Journal of Soft Computing Exploration

Homepage: https://shmpublisher.com/index.php/joscex

p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Improved human image density detection with comparison of YOLOv8 depth level architecture and drop-out implementation

Winda Yulita*[1], Uri Arta Ramadhani[2], Zunanik Mufidah[3], Gde KM Atmajaya [4], Radhinka Bagaskara[5] Rahman Indra Kesuma[6], Mohamad Meazza Aprilianda[7]

[1,5,6,7]Department of Informatics Engineering, Institut Teknologi Sumatera, Indonesia
[2,4]Department of Electrical Engineering, Institut Teknologi Sumatera, Indonesia
[3]Department of Biosystem Engineering, Institut Teknologi Sumatera, Indonesia

Article Info

Article history:

Received February 28, 2025
Revised March 15, 2025
Accepted March 26, 2025

Keywords:

Deep learning
YOLOv8
Human density detection
Smart building
Depths level

ABSTRACT

Energy inefficiency due to Air Conditioners (AC) running in empty rooms contribute to unnecessary energy consumption and increased CO_2 emissions. This study explores how different depth levels of the YOLOv8 architecture and dropout regularization can enhance human density detection for smarter AC control systems. By evaluating model accuracy through Mean Average Precision (mAP50-95), we provide quantitative insights into how these modifications improve detection performance. Our dataset consists of 1363 images taken in an office environment at ITERA under varying lighting conditions and different human presence densities. The results show that the YOLOv8m model performs best, achieving an mAP50-95 score of 0.814 in training and 0.813 in validation, outperforming other YOLOv8 variants. Furthermore, applying dropout regularization improves model generalization, increasing mAP50-95 from 0.552 to 0.6 and effectively reducing overfitting. This study highlights the balance between architectural depth and dropout regularization in YOLOv8, demonstrating its effectiveness in energy-efficient smart buildings. The findings support the potential of deep learning-based human density detection in improving energy conservation strategies, making it a valuable solution for intelligent automation systems.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Winda Yulita,
Department of Informatics Engineering,
Institut Teknologi Sumatera,
Jl. Terusan Ryacudu, Way Huwi, Kec. Jati Agung, Kabupaten Lampung Selatan, Lampung,
Email: winda.yulita@if.itera.ac.id
https://doi.org/10.52465/joscex.v6i1.556

1. INTRODUCTION

Many people keep the Air Conditioner (AC) running even if no one is in the room or when the room is empty, causing in excessive electricity use [1], [2]. For example, in the office of University Institut Teknologi Sumatera (ITERA), such behavior significantly contributes to increased energy consumption and operational costs. Additionally, poor performance of AC can contribute to the issue of global warming through high level of CO2 [3]. Hence an intelligent, adaptive AC temperature control module to be developed based on the number of people in the room.

A potential solution seems to be the use of Computer Vision using Deep Learning to be able to identify humain density in real-time. For example, using this technology, the number of individuals present in a room can be identified and subsequently utilized as a metric for automatic AC climate control. You Only Look Once (YOLO) is one of the most potent object-detection methods widely used in practice, where its latest version, YOLOv8, provides a massive increase in accuracy and inference speed over its previous versions [4], [5]. Due to these benefits, YOLOv8 can be used in human density monitoring systems to optimize energy efficiency.

Human density detection plays a crucial role in various applications, including crowd monitoring, public safety, urban planning, and resource management [6]. The development of deep learning-based object detection models, particularly the You Only Look Once (YOLO) architecture, has significantly improved real-time detection accuracy and efficiency. YOLOv8 incorporates enhanced depth configurations and optimization strategies, making it a promising tool for human image data-based density detection [7]. However, optimizing the architecture for specific applications remains a challenge, particularly in determining the most effective depth configurations and regularization techniques such as drop-out implementation [8].

In recent years, the integration of deep learning models in human density detection has demonstrated significant potential in optimizing energy consumption in smart buildings. However, existing approaches still face limitations in achieving a balance between computational efficiency and model accuracy, particularly when applied in real-time scenarios. While architectures like Faster R-CNN and SSD offer precise detection, they often struggle with high processing demands and poor generalization in varying environmental conditions. YOLO-based models have addressed some of these issues, but there remains a gap in understanding how architectural depth and dropout regularization influence detection performance. This study addresses this gap by systematically comparing YOLOv8 [9], [10] variants and examining the impact of dropout on model robustness. By providing empirical evidence on the trade-offs between these architectural choices, our research contributes to the advancement of energy-efficient automation solutions in smart building environments.

Despite these advancements, the impact of architectural depth level varians in YOLOv8, particularly in human density detection, remains underexplored. While deeper networks generally offer improved feature extraction capabilities, they also introduce higher computational costs and potential overfitting. Drop-out regularization is commonly applied to mitigate overfitting, yet its optimal implementation in YOLOv8 architectures is not well-documented. Understanding the trade-offs between model depth and drop-out implementation is essential for maximizing detection performance in real-world scenarios [11].

This study specifically compares the effects of different YOLOv8 depth level architectures on human density detection accuracy. The research objectives are to (1) evaluate the impact of varying YOLOv8 depths on detection accuracy, (2) determine the optimal depth level for balancing accuracy and computational efficiency, and (3) analyze the effectiveness of dropout regularization in reducing overfitting and improving model generalization. To achieve these objectives, a case study was conducted in an ITERA office room under varying lighting conditions, object positioning, and density variations. The evaluation focuses on detection accuracy metrics and the feasibility of deploying the model in an automated AC system.

The primary contribution of this research lies in its systematic comparison of YOLOv8 architectural depth variations and the role of dropout in enhancing model robustness. Unlike previous studies that focus on general YOLO-based detection, this study provides empirical insights into optimizing YOLOv8 for human density detection in smart buildings. The findings validate the potential of deep learning-based human detection systems in improving energy conservation strategies, reinforcing their applicability in real-world intelligent automation solutions.

2. METHOD

The research was conducted by implementing and testing the performance of YOLO in its latest iteration, which is YOLOv8. Using YOLOv8, we explore the potential of detecting human density in rooms based on image detection. We were also analyzing the impact of drop-out regularization on its overall detecting performance. The research procedure is seen in Figure 1.

.

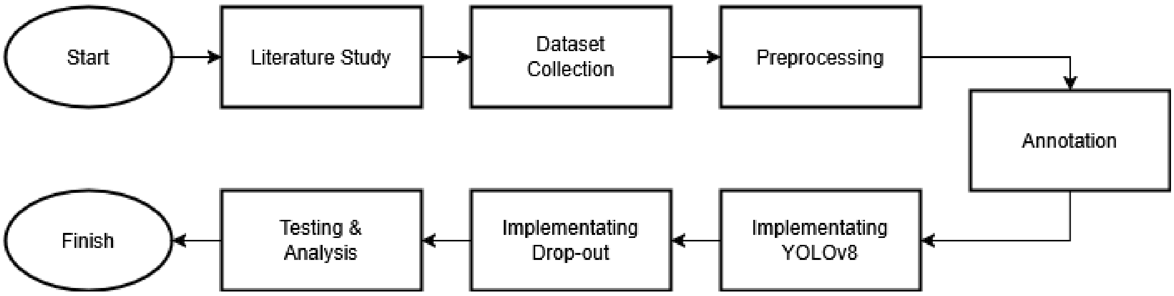

Figure 1. Research flow

Dataset Collection

The dataset, totalling 1363 images, is taken from meeting rooms in Building A at Institut Teknologi Sumatera. Each image contains varying amounts of people in the room, with different positions such as standing or sitting on a chair. There are two main rooms that were taken: the International Office meeting room and the Rector meeting room. From the total images, 415 images were taken from the International Office meeting room and 948 images were taken from the Rector meeting room. Figure 2 shows how the pictures are normally taken from each room.

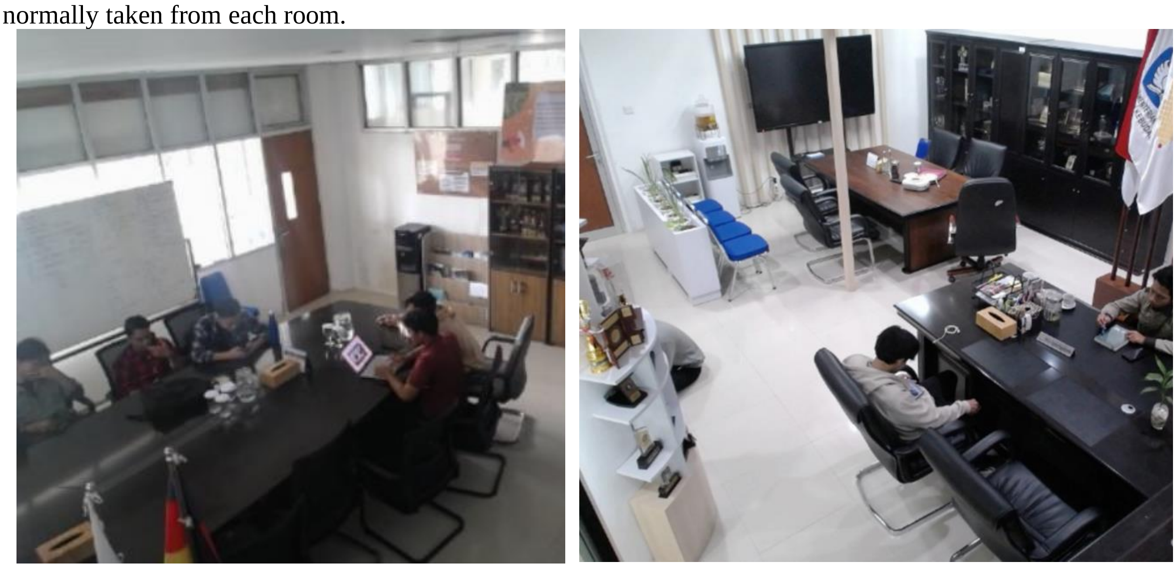

Figure 2. Dataset sample from International Office meeting room and the Rector meeting room

Preprocessing

The preprocessing stage can improve image quality and make it easier to detect objects. This step will resize each image to a specific size: 320 x 320 pixels. This step is crucial, in order to standardize the images and reduce their size, ensuring the model to reach decent efficiency [12]. This standardization also helps reduce the variation in object dimensions in the dataset, so that the model can generalize better [13].

Annotation

After the images have been resized, the dataset needs to be annotated with labels. The labels contain bounding box data, with several attributes such as: class, bounding box coordinate (xmin, ymin), bounding box height, and bounding box width. The labels are then saved into a .txt file, corresponding to each image. Each bounding box specifically annotates human heads in each image. These annotations will be then used for training the YOLOv8 model.

In addition to annotating human heads with bounding boxes, it is crucial to ensure that these annotations follow the YOLOv8 format, which requires specifying the class ID, normalized center coordinates (x_center, y_center), and normalized width and height of the bounding box [14]. Normalization ensures that the bounding box parameters remain consistent across different image resolutions by dividing their values by the respective image width and height, resulting in values between 0 and 1 [15]. The quality of annotation is a critical factor in the performance of object detection models, as poorly labeled data can significantly impact training outcomes and lead to inaccurate detections [16].

The annotated dataset then will be divided into three dataset groups, which is training dataset, validation dataset, and testing dataset. Training dataset will be used for training the YOLOv8 model. Validation dataset will be used to measure the model's performance and to avoid overfitting. Testing dataset will be used to test the accuracy of the mode, in which the testing dataset contains no repeated images from training dataset nor validation dataset. The ratio to divide the dataset is 60:25:15 for training dataset, validation dataset, and testing dataset, respectively.

YOLOv8 Implementation

The first stage will be an experiment to find the right hyperparameters to detect human objects using YOLOv8. The range of values or types of hyperparameters tested for YOLOv8 can be seen in Table 1 [17].

Table 1. Tested YOLOv8 hyperparameters

	Parameter
Optimizer	SGD, AdamW, RMSProp
Epoch	min = 0, max = 100
Batch	8, 16
Learning Rate	0.01, 0.001

The hyperparameter combinations in Table 2 are obtained from the results of all hyperparameter combinations that will be tested. All hyperparameter combinations are obtained using the grid search method. Each hyperparameter combination will be used for the YOLOv8 model training process which aims to obtain the best model.

Table 2. YOLOv8 hyperparameter combinations

Combination To	Epoch	Optimizer	Learning Rate	Batch Size
1	[0, 100]	SGD	0.01	8
2	[0, 100]	AdamW	0.01	8
3	[0, 100]	RMSProp	0.01	8
4	[0, 100]	SGD	0.001	8
5	[0, 100]	AdamW	0.001	8
6	[0, 100]	RMSProp	0.001	8
7	[0, 100]	SGD	0.01	16
8	[0, 100]	AdamW	0.01	16
9	[0, 100]	RMSProp	0.01	16
10	[0, 100]	SGD	0.001	16
11	[0, 100]	AdamW	0.001	16
12	[0, 100]	RMSProp	0.001	16

The next stage is that the model with the best hyperparameters will be used to compare various architectures based on the level of depth, namely YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8xv [18].

Drop-out Generalization

Once the best architecture of YOLOv8 was identified, the model was implemented with and without the drop-out technique to evaluate its impact on model generalization. Drop-out was applied at specific layers in the network to reduce the risk of overfitting and improve model performance on previously unseen data [19], [20]. Studies indicate that deeper architectures can capture more complex features; however, without proper regularization, they are prone to overfitting [21]. To mitigate this, dropout was strategically introduced in selected layers, particularly the fully connected layers, as this approach has been shown to enhance model robustness by preventing co-adaptation of neurons [22]. Additionally, the dropout rate was experimentally adjusted to optimize model performance, following methodologies outlined in prior research [7], [23]. The evaluation process involved training the model on the dataset using both configurations—one with dropout enabled and one without—allowing for a comparative analysis of its effect on accuracy and generalization.

Testing

The evaluation of the model performance in this study was carried out using the Mean Average Precision (mAP50-95) metric, which is a standard measure in object detection to assess the accuracy of the model at various levels of Intersection over Union (IoU). The mAP50-95 metric is calculated by taking the average of the precision values in the IoU range of 0.5 to 0.95 with an interval of 0.05, thus providing a comprehensive picture of the model's ability to recognize objects at different levels of accuracy [24]. The metric mAP (mean Average Precision) is represented as:

.

$$mAP = \frac{\sum_{i=1}^{N} AP}{N}$$

3. RESULTS AND DISCUSSIONS

In this study, we explore the performance improvement of Deep Learning models for human image-based density detection by comparing different depth variants of the YOLOv8 architecture and analyzing the impact of applying drop-out as a regularization technique. The experiments were conducted in three stages. The first stage is to determine the modeling with the best hyperparameters. The results of the comparison of the tested hyperparameters are shown in Table 3.

Table 3. YOLOv8 training results of each hyperparameter

Combinations	Epoch	Optimizer	Learning Rate	Batch Size	Val Loss	mAP@50:95
1	[0, 100]	SGD	0.01	8	5.162	0.299
2	[0, 100]	AdamW	0.01	8	4.707	0.308
3	[0, 100]	RMSProp	0.01	8	6.172	0.155
4	[0, 100]	SGD	0.001	8	4.875	0.296
5	[0, 100]	AdamW	0.001	8	4.758	0.301
6	[0, 100]	RMSProp	0.001	8	5.938	0.142
7	[0, 100]	SGD	0.01	16	4.909	0.286
8	[0, 100]	AdamW	0.01	16	4.712	0.328
9	[0, 100]	RMSProp	0.01	16	5.971	0.149
10	[0, 100]	SGD	0.001	16	4.856	0.289
11	[0, 100]	AdamW	0.001	16	4.720	0.316
12	[0, 100]	RMSProp	0.001	16	5.727	0.139

Hyperparameters in the 8th combination are declared to be the best configuration because they have the highest mAP@50-95 value (0.328), showing excellent performance in detecting heads with varying levels of overlap, which is important for detection consistency in real conditions. Although it has a slightly higher Validation Loss (4.712) and a slightly lower mAP@50 (0.411) than the 2nd hyperparameter combination, the mAP@75 value has a higher value (0.137). So the 8th combination was chosen as the most suitable hyperparameter configuration for the model and dataset in this study.

The second stage aims to determine the most optimal YOLOv8 variant based on the evaluation of the YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x models. Figure 3 shows the results of YOLOv8 detection with various depths levels.

YOLOv8s

YOLOv8m

YOLOvl

YOLOv8x

Figure 3. Human detection results with the YOLOv8 model (x, m, l, x)

The performance of the model in the first stage is evaluated using the Mean Average Precision (mAP50-95) metric in the training (train) and validation (val) stages. This metric is used to measure the model's ability to detect objects with varying levels of Intersection over Union (IoU) accuracy. Table 4 presents the performance evaluation results of each YOLOv8 variant, namely YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x.

Table 4. Performance comparison of YOLOv8(x,m,l,x) models

Model	Training (mAP50-95)	Validation (mAP50-95)
YOLOv8x	0.804	0.805
YOLOv8m	0.814	0.813
YOLOv8l	0.813	0.813
YOLOv8x	0.811	0.812

Based on the results of the experiments conducted, the YOLOv8m model showed the best performance in detecting density based on human image data compared to other YOLOv8 variants. This is indicated by the mAP50-95 value at the training stage (train) of 0.814 and validation (val) of 0.813, which is the highest value among all models tested. In comparison, the YOLOv8s model, which is a variant with a lighter architecture, has an mAP50-95 of 0.804 at the training stage and 0.805 at the validation stage, while YOLOv8x, which is a model with a more complex architecture, actually experienced a slight decrease in performance with an mAP50-95 of 0.811 at training and 0.812 at validation. This finding indicates that increasing the depth of the architecture is not always directly proportional to increasing detection accuracy.
The next stage is to analyze the effect of applying drop-out as a regularization technique in improving model generalization and reducing overfitting by using different environments. The results of the model performance comparison obtained from this experiment are presented in Table 5.

Table 5. Comparison of Dropout Effects on YOLOv8 Model Performance

Model	Epoch	Optimizer	Learning Rate	Batch Size	Dropout	Validation (mAP50-95)
YOLOv8m without dropout	[0,100]	AdamW	0.01	16	-	0.552
YOLOv8m with dropout	[0,100]	AdamW	0.01	16	0.5	0.6

The comparison results show that YOLOv8m without drop-out experiences a performance degradation with a mAP50-95 value of 0.552, while the application of drop-out results in a performance improvement with a mAP50-95 of 0.6. These findings indicate that drop-out contributes to improving the generalization of the model in different environments, so it can be an effective strategy in improving the robustness of density detection based on human image data.

.

4. CONCLUSION

This study explores the performance improvement of Deep Learning models in human image-based density detection through a comparison of various depth variants of the YOLOv8 architecture and an analysis of the application of dropout as a regularization technique. The experimental results show that YOLOv8m has the best performance compared to other variants, with a Mean Average Precision (mAP50-95) value of 0.814 in the training stage and 0.813 in the validation stage. This indicates that increasing the depth of the architecture is not always directly proportional to increasing detection accuracy. In addition, the application of dropout has been shown to improve model generalization, where YOLOv8m without dropout experienced a decrease in performance with mAP50-95 of 0.552, while with dropout, the model gained a performance increase of up to 0.6. These findings confirm that dropout can be an effective strategy in improving model resilience to environmental variations, thus contributing to the development of a more accurate and robust Deep Learning-based density detection system. For future studies, increasing the dataset's size and further experimentations on other hyperparameters configurations has potential to increase the overall model's performance. Additionally, while the drop-out as a regularization technique did improved the current model, further studies towards other regularization techniques could unlock the full potential of YOLOv8's modelling performance.

REFERENCES

[1] Y. Peng *et al.*, "Energy Consumption Optimization for Heating, Ventilation and Air Conditioning Systems Based on Deep Reinforcement Learning," *IEEE Access*, vol. 11, no. July, hal. 88265–88277, 2023, doi: 10.1109/ACCESS.2023.3305683.

[2] V. L. Erickson dan A. E. Cerpa, "Occupancy based demand response HVAC control strategy," *BuildSys'10 - Proc. 2nd ACM Work. Embed. Sens. Syst. Energy-Efficiency Build.*, no. May, hal. 7–12, 2010, doi: 10.1145/1878431.1878434.

[3] A. M. Vicedo-Cabrera *et al.*, "The burden of heat-related mortality attributable to recent human-induced climate change," *Nat Clim Chang*, vol. 11, no. 6, hal. 492–500, 2021.

[4] I. P. Sary, S. Andromeda, dan E. U. Armin, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images," *Ultim. Comput. J. Sist. Komput.*, vol. 15, no. 1, hal. 8–13, 2023, doi: 10.31937/sk.v15i1.3204.

[5] S. Rajakumar dan R. B. Azad, "A novel YOLOv8 architecture for human activity recognition of occluded pedestrians," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 5, hal. 5244–5252, 2024, doi: 10.11591/ijece.v14i5.pp5244-5252.

[6] D. Huang, Y. Li, J. Qu, S. Zhang, dan Q. Wang, "YOLOv8-MGH: Dense Crowd Object Detection," *Adv. Comput. Graph. *, vol. 15338, hal. 133–144, 2025, doi: https://doi.org/10.1007/978-3-031-81806-6_10.

[7] J. Terven, D. M. Córdova-Esparza, dan J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, hal. 1680–1716, 2023, doi: 10.3390/make5040083.

[8] A. Lemay *et al.*, "Improving the repeatability of deep learning models with Monte Carlo dropout," *npj Digit. Med.*, vol. 5, no. 1, hal. 1–11, 2022, doi: 10.1038/s41746-022-00709-3.

[9] Z. Zhou, H. Li, S. Zhou, L. Yan, dan H. Yang, "A deep learning-based algorithm for fast identification of multiple defects in tunnels," *Eng. Appl. Artif. Intell.*, vol. 145, 2025, doi: https://doi.org/10.1016/j.engappai.2025.110035.

[10] Z. Qi, D. Pan, T. Niu, Z. Ying, dan P. Shi, "Bridge the gap between practical application scenarios and cartoon character detection: A benchmark dataset and deep learning model," *Displays*, vol. 84, 2024, doi: https://doi.org/10.1016/j.displa.2024.102793.

[11] J. Zhao *et al.*, "A deep learning-based study on visual quality assessment of commercial renovation of Chinese traditional building facades," *Environ. Impact Assess. Rev.*, vol. 113, 2025, doi: https://doi.org/10.1016/j.eiar.2025.107862.

[12] N. P. Damayanti, M. N. D. Ananda, dan F. W. Nugraha, "Lung cancer classification using convolutional neural network and DenseNet," *J. Soft Comput. Explor.*, vol. 4, no. 3, hal. 133–141, 2023, doi: 10.52465/joscex.v4i3.177.

[13] D. Magdaleno, M. Montes, B. Estrada, dan A. Ochoa-Zezzatti, "Utilizing a YOLOv8 Segmentation-Based Model for Automated Defect Detection in Bread Images," *Innov. Appl. Artif. Neural Networks to Data Anal. Signal Process.*, hal. 499–532, 2024, doi: https://doi.org/10.1007/978-3-031-69769-2_20.

[14] P. Wspanialy, J. Brooks, dan M. Moussa, "An Image Labeling Tool and Agricultural Dataset for Deep Learning," no. September, 2020, doi: 10.48550/arXiv.2004.03351.

[15] A. Alkalbani, M. Saqib, A. S. Alrawahi, A. Anwar, C. Adak, dan S. Anwar, "RDD4D: 4D Attention-Guided Road Damage Detection And Classification," hal. 1–13, 2025, [Daring]. Tersedia pada: http://arxiv.org/abs/2501.02822.

[16] S. Kale, P. D., Mahajan, T., Kanawade, A., Shete, S., & Jadhav, "Comparative Analysis of Image Annotation Tools: LabelImg, VGG Annotator, Label Studio, and Roboflow," vol. 11, no. 5, hal. n398–n403, 2024.

[17] K. Kim, K. Kim, dan S. Jeong, "Application of YOLO v5 and v8 for Recognition of Safety Risk Factors at Construction Sites," *Sustainability*, vol. 15, no. 20, hal. 15179, 2023, doi: 10.3390/su152015179.

[18] U. Patel, R. Vaghela, Y. Popat, H. Patel, J. Sarda, dan A. Bhoi, "Multi-Class Event Classification using YOLOv8," *15th Int. Conf. Inf. Commun. Technol. Converg.*, hal. 2078–2081, 2024.

[19] M. Khan, "Understanding dropout in deep neural networks," *IEEE Comput. Intell. Mag*, vol. 18, hal. 45–53, 2023.

[20] S. R. Vinta, B. Dhanalaxmi, A. T. M. Lavanya, dan Y. Raju, "Banana Disease and Deficiency Detection Using YOLOv8," *Mach. Vis. Augment. Intell.*, vol. 1211, 2024, doi: https://doi.org/10.1007/978-981-97-4359-9_42.

[21] J. Xie, Z. Ma, J. Lei, G. Zhang, J.-H. Xue, dan Z.-H. Tan, "Advanced Dropout: A Model-free Methodology for Bayesian Dropout Optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, hal. 4605–4625, 2022, doi: 10.1109/TPAMI.2021.3083089.

[22] P. Hidayatullah, N. Syakrani, M. R. Sholahuddin, T. Gelar, dan R. Tubagus, "YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review," 2025, [Daring]. Tersedia pada: http://arxiv.org/abs/2501.13400.

[23] M. R. Ningsih *et al.*, "Sign Language Detection System Using YOLOv5 Algorithm to Promote Communication Equality People with Disabilities," *Sci. J. Informatics*, vol. 11, no. 2, hal. 549–558, 2024, doi: 10.15294/sji.v11i2.6007.

[24] X. Li, Q. Wang, X. Yang, K. Wang, dan H. Zhang, "Track Fastener Defect Detection Model Based on Improved YOLOv5s," *Sensors*, vol. 23, no. 14, 2023, doi: 10.3390/s23146457.