecture of YOLO.

Table 1. Literature Study

Name & Years	Title	Method	Result
Rachmat Muward, et al[17]	Human Object Detection for Real-Time Camera using Mobilenet-SSD	MobileNetV2-SSD for real-time human detection on a NanoPi M4V2 device using Python and OpenCV.	Achieved mAP of 0.9705, processing time of 67 ms/frame, 100% stable detection on a 6-second video, with low CPU/RAM usage and efficient system performance.
Muhammad Abdul Aziz, et al[18]	Pengujian Deteksi Objek Manusia Menggunakan Jetson Nano Dengan Model Ssd Mobilenetv2	SSD MobileNetV2 implemented on NVIDIA Jetson Nano to evaluate real-time human detection under varying lighting and crowd density, tested at 5W and 10W power modes using frame-rate measurement and confusion matrix analysis.	Achieved 24 FPS (10W) and 17 FPS (5W). Accuracy varies by condition: bright (86%), dim (53%), crowded (85%), and non-crowded (98%), showing strong performance in optimal lighting but reduced accuracy in low-light or crowded scenarios.
Tahreer Abdul Ridha Shyaa, et al[19]	Enhancing real human detection and people counting using YOLOv8	YOLOv8 (small model) with ROI-based entry/exit tracking for real human detection and people counting in images and videos, using COCO dataset weights and threshold-based bounding box filtering.	Achieved accurate human detection with 44.9 mAP on COCO val2017 and 100% correct people counting in real test video. System successfully distinguished between entering and exiting persons and triggered warnings when crowd thresholds were reached.
Joel Hamim Sim, et al[20]	Evaluating YOLOv5 and YOLOv8: Advancements in Human Detection	Comparative evaluation of YOLOv5 and YOLOv8 for real-time human detection using the SEMMA methodology. Dataset prepared and labeled via Roboflow, trained for 100 epochs on Google Colab. Performance assessed using precision, recall, and F1-score derived from confusion matrices.	YOLOv8 outperforms YOLOv5 with higher precision (75.85%) and F1-score (79.21%). YOLOv5 shows slightly higher recall (83%). Overall, YOLOv8 delivers more accurate and efficient human detection performance.

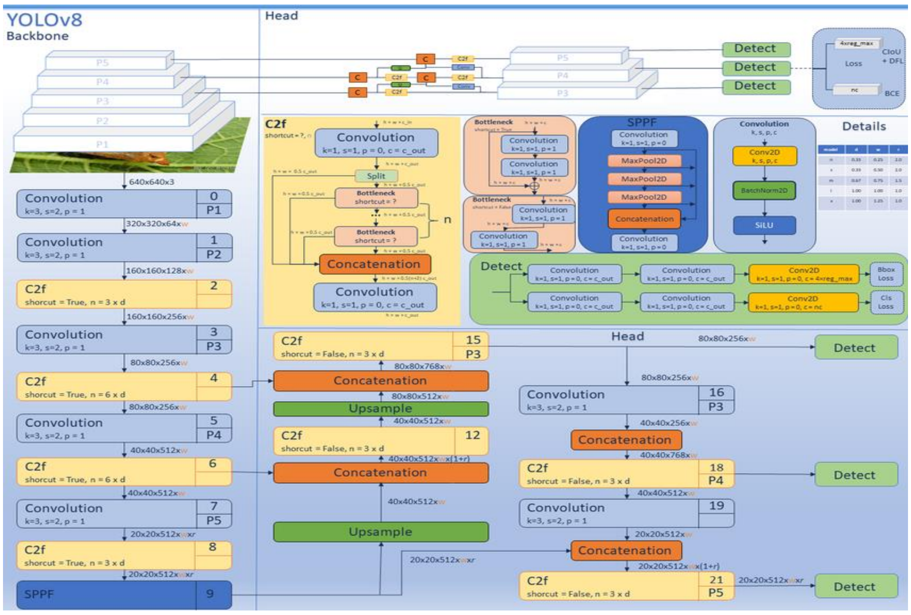


Figure 1. Architecture of YOLO [25]

2.4. Convolutional Neural Network

CNN are grid-based neural networks designed to interpret data. For instance, a 1D temporal grid can display time-series data, and a 2D pixel grid can display image data[26]. TensorFlow is a framework that was developed by Google. It has extensive support for the CNN algorithm, which has made it one of the most widely used methods for object detection[27]. The CNN is an extension of the Multilayer Perceptron (MLP) that was made to look at two-dimensional data[28]. CNNs consist of numerous crucial layers, including the activation function, pooling layer, fully connected layer, and convolutional layer, that work together to process features and extract information from incoming data[29]. CNNs have evolved into several architectures to address specific challenges in visual data processing; they include MobileNet, ResNet, and LeNet[30]. Arsitecture of CNN MobileNet can be seen in Figure 2.

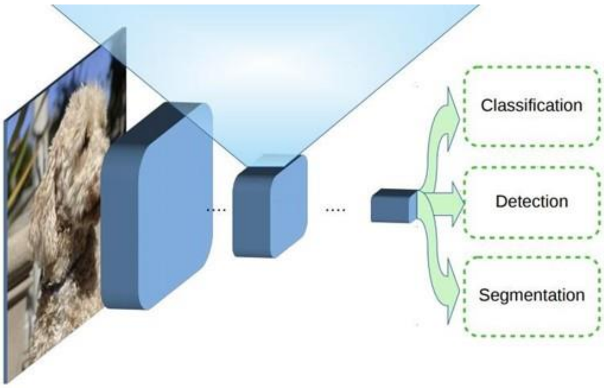


Figure 2. Architecture of CNN MobileNet [31]

2.5. Dataset Collecting and Preprocessing

In Table 2 are some of the datasets used for this research.

Table 2. Dataset used

Example of Dataset	
	
	

The dataset for the study came from Roboflow, an online service that provides ready-to-use datasets with annotations for many computer vision tasks, mainly object recognition. The chosen dataset is mostly made up of pictures of people, complete with automatically generated bounding boxes and class labels. These comments make the data processing stage go much faster by getting rid of the need for human labeling, which can be time-consuming and lead to mistakes.

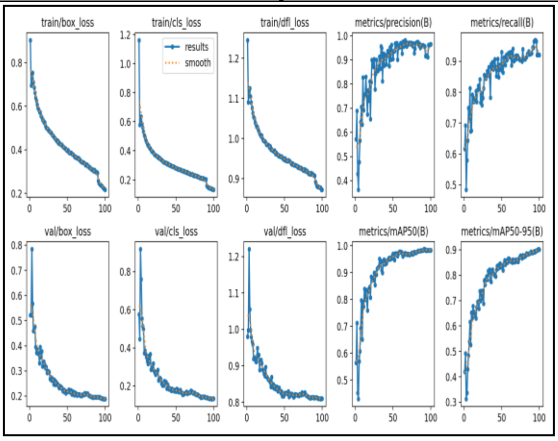
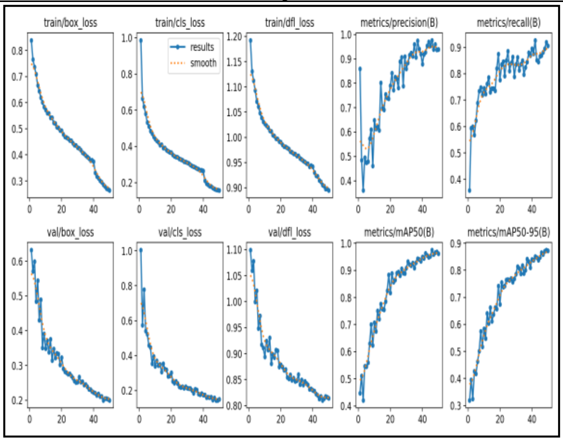
A total of 12,334 shots were used in this study. To divide the dataset, a ratio of 80:20 was used. Of the total 10,867 pictures, 2,467 were used for training, and the rest were split evenly between testing and validation, making 1,234 images for testing and 1,233 images for validation. In this part, we checked to see if the model could be taught and tested using data that had never been seen before.

In each picture in the dataset, there are different lighting conditions, backgrounds with lots of details, different body positions, and different camera angles. It is very helpful when this variety is used on new, unknown data because it makes the model better at generalization.

2.6. Training Data

Specifically, the object detection model was trained using the real-time optimized lightweight YOLO nano v8 architecture, which is a part of the YOLO 8. As part of the training process, the Google Colab platform was utilized along with an A100 GPU and the Ultralytics library version 8.3.150. The model was trained in two sessions, one lasting 100 epochs and the other 50. The input picture size was 640×640 pixels (imgsz=640), the batch size was 32, and there was an early stopping set up with a patience of 15 epochs (patience=15). Training plots were enabled (plots=True) to provide a more comprehensive view of the learning process. Training result can be seen in Table 3.

Table 3. Training result

100 Epoch	50 Epoch
	

After 50 epochs of training, the results are remarkably consistent with those after 100 epochs. In both the training and validation sessions, the loss values, specifically, box loss, classification loss (cls_loss), and distribution focal loss (dfl_loss), consistently decreased. This suggests that the model converged efficiently in both situations. The assessment measures, on the other hand, reveal noticeable disparities in performance. Over the course of 50 epochs of training, the precision reached approximately 0.90, and after 100 epochs, it rose to around 0.95. After 50 epochs, the recall approached 0.90, and after 100 epochs, it reached approximately 0.93, continuing a similar trend. And in the 100-epoch training, the mAP50 was higher than 0.92, whereas in the 50-epoch training, it was marginally higher than 0.88. The mAP50-95 metric was over 0.85 after 100 epochs and over 0.83 after 50 epochs. This improvement indicates that for numerous Intersection over Union (IoU) criteria, the accuracy of predictions improved as the training period was lengthened.

Overall, the results from the 100-epoch training were much superior than those from the 50-epoch model, notably in terms of detection accuracy and precision. This suggests that increasing the number of training epochs can improve model performance, although the rate of improvement tends to decrease as training progresses. Along with these two measurements, the F1 Score was developed to analyze the trade-off between recall and precision.

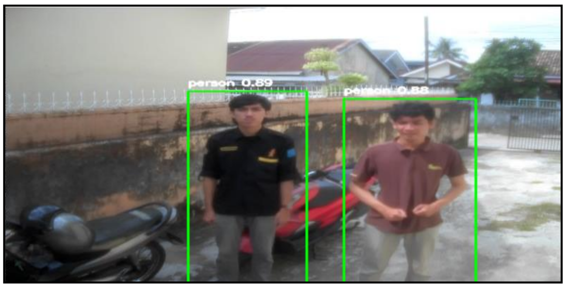
During the fifty-epoch training period, both precision and recall achieved a score of 0.90, resulting in an F1 Score of 0.90. With an accuracy of 0.95 and a recall of 0.93, the F1 Score increased to 0.94 over the course of the training cycle that lasted for 100 epochs. The fact that the F1 Score has increased indicates that the model's ability to reliably classify objects while simultaneously minimizing the number of false positives has improved as a result of the longer training period. Additionally, the individual detection metrics have also improved. As a consequence, the detection performance obtained from the 100-epoch training was more consistent and thorough.

2.7. Model Evaluation

Evaluation of the model was carried out using test and validation datasets, which were not utilized during training. The evaluation findings demonstrated that the model successfully identified human presence in various settings on 1,234 validation images and 1,233 test images. We utilized training and validation metric graphs to display the results, which were based on training results across 100 epochs. Consistently decreasing loss values throughout training and validation point to data learning by the model. In addition, the model did not exhibit any overfitting issues, as there was no noticeable difference in training and validation loss.

In Table 4 is the model was tested under two different lighting conditions: adequate illumination and low light. In well-lit settings, the model accurately identified human items and displayed their characteristics, such as shape, texture, and contour, with great clarity. Thanks to this picture-perfect illumination, the model performed admirably. In contrast, the model's detection accuracy dropped when the lighting was poor. Because key object properties were less visible due to the decreasing illumination, some items were either missed or mistakenly detected. This research proves that illumination is a key factor influencing item detection accuracy.

Table 4. Test results under two lighting conditions

Adequate illumination condition	Low light condition
	
	
	

3. DISCUSSION AND ANALYSIS

Within this study the author developed and trained the YOLOv8n model utilizing a dataset comprising 12,334 images, which were divided into training, validation, and testing sets sourced from Roboflow. Conversely, the author did not directly implement the CNN model; instead, they referenced its performance metrics from a journal article titled "Human Object Detection Testing Using Jetson Nano with the SSD MobileNetV2 Model by Abdul Aziz M et al. (2024)"[18]. The evaluation results of the CNN model provided a benchmark for assessing the performance of the YOLOv8n model in this study. Experiments were conducted utilizing the identical dataset from Roboflow to evaluate and compare the efficacy of the CNN (MobileNetSSD, as referenced in the journal) and the YOLOv8n algorithm in detecting human objects.

Experiments were conducted to evaluate and compare the efficacy of the CNN-based MobileNet-SSD model with the YOLOv8n algorithm in detecting human objects. 12,334 images from a publicly available dataset provided by Roboflow were utilized to train the YOLOv8n model. The photos were divided using an 80:20 ratio: 1,234 for validation, 1,233 for testing, and 9,867 for training. This division facilitated proper judgment on unseen data and ensured optimal training. Conversely, a previous work presented the performance results for the MobileNet-SSD model[18], utilizing a different dataset for training the model. Instead, in analogous experimental conditions, it serves as a benchmark for comparison.

This experimental benchmarking enables a direct performance comparison between the YOLOv8n model and MobileNet-SSD, ensuring an objective assessment that aligns with previous research findings. The evaluation metrics obtained for each model are presented in Table 5. The evaluation results indicate that YOLOv8, particularly the Nano version (YOLOv8n), trained for 100 epochs, achieved an F1-score of 0.94, with a precision of 0.95 and a recall of 0.93. These results can be attributed to YOLOv8's single-stage, end-to-end detection mechanism, which performs localization and classification simultaneously, thereby eliminating intermediate region proposal steps and reducing localization errors.

This performance is significantly superior to that of MobileNet-SSD, which attained an F1-score of only 0.85. The results indicate that, even under challenging conditions such as low light, YOLOv8 consistently demonstrates superior object categorization and localization capabilities. The end-to-end, single-stage detection architecture of YOLOv8 is a key factor in its superiority. Unlike MobileNet-SSD, which employs a two-step approach involving region proposals followed by classification, YOLOv8 performs object localization and classification simultaneously in a single processing step.

The YOLOv8 model was evaluated under two different lighting conditions: low-light and adequate (bright) lighting. Precision values in both settings surpassed 0.90, demonstrating consistently robust outcomes. This indicates that under diverse visual conditions, including light noise, harsh shadows, and compromised image quality (e.g., blur or overexposure), YOLOv8 demonstrates robust generalization capabilities. In contrast, the findings of Abdul Aziz et al. (2024) show a notable decline in MobileNet-SSD performance under low-light conditions, reinforcing that MobileNet-SSD is more affected by visual noise compared to YOLOv8.

Abdul Aziz's study indicates a notable decline in MobileNet-SSD performance under low light conditions, with the F1-score decreasing from 0.85 to approximately 0.80. This indicates that visual noise, commonly found in surveillance systems, particularly those utilizing low-cost cameras or insufficient lighting, exhibits reduced resistance to the CNN-based MobileNet-SSD architecture.

YOLOv8n's architecture is more complex than that of MobileNet-SSD, particularly in terms of layer structure and parameter count, despite being the lightweight variant of the YOLO family. The results indicate that the model achieves an F1-score of 0.90 after just 50 epochs of training, demonstrating competitive performance. Extending the training period to 100 epochs enhances the model's effectiveness, resulting in an F1-score of 0.94. This improvement suggests that the model continues to optimize feature extraction and has not yet reached overfitting, reflecting efficient convergence across epochs.

YOLOv8 requires increased computational time and resources for training, thus demanding high-performance hardware, such as the A100 GPU, utilized in the Google Colab environment. These requirements must be thoroughly evaluated prior to implementation in embedded systems, as they may impose constraints in resource-limited scenarios. This highlights a key limitation of YOLOv8, as deployment on low-power platforms may not be feasible without model compression, pruning, or quantization. Table 6 presents a summary of the relative advantages and disadvantages of the two models examined in this study.

Table 5. Evaluation metric comparison

Metriks evaluasi	YOLOV8n		CNN Mobile Net[18]
	100 Epoch	50 Epoch	
Precision	95%	90%	86.5%
Recall	93%	90%	84.1%
F1-Score	94%	90%	85.2%
Accuracy	-	-	88.2%
mAP50	92%	88%	Not Available
mAP50-95	85%	83%	Not Available

Table 6. Comparative Strength

Criteria	Yolov8n	MobileNet-SSD
Detection accuracy	Very high (F1:0,94)	Moderate (F1:0,85)
Noise robustness	High (stable in low light conditions)	Susceptible to poor lighting conditions
Training requirements	High (requires GPU and longer training time)	Lighter (less resource-intensive)
Embedded compatibility	Yes, with model optimization	Highly suitable for low-power devices
Production scalability	Highly suitable for smart surveillance	Suitable for lightweight or small-scale apps

The results of this investigation indicate that YOLOv8 is the superior choice when the primary concerns are detection accuracy and reliability. However, MobileNet-SSD remains a practical choice when hardware limitations are significant, particularly in edge computing deployments using Jetson Nano or similar platforms. Although MobileNet-SSD may remain viable for less dynamic applications, the findings overall suggest that YOLOv8n is the superior model for systems requiring precise and rapid detection of human objects.

4. CONCLUSION

This study provides a comparative performance evaluation between YOLOv8n and MobileNet-SSD in human detection tasks. Based on an annotated dataset of 12,334 images, YOLOv8n achieved superior detection performance, attaining an F1-score of 0.94, which demonstrates strong robustness against diverse lighting conditions, complex backgrounds, and varying human postures. In contrast, the MobileNet-SSD model, although lightweight and suitable for low-power deployments, showed limited adaptability and lower detection reliability in challenging visual environments.

The strengths of this study lie in its direct benchmarking of a modern single-stage detection architecture against a classical CNN-based lightweight model, while also evaluating performance under realistic environmental conditions. However, this research has notable limitations, including the high computational demands and extended training duration required by YOLOv8n. Additionally, the MobileNet-SSD performance values were referenced from prior literature rather than retrained on identical data, which may contribute to baseline discrepancies.

For future work, optimization techniques such as quantization, pruning, mixed-precision acceleration, and TensorRT deployment are recommended to enable real-time operation of YOLO models on edge-computing platforms with limited processing resources. Furthermore, future studies should retrain both MobileNet-SSD and the latest YOLO architectures (e.g., YOLOv9, YOLOv10, and subsequent releases) using the same unified dataset to ensure standardized benchmarking and eliminate comparative bias. Broader testing should also include real-time multi-camera surveillance environments, occlusion-heavy scenarios, high-density crowd tracking, and extreme illumination conditions. Employing temporal identity tracking modules, such as DeepSORT or ByteTrack, is also encouraged to enhance continuity in dynamic scene analysis.

REFERENCES

[1] I. Nihayatul Husna, M. Ulum, A. Kurniawan Saputro, D. Tri Laksono, and D. Neipa Purnamasari, "Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN)," *Seminar Nasional Fortei Regional*, vol. 7.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.10934

[3] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," 2021. [Online]. Available: https://github.com/ultralytics/yolov3

[4] Y. Wang, H. Wang, and Z. Xin, "Efficient Detection Model of Steel Strip Surface Defects Based on YOLO-V7," *IEEE Access*, vol. 10, pp. 133936–133944, 2022, doi: 10.1109/ACCESS.2022.3230894.

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

[6] H. Herdianto, D. Nasution, H. Hafni, and S. Ramadhan, "Penerapan Deep Learning Yolo Untuk Deteksi Manusia," 2024.

[7] H. Herdianto, H. Hafni, D. Nasution, and S. Ramadhan, "Implementasi Metode Yolo pada Deteksi Objek Manusia," *METHOMIKA Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, vol. 8, no. 2, pp. 234–240, Oct. 2024, doi: 10.46880/jmika.Vol8No2.pp234-240.

[8] 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/

[9] J. Justam, A. Malik, E. Erlita, D. Mangellak, and Y. Yuyun, "Perbandingan Kinerja YOLO vs Faster R-CNN untuk Deteksi & Estimasi Berat Ikan," *Jurnal Ilmiah Sistem Informasi dan Teknik Informatika (JISTI)*, vol. 7, no. 2, pp. 363–376, Oct. 2024, doi: 10.57093/jisti.v7i2.273.

[10] T. Palwankar and K. Kothari, "Real Time Object Detection using SSD and MobileNet," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 3, pp. 831–834, Mar. 2022, doi: 10.22214/ijraset.2022.40755.

[11] I. Inayatul Arifah, F. Nur Fajri, and G. Qorik Oktagalu Pratamasunu, "Deteksi Tangan Otomatis Pada Video Percakapan Bahasa Isyarat Indonesia Menggunakan Metode YOLO Dan CNN," 2022. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[12] F. Habib Hawari, F. Fadillah, M. Rifqi Alviandi, and T. Arifin, "Klasifikasi Penyakit Padi Menggunakan Algoritma CNN (Convolutional Neural Network)," *JURNAL RESPONSIF*, vol. 4, no. 2, pp. 184–189, 2022, [Online]. Available: https://ejurnal.ars.ac.id/index.php/jti

[13] N. Khairunisa, . C., and A. Jamaludin, "Analisis Perbandingan Algoritma CNN dan YOLO dalam Mengidentifikasi Kerusakan Jalan," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4434.

[14] M. R. Dewanto, M. N. Farid, M. A. Rafdi Syah, A. A. Firdaus, and H. Arof, "YOLO vs. CNN Algorithms: A Comparative Study in Masked Face Recognition," *Scientific Journal of Informatics*, vol. 11, no. 1, pp. 139–146, Feb. 2024, doi: 10.15294/sji.v11i1.48723.

[15] 디지털콘텐츠공학과원광대학교, 전자전기공학부단국대학교, Y.-H. Lee, and Y. Kim, "객체 검출을 위한 CNN과 YOLO 성능 비교 실험 Comparison of CNN and YOLO for Object Detection," 2020.

[16] Z. Su, "Comparative Analysis of CNN-Based Object Detection Models: Faster R-CNN, SSD, and YOLO," 2025.

[17] R. Muwardi, J. Mada, R. Permana, H. Gao, and M. Yunita, "Human Object Detection for Real-Time Camera using Mobilenet-SSD," *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 3, no. 2, pp. 141–150, 2023, doi: 10.5162/jiae.v3i2.108.

[18] M. Abdul Aziz, A. S. Rachman, I. Made, and B. Suksmadana, "Pengujian Deteksi Objek Manusia Menggunakan Jetson Nano Dengan Model Ssd Mobilenetv2," *Jurnal Informatika Teknologi dan Sains*, 2024.

[19] T. A. R. Shyaa and A. A. Hashim, "Enhancing real human detection and people counting using YOLOv8," in *BIO Web of Conferences*, EDP Sciences, Apr. 2024. doi: 10.1051/bioconf/20249700061.

[20] N. Ma Muriyah, J. H. Sim, and A. Yulianto, "Evaluating YOLOv5 and YOLOv8: Advancements in Human Detection," *Journal of Information Systems and Informatics*, vol. 6, no. 4, pp. 2999–3015, Dec. 2024, doi: 10.51519/journalisi.v6i4.944.
[21] A. Gallu, A. R. Himamunanto, and H. Budiati, "Pengenalan Emosi pada Citra wajah menggunakan Metode YOLO," 2024.
[22] M. Abdurahman Kamil and Y. Mitha Djaksana, "JIIC: Jurnal Intelek Insan Cendikia Perbandingan Algoritma Convolutional Neural Network (CNN) Dan Algoritma You Only Look Once (YOLO) Untuk Deteksi Wajah Comparison Of Convolutional Neural Network (CNN) Algorithm And You Only Look Once (YOLO) Algorithm For Facial Detection", [Online]. Available: https://jicnusantara.com/index.php/jiic
[23] A. A. Rasjid, B. Rahmat, and A. N. Sihananto, "Implementasi YOLOv8 Pada Robot Deteksi Objek," *Journal of Technology and System Information*, vol. 1, no. 3, p. 9, Jul. 2024, doi: 10.47134/jtsi.v1i3.2969.
[24] N. Dwi Grevika Drantantiyas *et al.*, "Performasi Deteksi Jumlah Manusia Menggunakan YOLOv8," 2023. [Online]. Available: https://universe.roboflow.com/csgo-head-detection/head-datasets
[25] M. Vilar-Andreu, L. García, A.-J. Garcia-Sanchez, R. Asorey-Cacheda, and J. Garcia-Haro, "Enhancing Precision Agriculture Pest Control: A generalized Deep Learning Approach with YOLOv8-based Insect Detection," 2023, doi: 10.1109/ACCESS.2023.0322000.
[26] N. Wakhidah, T. Pungkasanti, A. Praba, and R. Pinem, "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Deteksi Objek menggunakan Deep Learning untuk Mengetahui Tingkat Kerumunan Mahasiswa".
[27] L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once)," 2021.
[28] I. Wayan Suartika E.P, A. Y. Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101".
[29] D. Permata Sari, M. Arya Cendekia Putra, and R. Kusumanto, "Implementasi Pengenalan Wajah Berbasis CNN Dan RFID Untuk Area Akses Aman Di Fasilitas Ruang Dosen Polsri Dengan Hosting Lokal", doi: 10.5281/zenodo.13047545.
[30] A. Reza Fahcruroji, M. Yunita Wijaya, and I. Fauziah, "Implementasi Algoritma CNN Mobilenet Untuk Klasifikasi Gambar Sampah Di Bank Sampah".
[31] C. Kalamani, R. Bhuvaneswaran, S. Dhanapal, and S. Rajkumar, "Animal and Violence Detection Using Artificial Neural Network." [Online]. Available: www.ijmer.com

BIBLIOGRAPHY OF AUTHORS

Dewi Permata Sari is a lecturer in the Department of Electronics Engineering at the Polytechnic of Sriwijaya, Palembang, Indonesia. She obtained her bachelor's degree in Electrical Engineering from Universitas Sriwijaya in 2001, and later completed her master's degree in High Voltage Electrical Engineering at Universitas Bina Darma in 2012. Her teaching areas include SCADA systems, automation, LabVIEW, microcontrollers, and industrial safety standards. Her research interests focus on embedded systems, IoT-based automation, and robotics for educational and public service applications. She can be contacted at email: dewi_permatasari@polsri.ac.id

M. Akbar Tri Ramadhani is a student currently pursuing his studies at Politeknik Negeri Sriwijaya, focusing in Electrical Engineering. With a keen interest in automation, control systems, and intelligent technologies, Akbar actively enhances both his theoretical understanding and practical skills through academic studies and hands-on technological projects. As an Electrical Engineering student, he is noted for his focus and attention in mastering areas such as embedded systems, sensor-based automation, and AI-based control techniques. Akbar also exhibits a strong excitement for contributing to the design of smart, efficient, and innovative engineering solutions in line with the rapid growth of Industry 4.0 technologies.

Abdurrahman, ST, M.Kom, is a lecturer at the Department of Electrical Engineering at Politeknik Negeri Sriwijaya. He completed his undergraduate education in Engineering and continued his master's studies in Computer. He actively teaches courses related to digital systems, programming, and information technology, while also guiding students in their final projects and research. In addition to teaching, Abdurrahman is also involved in the development of applied technologies that integrate electronic and computing systems.