p-ISSN : 2301-8402, e-ISSN : 2685-368X, available at : https://ejournal.unsrat.ac.id/index.php/elekdankom

Human Pose Estimation Using LeYOLO-Nano Architecture

Vicky Nolant Setyanto Lahimade, Imanuel Kutika,
Tomi Todingan, Vecky Poekoel, Muhamad Dwisnanto Putro
Master Program of Informatics, Postgraduate Program, Sam Ratulangi University Manado
e-mails : vickylahimade026@student.unsrat.ac.id, imanuelkutika026@student.unsrat.ac.id,
tomitodingan026@student.unsrat.ac.id, vecky.poekoel@unsrat.ac.id, dwisnantoputro@unsrat.ac.id
Received: 25 April 2025; revised: 27 Mei 2025; accepted: 16 Juni 2025

***Abstract* — Human pose estimation is essential in numerous practical applications, particularly in scenarios demanding fast processing and optimal resource usage, such as surveillance, human with computer interaction, and robotic systems. This technology aims to detect and analyze human body keypoints in images or videos, which is a critical step in understanding an individual's movements and behaviors. This article evaluates the performance of the original LeYOLO-Nano architecture, a lightweight variant of the YOLO architecture, in the application of human body keypoint detection for the purpose of pose estimation. Using MSCOCO 2017 dataset, which includes a wide range of real-world conditions, this model achieved a mAP50 of 0.69 and a mAP50:95 of 0.362, demonstrating its ability to detect human poses with adequate accuracy. Moreover, the model is capable of handling data at a rate of 20.78 frames per second using a standard CPU, highlighting its effectiveness for real-time use on edge devices with restricted computing power. LeYOLO-Nano enables efficient human pose estimation on low-power devices, with the potential to further optimize speed and accuracy in real-world applications.**

***Key words*— Human pose estimation, keypoint detection, LeYOLO, real-time vision**

I. INTRODUCTION

Human pose estimation involves detecting and identifying the positions main body points in the picture or video [1]. This technology has become increasingly important in various real-world applications such as activity recognition, interactive gaming, healthcare, and rehabilitation [2],[3],[4]. Additionally, pose estimation is also utilized in sign language recognition [5].

Despite significant advancements in pose estimation techniques, most methods still face challenges related to computational complexity and efficiency. High-performance models typically require powerful GPUs, large memory capacity, and long processing times [6]. This makes them difficult to deploy on edge devices such as smartphones, embedded systems, and surveillance cameras. Therefore, lightweight and efficient models are crucial for real-time applications [7].

One of the architectures known for its efficiency is YOLO (You Only Look Once), which processes images in a single pass, enabling fast detection [8]. However, for complex tasks like human pose estimation, YOLO models can still be computationally demanding. To address this, LeYOLO, a lightweight variant of YOLO, was introduced by optimizing the architecture to improve computational efficiency without compromising performance [9]. One of its smallest versions, LeYOLO-Nano, is highly suitable for resource-constrained devices due to its reduced parameters and simplified components.

Several previous studies have adapted the YOLO architecture for human pose estimation, such as YOLO-Pose [10], YOLOv8-PoseBoost [11], Gaussian Matric Mode-YOLO [12], and YOLO-Rlepose [13] which modified the head structure or applied loss functions such as OKS. Zhang et al. [14] proposed an adversarial training method that improved accuracy but was computationally expensive. Jie Ou and Hong Wu [15] using depthwise separable, while Ding et al. [16] introduced a heatmap-free approach. Li et al. [17] and Zhang et al. [18] also offered lightweight models, although they still required significant computational resources.

In contrast to these approaches, this study specifically evaluates the original performance of the LeYOLO-Nano architecture without adding any additional modules, to explore how far its core capabilities can go in efficiently performing human pose estimation. The following is a summary of the main contributions of this research:

1. This paper presents a baseline evaluation of the LeYOLO-Nano model's performance for human pose estimation, focusing solely on its core architecture without the integration of additional modules or attention mechanisms. This allows for a clear assessment of the model's inherent capabilities and efficiency.
2. This work applies the LeYOLO-Nano architecture specifically to human pose estimation, a critical task with broad applicability in domains such as healthcare, human-computer interaction, and activity recognition Khan et al. [19].
3. This research highlights the potential of LeYOLO-Nano as an efficient model for real-time human pose estimation on resource-constrained devices, showcasing its suitability for deployment in practical, real-world applications.

II. METHOD

The LeYOLO-Nano architecture, as shown in Figure 1, is a lightweight model designed for realistic object recognition and human poses estimated on edge devices. It consists of three main components: the Backbone, which serves as the initial part of the network is responsible by extracting important the features of an input picture. At this stage, visual information such as shapes, edges, and basic patterns are encoded into feature representations that can be processed by the network further. LeYOLO-Nano uses a lightweight and efficient convolutional structure as the backbone to remain resource-efficient and suitable for edge devices. In addition, some versions of the backbone can integrate techniques such as depthwise separable convolution or C3 modules to improve efficiency without sacrificing accuracy.

The Neck is in charge of combining and refining features from different depth levels generated by the backbone. Typically, it uses FPN (Feature Pyramid Network) [20] or PANet (Path Aggregation Network) [21] structures to enable multi-scale detection. By combining features from high and low resolution, the neck helps the model recognize objects or human body parts in different sizes. This process is important in pose estimation, as body parts such as hands or feet may be very small in the image, and the head is the final part of the architecture that is responsible for generating the final output in the form of predictions. For object detection, the head of the network is responsible for estimating the bounding box location, identifying the object category, and assigning a confidence level to each prediction. For human pose estimation, the head predicts the coordinates of keypoints such as the shoulders, knees, elbows, head and others. The head is designed to be lightweight yet accurate, using a direct regression strategy on keypoints while considering computational efficiency.

A. Backbone

The Backbone extracts features from input images, starting with STEM, a lightweight convolutional module that reduces spatial size and computational load. The image then passes through Depthwise Separable Convolution and Bottleneck CSP blocks to improve efficiency without losing accuracy. The final component, SPPF (Spatial Pyramid Pooling Fast), enhances global spatial perception by capturing context from multiple receptive field sizes without adding significant parameters. Inverted residual blocks are a key structure in the MobileNetV2 architecture designed to improve computational and memory efficiency on mobile devices. Experimental results indicate that the proposed approach achieves superior classification accuracy while preserving computational efficiency comparable to that of existing lightweight architectures, including MobileNetV1 and ShuffleNet.

1) SPPF

SPPF (Spatial Pyramid Pooling Faster) is a module designed to efficiently enhance the receptive field of convolutional neural networks by pooling feature maps at multiple scales. It improves upon the traditional Spatial Pyramid Pooling (SPP) by using three sequential max pooling operations with a fixed kernel size (typically 5×5), allowing the network to capture multi-scale spatial information more rapidly and with fewer parameters. SPPF is commonly used in lightweight and real-time models such as YOLOv5 and YOLOv8, as it enriches feature representations without adding significant computational overhead.

2) Inverted Residual Block

The Inverted Residual Block, introduced in MobileNetV2, is an architectural innovation designed to optimize the trade-off between performance and computational efficiency, particularly for mobile and real-time deep learning applications. In contrast to traditional residual blocks that increase channel dimensions at the output, the inverted residual block follows a reverse approach: it first expands the input feature map using a pointwise (1×1) convolution, then applies a depthwise separable convolution to process spatial information efficiently, and finally projects the features back to a lower-dimensional representation using another pointwise convolution.

B. Neck

The Neck combines features across scales using FPAN (Faster Path Aggregation Network), a variant of PANet that accelerates feature propagation. FPAN uses top-down and bottom-up pathways with skip-connections to preserve crucial information. Lightweight fusion techniques, like concatenation and 1×1 convolutions, are used to merge features efficiently.

C. Head

The LeYOLO head predicts 17 human body keypoints (e.g., head, shoulders, elbows, hips) for each detected individual. It uses multi-scale predictions with either heatmap regression or offset prediction. The anchor-free mechanism allows flexible pose detection across varying object sizes and positions. The OKS (Object Keypoint Similarity) and MSE (Mean Squared Error) loss functions ensure stable training and accurate keypoint predictions. The full architecture, including STEM, FPAN, Bottleneck CBA, STEM, SPPF, and Head Pose, is illustrated in Figure 2.

D. Keypoints Loss Function

The Object Keypoint Similarity (OKS_total), which is used to evaluate the similarity between predicted and ground truth keypoints in object pose estimation. The formula averages the OKS scores across all visible or labeled keypoints. For each keypoint *i* the individual OKS score OKS*i* is considered only if the keypoint is visible or labeled, as indicated by the visibility variable *vi* the indicator function 1*vi*>0 serves as a filter that assigns a value of 1 when the keypoint is visible (i.e., *vi*>0) and 0 otherwise. This ensures that only valid keypoints contribute to the final score. The numerator of the equation sums the OKS scores of visible keypoints, while the denominator counts how many keypoints are visible. The result is the mean OKS score over only the visible keypoints, which provides a fair and robust metric for evaluating keypoint detection performance without being affected by missing or unlabeled points.

$$OKS_{total} = \frac{\sum_i OKS_i \cdot 1_{v_i} > 0}{\sum_i 1_{v_i} > 0} \quad (1)$$

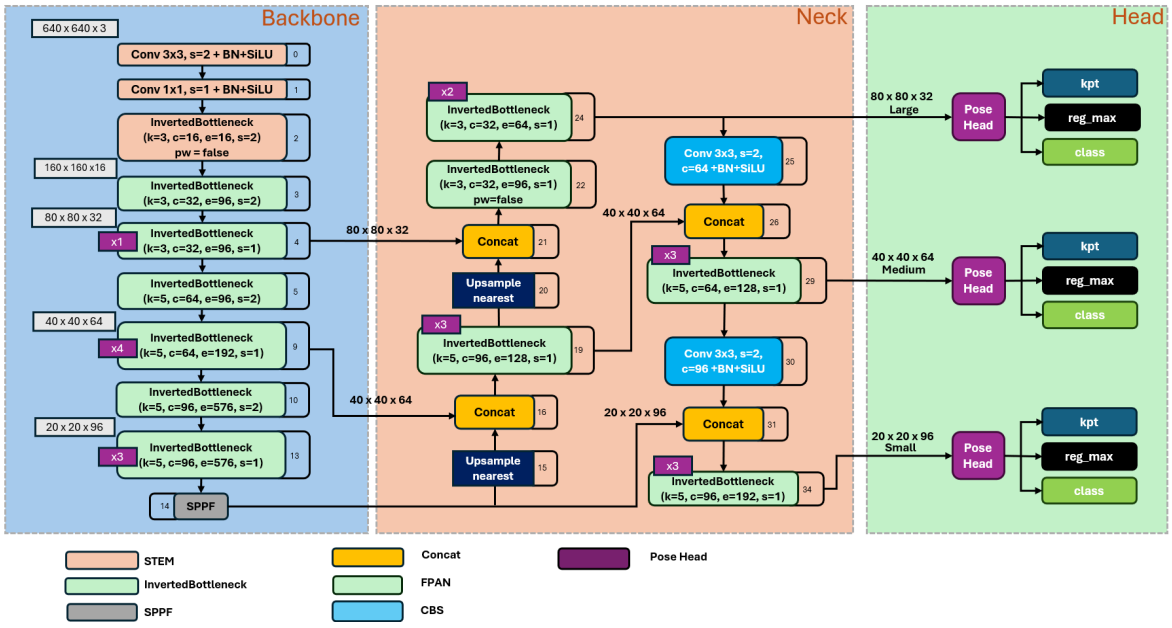

Figure 1. LeYOLO Architecture Structure for Pose Estimation. It includes Backbone, Neck, and Head layer

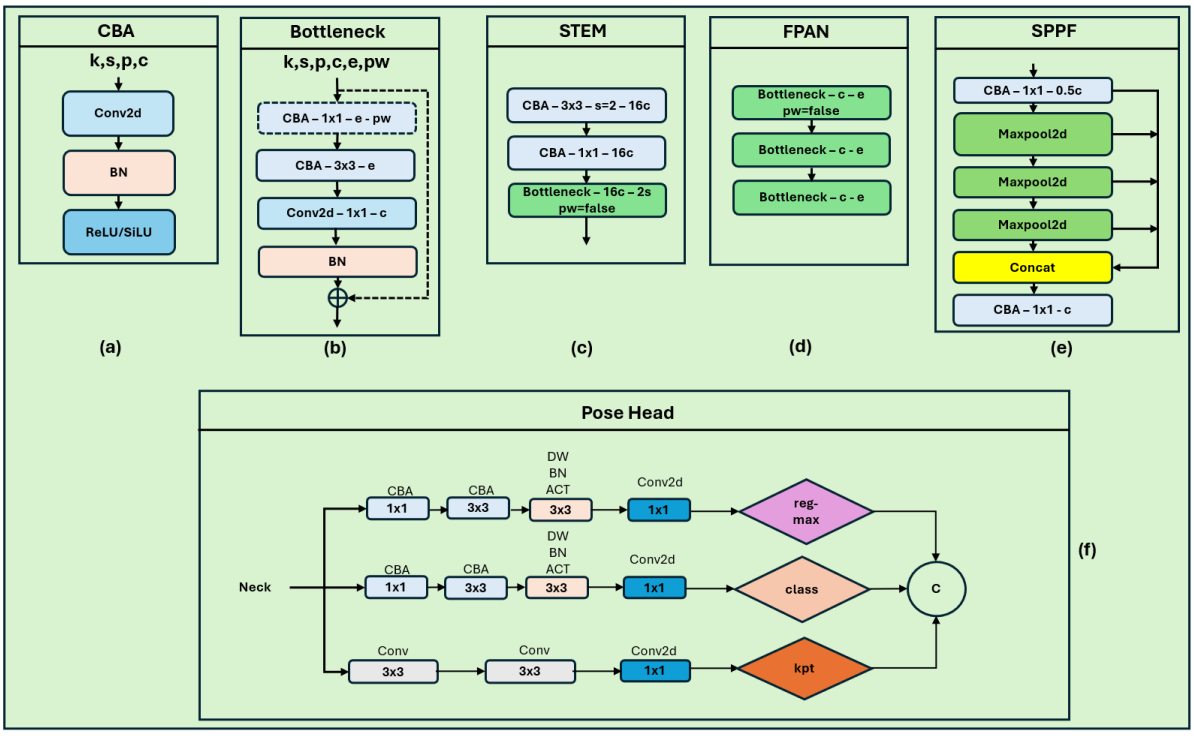

Figure 2. Key Modules in the Optimized LeYOLO-Nano Architecture. (a). CBA Block (Convolution - Batch Normalization - Activation), (b). Bottleneck CSP Module, (c). STEM Block for Initial Feature Extraction, (d). FPAN Module for Multi-Scale Feature Fusion, (e). SPPF Block for Capturing Spatial Context, (f). Pose Estimation Head for Predicting 17 Keypoints

III. Results and Discussion

This part provides evaluation results LeYOLO-Nano baseline architecture for human pose estimation using the MSCOCO 2017 dataset. Qualitatively, the results show that LeYOLO-Nano is capable of estimating human poses with reasonable accuracy. The model successfully detects key body points such as the shoulders, hips, ankles, elbows, head even under challenging conditions such as occlusion, overlapping individuals, and complex poses [22]. This ability indicates the model's robustness in handling various real-world scenarios involving multiple subjects and diverse body orientations. The

visual results suggest that, even without attention modules or other architectural enhancements, the baseline model maintains strong spatial awareness and contextual understanding. The precision of keypoint placement appears stable with minimal spatial jitter, especially at major joints like the shoulders and hips, which are essential for downstream applications such as action recognition or pose-based interactive systems. The model's performance was quantitatively assessed using the MSCOCO validation data on a standard CPU configuration, with results are displayed in Table 3. This is shown presents the performance evaluation results of the LeYOLO-Nano architecture in the object detection task. Based on the table, the model has an mAP50 value of 0.69, which indicates a fairly good accuracy in detecting objects with an Intersection over Union (IoU) [27] threshold of mAP0.5. Meanwhile, mAP50:95 have value 0.362 reflects the model's performance in various levels of detection difficulty, ranging from IoU 0.5 to 0.95, which gives a more rigorous and comprehensive picture of accuracy. In terms of complexity, the model only has about 1.254 million parameters, which is very lightweight compared to other object detection models. In addition, this model only requires about 3.2 GFLOPs in one inference, making it computationally efficient and suitable for application on edge devices with limited resources. The model's performance in real-time is also evident from the FPS (CPU) value of 20.78, which shows that LeYOLO-Nano is able to perform the detection process at a fairly high speed even when run on a CPU without GPU support.

A. Training Configuration

Table I outlines the training setup applied in the LeYOLO-Nano model experiment for human pose estimation. The model was trained for 100 epochs using a batch size of 32. All input images were uniformly resized to 640×640 pixels to maintain consistent input dimensions and optimize computational performance. The learning rate used was 0.01, which is a common initial value for lightweight architectures. The optimizer used was Stochastic Gradient Descent (SGD), which is known to be effective for training convolutional models on large-scale datasets. In Table II, testing and training were conducted using the following hardware and software configuration: Operating System (Ubuntu): A Linux-based operating system widely used for deep learning development due to its stability and compatibility with frameworks like PyTorch. Framework (PyTorch 2.0): PyTorch version 2.0 was used for its performance improvements and flexible API, which supports the implementation of YOLO and its derivatives. Python Version (3.9.20): A stable version of Python compatible with PyTorch dependencies and other libraries used in this project. Training CPU (AMD Ryzen 5 4500): Used for preprocessing and light computations during the training process. Training using GPU (RTX 4060 Ti 16GB): This GPU was utilized to accelerate the model training process by providing high parallel computing capabilities, especially during backpropagation and batch data processing. The use of a GPU allows training time to be more efficient compared to using a CPU, enabling the model to be developed and refined more quickly.

TABLE I
TRAINING CONFIGURATION

Metric	Value
Epoch	100
Batch size	32
Images size	640x640
Learning rate	0.01
Optimizer	SGD

TABLE II
SOFTWARE AND HARDWARE

Component	Spesification
Operating system	Ubuntu
Framework	PyTorch 2.0
Python Version	3.9.20
CPU (Training)	AMD Ryzen 5 4500 6-core
Gpu (Training)	RTX 4060 Ti 16GB

The 16GB memory capacity also offers flexibility in handling more complex models or larger datasets.

Inference using CPU (Intel i7 12700 20-core): Inference testing was conducted using this CPU to evaluate the model performance for run to devices without GPU acceleration. This process simulates real-world scenarios in which the model is deployed on edge devices or embedded systems, such as robots, smart cameras, or IoT devices. Using a 20-core CPU provides a realistic picture of processing speed and model efficiency under limited computational conditions, which is crucial for real-time and energy-efficient applications.

B. Dataset

The MSCOCO 2017 dataset is used to train and evaluate LeYOLO-Nano for human pose estimation [23]. It contains diverse real-world conditions such as varying lighting, occlusion, and different body orientations. A total of 118,287 images were utilized for the training phase, and 5,000 images were set aside for validation to assess the model's ability to generalize. The dataset includes 17 annotated keypoints per person, making it well-suited for human pose estimation tasks evaluation on Dataset

Figure 3 ilustrates the performance evaluation results the model LeYOLO-Nano tested on the validation subset of the 2017 MS COCO dataset. In this figure, the results of human detection and pose estimation are presented with red bounding boxes and colored keypoints that indicate body parts like shoulders, knees, ankles, elbows, and head. This visualization demonstrates the model's ability to recognize and map human poses in various complex activities, such as surfing, playing soccer, skiing, and walking in crowded environments. These activities feature challenges such as extreme poses, body overlaps, and partially or fully hidden body parts, which test the model's robustness and accuracy in non-ideal real-world conditions.

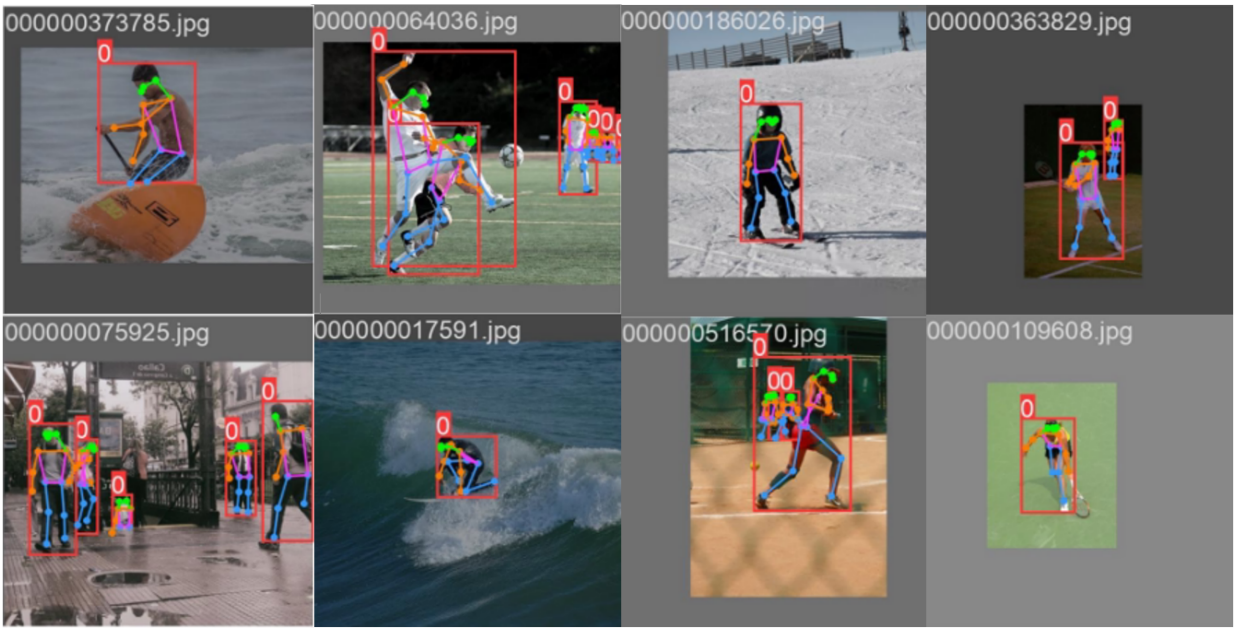

Figure 3. Pose estimation result of LeYOLO-nano on MSCOCO dataset

TABLE III
COMPARISON OF MODELS PERFORMANCE

Metric	LeYOLO	MobileNetV1	ShuffleNet
mAP50	0.69	0.538	0.679
mAP50;95	0.362	0.235	0.35
Params	1.254.076	456.814	1.106.334
GFlops	3.2	1.67	3.24
FPS (CPU)	20.78	39.60	26.25

The results also demonstrate the model's capability to sustain detection accuracy despite changes in lighting, camera angles, and object scales. In general, the LeYOLO-Nano model is still able to detect human presence and perform pose estimation; however, several limitations are observed under difficult conditions. In some cases, such as surfing activities or back-facing poses, the estimation of body parts like hands, feet, and face is not fully accurate. This also occurs when parts of the body are occluded by other objects or when the body orientation is not directly facing the camera. Furthermore, in images with small objects or poor lighting, the pose estimation results appear less precise. Therefore, the results in Figure 3 can serve as a reference that the LeYOLO-Nano model still has weaknesses in handling complex poses and occlusion, images (with file names shown above each image). Red bounding box indicates the location of detected human presence. Label 0: the class label for humans. Skeletal Keypoints (colored dots): Represent the estimated pose of human body parts, with connecting lines forming the pose skeleton.

Based on Table III, YOLO-Nano demonstrates fairly good performance in object detection tasks by an mAP50 the value 0.69 and an mAP50:95 the value 0.362. Additionally, model parameter count of only 1.254.076 indicates that it is very lightweight, requiring minimal memory resources to operate.

In terms of computational efficiency, the requirement of 3.2 GFLOPs suggests that YOLO-Nano can run with low processing demands, making it ideal for edge devices or embedded systems. With only the CPU, this model is capable of processing up to 20.78 frames per second, which shows a performance that is fast enough for the needs of real-time applications. This makes this model a practical and efficient solution for systems with hardware limitations.

IV. CONCLUSIONS

This research conducted an initial evaluation of the LeYOLO-Nano architecture for 2D human pose estimation using the MS COCO 2017 validation set, without incorporating additional modules such as attention or feature fusion, to serve as a baseline for future studies [24], [23],[25]. The model achieved a mAP50 of 0.69 and a speed of 20.78 FPS on a CPU, indicating its potential for real-time applications like video surveillance and human-computer interaction [24],[25]. However, the model's performance declines in challenging conditions such as occlusion, extreme body poses, and unusual orientations [23],[25], which supports the research objective of identifying current limitations in pose estimation. To address these challenges, future work should consider integrating attention mechanisms like CBAM, SE Block, or GCNet to enhance spatial focus and robustness [26],[27],[28],[29],[11]. Additionally, a hybrid modular approach combining appearance, contextual, and semantic features could improve accuracy and generalization without significant computational costs.

ACKNOWLEDGEMENTS

The authors extend their heartfelt gratitude to the AIVISION research team for their invaluable support and collaboration throughout the course of this study. Their technical knowledge and provision of computational resources were crucial to this work, and their insightful feedback greatly contributed to

refining both the research methodology and the final manuscript.

REFERENCES

[1] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," May 30, 2019, *arXiv*: arXiv:1812.08008. doi: 10.48550/arXiv.1812.08008.

[2] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, and C. Yang, "The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation," *IEEE Access*, vol. 8, pp. 133330–133348, 2020, doi: 10.1109/ACCESS.2020.3010248.

[3] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Jun. 03, 2021, *arXiv*: arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929.

[4] J. Hwang, J. Yang, and N. Kwak, "Exploring Rare Pose in Human Pose Estimation," *IEEE Access*, vol. 8, pp. 194964–194977, 2020, doi: 10.1109/ACCESS.2020.3033531.

[5] Y. Liu, "OpenPose-Based Yoga Pose Classification Using Convolutional Neural Network," *Highlights Sci. Eng. Technol.*, vol. 23, pp. 72–76, Dec. 2022, doi: 10.54097/hset.v23i.3130.

[6] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local Neural Networks," Apr. 13, 2018, *arXiv*: arXiv:1711.07971. doi: 10.48550/arXiv.1711.07971.

[7] M. D. Putro, D.-L. Nguyen, and K.-H. Jo, "A Fast CPU Real-Time Facial Expression Detector Using Sequential Attention Network for Human–Robot Interaction," *IEEE Trans. Ind. Inform.*, vol. 18, no. 11, pp. 7665–7674, Nov. 2022, doi: 10.1109/TII.2022.3145862.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," May 09, 2016, *arXiv*: arXiv:1506.02640. doi: 10.48550/arXiv.1506.02640.

[9] L. Hollard, L. Mohimont, N. Gaveau, and L. A. Steffenel, "LeYOLO, New Embedded Architecture for Object Detection," *Proc. Conf. Robots Vis.*, May 2025, doi: 10.21428/d82e957c.aed2cb06.

[10] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss," Apr. 14, 2022, *arXiv*: arXiv:2204.06806. doi: 10.48550/arXiv.2204.06806.

[11] F. Wang, G. Wang, and B. Lu, "YOLOv8-PoseBoost: Advancements in Multimodal Robot Pose Keypoint Detection," *Electronics*, vol. 13, no. 6, Art. no. 6, Jan. 2024, doi: 10.3390/electronics13061046.

[12] A. Arif, Y. Yasin Ghadi, M. Alarfaj, A. Jalal, S. Kamal, and D.-S. Kim, "Human Pose Estimation and Object Interaction for Sports Behaviour," *Comput. Mater. Contin.*, vol. 72, no. 1, pp. 1–18, 2022, doi: 10.32604/cmc.2022.023553.

[13] Y. Jiang, K. Yang, J. Zhu, and L. Qin, "YOLO-Rlepose: Improved YOLO Based on Swin Transformer and Rle-Oks Loss for Multi-Person Pose Estimation," *Electronics*, vol. 13, no. 3, Art. no. 3, Jan. 2024, doi: 10.3390/electronics13030563.

[14] C.-J. Chou, J.-T. Chien, and H.-T. Chen, "Self Adversarial Training for Human Pose Estimation," Aug. 15, 2017, *arXiv*: arXiv:1707.02439. doi: 10.48550/arXiv.1707.02439.

[15] J. Ou and H. Wu, "Efficient Human Pose Estimation with Depthwise Separable Convolution and Person Centroid Guided Joint Grouping," Dec. 06, 2020, *arXiv*: arXiv:2012.03316. doi: 10.48550/arXiv.2012.03316.

[16] J. Ding, S. Niu, Z. Nie, and W. Zhu, "Research on Human Posture Estimation Algorithm Based on YOLO-Pose," *Sensors*, vol. 24, no. 10, Art. no. 10, Jan. 2024, doi: 10.3390/s24103036.

[17] Y. Li, X. Wang, W. Liu, and B. Feng, "Pose Anchor: A Single-Stage Hand Keypoint Detection Network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2104–2113, Jul. 2020, doi: 10.1109/TCSVT.2019.2912620.

[18] X. Zhang, D. Zhang, J. Ge, K. Hu, L. Yang, and P. Chen, "Multi-stage Real-time Human Head Pose Estimation," in *2019 6th International Conference on Systems and Informatics (ICSAI)*, Nov. 2019, pp. 563–567. doi: 10.1109/ICSAI48974.2019.9010492.

[19] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, *A Guide to Convolutional Neural Networks for Computer Vision*. in Synthesis Lectures on Computer Vision. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-031-01821-3.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 936–944. doi: 10.1109/CVPR.2017.106.

[21] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8759–8768. doi: 10.1109/CVPR.2018.00913.

[22] A. Arif, Y. Yasin Ghadi, M. Alarfaj, A. Jalal, S. Kamal, and D.-S. Kim, "Human Pose Estimation and Object Interaction for Sports Behaviour," *Comput. Mater. Contin.*, vol. 72, no. 1, pp. 1–18, 2022, doi: 10.32604/cmc.2022.023553.

[23] "COCO - Common Objects in Context." Accessed: Jun. 06, 2025. [Online]. Available: https://cocodataset.org/#home

[24] L. Hollard, L. Mohimont, N. Gaveau, and L. A. Steffenel, "LeYOLO, New Embedded Architecture for Object Detection," *Proc. Conf. Robots Vis.*, May 2025, doi: 10.21428/d82e957c.aed2cb06.

[25] J. Ding, S. Niu, Z. Nie, and W. Zhu, "Research on Human Posture Estimation Algorithm Based on YOLO-Pose," *Sensors*, vol. 24, no. 10, Art. no. 10, Jan. 2024, doi: 10.3390/s24103036.

[26] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," Jul. 18, 2018, *arXiv*: arXiv:1807.06521. doi: 10.48550/arXiv.1807.06521.

[27] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," May 16, 2019, *arXiv*: arXiv:1709.01507. doi: 10.48550/arXiv.1709.01507.

[28] Q. Hou, D. Zhou, and J. Feng, "Coordinate Attention for Efficient Mobile Network Design," Mar. 04, 2021, *arXiv*: arXiv:2103.02907. doi: 10.48550/arXiv.2103.02907.

[29] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," Apr. 07, 2020, *arXiv*: arXiv:1910.03151. doi: 10.48550/arXiv.1910.03151.

Vicky Nolant Setyanto Lahimade, born in Lehupu on November 15, 1995, is currently pursuing postgraduate studies in the Informatics Program at Sam Ratulangi University. He graduated from the Department of Electrical Engineering, Faculty of Engineering, Sam Ratulangi University. During his studies, he actively participated in various academic activities, including small researches and system building using microcontrollers, with main interests in the field of control systems and embedded systems. Driven by a passion for technology and intelligent systems, the author continued his academic journey at the graduate level, concentrating on artificial intelligence and computer vision. His ongoing research, titled "Human Pose Estimation Using LeYOLO-Nano Architecture," investigates the application of compact deep learning models for real-time pose detection on hardware with constrained computing capabilities. With a strong interest in integrating hardware and software, the author's expertise includes robotics, embedded system programming, and the implementation of artificial intelligence algorithms for real-world applications such as object tracking and human activity recognition. The author can be contacted via email at vickylahimade026@student.unsrat.ac.id.