Indonesian Journal of Data and Science

Volume 6 Issue 2
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v6i2.228

*Research Article*

# Yoga Posture Recognition and Classification Using YOLOv5

**Afwatul Maqbullah [1], Anik Nur Handayani [2,*]**

[1] Universitas Negeri Malang, Malang, Indonesia, afwatul.maqbullah.2105366@students.um.ac.id
[2] Universitas Negeri Malang, Malang, Indonesia, aniknur.ft@um.ac.id

Correspondence should be addressed to Anik Nur Handayani; aniknur.ft@um.ac.id

Received 05 November 2024; Accepted 28 May 2025; Published 31 July 2025



**Abstract:**

Yoga, a centuries-old health practice from India, has gained global recognition for its benefits to physical, mental, and emotional well-being. However, incorrect execution of yoga poses can lead to injuries or diminished results. This research develops an automated system for recognizing and classifying yoga postures using YOLOv5, a state-of-the-art deep learning algorithm. YOLOv5, part of the YOLO (You Only Look Once) series, is designed for real-time object detection and offers enhanced performance through features like anchor-free detection and adaptive training strategies. The study collects a dataset of 1,000 images across 20 yoga pose categories, followed by manual annotation and training using transfer learning. Validation results show strong performance, achieving an accuracy of 90% with precision and recall scores of 0.942 and 0.941, respectively, and mAP@50 and mAP@50-95 values of 0.976 and 0.866. Despite challenges with certain poses showing lower accuracy due to variations in posture and dataset limitations, the model demonstrates robustness in detecting and classifying yoga postures effectively. This system has potential applications in artificial intelligence-driven yoga education, enabling practitioners to train independently with real-time feedback.

**Keywords:** Artificial Intelligence; Deep Learning; Pose Classification; Yoga Posture Recognition; YOLOv5.
**Dataset link:** https://www.kaggle.com/datasets/tr1gg3rtrash/yoga-posture-dataset

## 1. Introduction

YOGA, a holistic health practice that originated in India thousands of years ago [1], has become increasingly popular globally due to its wide range of benefits for physical, mental, and emotional well-being [2], [3]. This ancient practice involves a series of poses or postures known as asanas, each designed to achieve specific objectives. For instance, some asanas aim to improve flexibility and strength, while others focus on correcting body posture, enhancing balance, reducing stress, or boosting mental concentration [4]. Despite its many advantages, performing yoga poses incorrectly can lead to serious consequences, such as injuries, muscle strain, or reduced effectiveness in achieving the intended benefits [5]. This highlights the importance of proper guidance and feedback during practice. To address this need, developing a system capable of accurately recognizing and classifying yoga postures can be highly beneficial. Such a system would support yoga instructors in providing real-time corrections and assist practitioners in achieving optimal results and injury prevention.

In the era of digital transformation, the development of artificial intelligence (AI) technology, especially in the field of computer vision, provides new opportunities to improve the yoga learning experience [6]. One notable algorithm in this field is YOLO [7], which excels at detecting objects in images and videos in real-time with remarkable precision [8], [9]. Among its versions, YOLOv5 stands out for its exceptional speed and accuracy. These qualities position YOLOv5 as a promising tool for creating AI-based systems to classify and recognize yoga postures effectively.

In this context, the research focuses on the development of a YOLOv5-based system capable of detecting and classifying various yoga postures automatically and with high accuracy. Furthermore, it evaluates the performance of YOLOv5 in identifying and classifying different yoga poses, its capability to handle dataset variations, and its feasibility for integration into AI-powered devices with limited computational resources. By employing this system, individuals are enabled to practice yoga independently with enhanced safety and efficiency, minimizing the risk of injuries due to incorrect postures while improving the overall quality of their yoga practice [10].

This research provides a valuable contribution to health technology, particularly in advancing yoga education. By implementing a system capable of delivering automatic feedback, individuals can practice yoga more safely and effectively [11]. The integration of AI-driven solutions not only addresses the challenge of limited instructor access but also ensures that users can achieve proper form, reducing the risk of injury. Furthermore, the outcomes of this research have broader applications in fitness training platforms, health devices, and other AI-driven solutions, potentially advancing human activity analysis in fields such as sports and healthcare.

Research has demonstrated that YOLOv5 excels in object detection with notable speed and accuracy [12], [13], but its application in yoga posture recognition remains underexplored, representing a critical research gap. Compared to earlier models such as R-CNN, YOLOv5 offers a more efficient and resource-friendly solution for posture detection. The growing integration of AI in fitness highlights the need for models capable of robust performance on devices with limited computational resources, making YOLOv5 an ideal candidate for advancing the effectiveness and accessibility of AI-powered yoga solutions.

The study acknowledges several limitations, including its reliance on high-quality datasets and the challenge of generalizing the model to unfamiliar poses. Despite these challenges, it provides a unique perspective by integrating object detection technology with the traditional health practice of yoga. This approach bridges existing knowledge gaps and offers practical advantages, such as broader accessibility to quality yoga training and reduced risk of injuries caused by incorrect postures.

Built on comprehensive literature insights and current trend analysis, this research aims to contribute meaningfully to AI development and public health by promoting safer, more accessible, and effective yoga practices.

## 2. Method:

This study employs a deep learning approach to classify yoga postures using the YOLOv5 algorithm. The research methodology encompasses seven key stages, as illustrated in **Figure 1**.

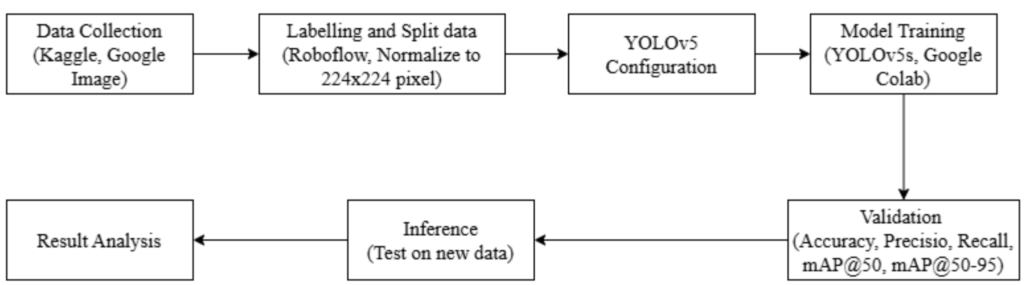



**Figure 1.** Research Flow Diagram

The initial stage involves the collection of datasets from diverse sources, including public repositories such as Kaggle and Google Images. The second stage focuses on dataset preparation and annotation using the Roboflow platform. The third stage involves configuring YOLOv5 by defining essential parameters such as image resolution, the number of training epochs, and the division of the dataset into training, validation, and test sets. The fourth stage is the training phase, carried out on Google Colab utilizing GPU resources to expedite the process. The fifth stage involves validating the trained model using the designated validation dataset. The subsequent stage, inference, tests the model's capacity to detect and classify new yoga postures, ensuring its reliability in practical applications. The final stage focuses on analyzing the evaluation results to identify the model's strengths and limitations.

### Data Collection

The initial stage involves the collection of datasets from diverse sources, including public repositories such as Kaggle and Google Images. This step ensures dataset variability, essential for training a robust model capable of handling differences in postures, lighting conditions, and image perspectives [14].

The dataset consists of 20 yoga postures: *Adho Mukha Svanasana*, *Adho Mukha Vrksasana*, *Anjaneyasana*, *Ardha Chandrasana*, *Ardha Pincha Mayurasana*, *Baddha Konasana*, *Bakasana*, *Balasana*, *Bitilasana*, *Camatkarasana*, *Dhanurasana*, *Eka Pada Rajakapotasana*, *Garudasana*, *Halasana*, *Hanumanasana*, *Malasana*, *Marjaryasana*, *Padmasana*, *Setu Bandha Sarvangasana*, and *Ustrasana*, with each posture represented by 50 images. In total, the dataset comprises 1,000 images.

### Labelling and Split Data

The second stage focuses on dataset preparation and annotation using the Roboflow platform [15]. During this stage, bounding boxes are applied to the images, and labels are assigned to ensure that the dataset is well-structured and ready for training. Accurate annotation is critical in enhancing the model's ability to detect and classify yoga postures effectively [16].

As part of this stage, the images were labeled manually using the Roboflow platform, with additional preprocessing steps such as resizing and normalizing them to 224×224 pixels. The labeling process was performed manually to ensure that each image had accurate annotations [17]. This labeled dataset is a critical foundation for training the model to recognize patterns effectively. During the labeling process, the dataset was divided into training (80%), validation (10%), and test (10%) sets. It plays a crucial role in assessing the model's ability to handle unfamiliar data effectively, ensuring its reliability in real-world scenarios [18].

### YOLOv5 Configuration

The YOLO network requires an initial configuration file and weights during the early stages of the pre-training process [19]. In YOLOv5, critical parameters such as image resolution, training duration, and dataset segmentation for training, validation, and testing are established. Proper configuration optimizes the model for the specific dataset and improves its generalization capabilities.

The YOLOv5 repository was acquired from GitHub, with the dataset configuration file (.yaml) customized to reflect the dataset structure. Model parameters such as image resolution and batch size were configured to enhance training efficiency. Additionally, the YOLOv5 architecture, which consists of three main components as illustrated in **Figure 2** [20], [21], was selected based on the specific requirements of the project, including accuracy and inference speed. The backbone, utilizing Darknet53, extracts basic features from the image. The neck acts as a bridge between the backbone and the head, combining and refining the extracted features. Finally, the head is divided into three branches, each predicting bounding boxes and object classes at different scales [22]. This configuration serves as a critical initial step to ensure the model can adapt effectively to the available data.

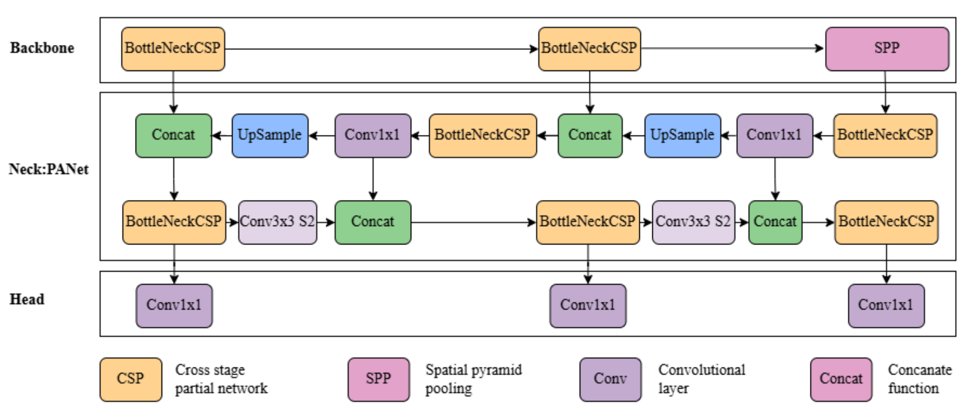



**Figure 2.** YOLOv5 Architecture

### Training

The YOLOv5 model was trained on the Google Colab platform, which provides free access to GPUs significantly accelerating the training process [23], [24]. The GPU acceleration enables efficient fine-tuning of the model's weights over multiple epochs, improving performance and convergence speed.

The dataset, pre-processed in YOLOv5 format via Roboflow, was uploaded to Google Colab for further use. Using the YOLOv5s architecture as a starting point, the transfer learning method utilized pre-trained weights from a prior model. The training lasted 250 epochs, with the image resolution set to 224×224 pixels and a batch size 16, ensuring a balance between computational efficiency and memory management.

### Validation

The model underwent evaluation through a validation phase utilizing a validation set to evaluate how effectively the model can identify and classify yoga postures on data it has not been exposed to during training. Key metrics analyzed in YOLOv5 validation include Precision (P), Recall (R), mean Average Precision at 50% IoU (mAP50), mean Average Precision across IoU thresholds from 50% to 95% (mAP50-95), and accuracy [25]. Precision refers to a model's ability to detect only relevant objects, minimizing false detections of yoga postures. Recall measures how well the model identifies all actual yoga postures by calculating the percentage of correct positive predictions among all ground truth bounding boxes [26].

Mean Average Precision at 50% IoU (mAP50) evaluates the model's detection accuracy by averaging the precision of overall object categories at a fixed IoU threshold of 50%. It provides an initial performance assessment of the detector's capability to distinguish objects from the background. Meanwhile, mAP50-95 extends this evaluation by averaging the precision over multiple IoU thresholds, ranging from 50% to 95% in 5% increments. This metric offers a more comprehensive assessment of the model's robustness by considering its ability to detect objects at varying degrees of overlap with ground truth annotations [27]. The validation results also help identify the model's weaknesses, such as classes with low precision or recall, which can serve as a reference for improvements in subsequent stages. In addition to these metrics, (1) measures accuracy, representing the percentage of correctly identified samples. [28]. To calculate accuracy in this research, the equation below was utilized:

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{True\ Positive\ (TP) + False\ Positive\ (FP) + False\ Negative\ (FN) + True\ Negative\ (TN)} \quad (1)$$

### Interface

The inference stage focuses on detecting yoga postures from new datasets. It assesses the model's accuracy in recognizing and classifying these postures. The model is tested across multiple data variations, including different angles, lighting conditions, and backgrounds. This helps validate its performance and ensure its robustness in real-world scenarios [7].

Additionally, the results include confidence score evaluations. These scores provide insight into the model's certainty in its predictions. This information lays the foundation for optimizing the model's performance in future iterations. These insights support parameter adjustments, augmentation refinements, and architectural modifications to enhance model accuracy and robustness.

### Analysis

The evaluation results are analyzed to determine the strengths and weaknesses of the YOLOv5 model in recognizing and classifying yoga postures. This includes identifying potential misclassifications and areas where the model performs well. The analysis involves calculating accuracy and comparing the model's predictions with the actual labels.

The resulting data provide insights into aspects of the model that require improvement or should be maintained. By analyzing these findings, researchers can identify specific areas where adjustments are needed to enhance performance. These insights play a crucial role in refining the model and ensuring it continues to improve. In future stages, this evaluation will guide optimization efforts, leading to greater accuracy and reliability.

## 3. Results and Discussion

### Results

The results of the YOLOv5 training data evaluation are shown in **Figure 3**. The first row displays training loss for bounding box regression (train/box_loss), object classification (train/obj_loss), and class confidence (train/cls_loss), while the second row shows the corresponding validation losses. The remaining plots present performance metrics, including precision, recall, and mAP at different IoU thresholds. The decreasing loss and improving metrics indicate successful model convergence and enhanced detection performance.

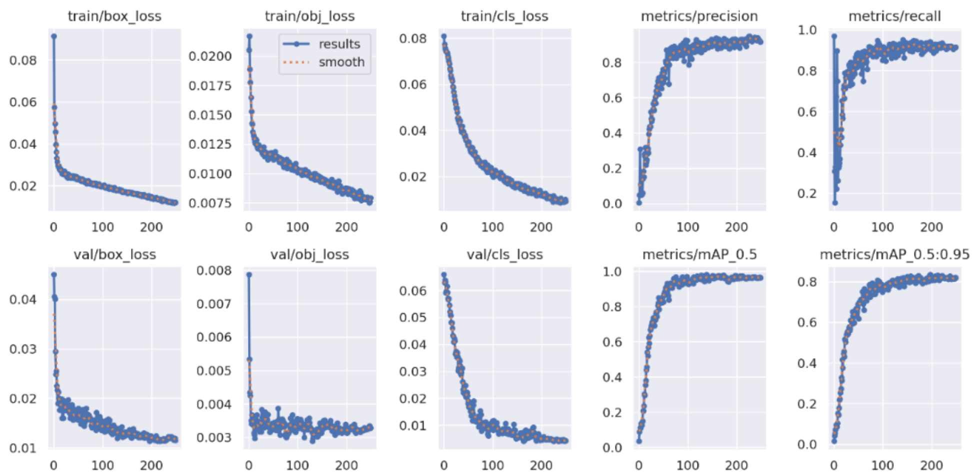
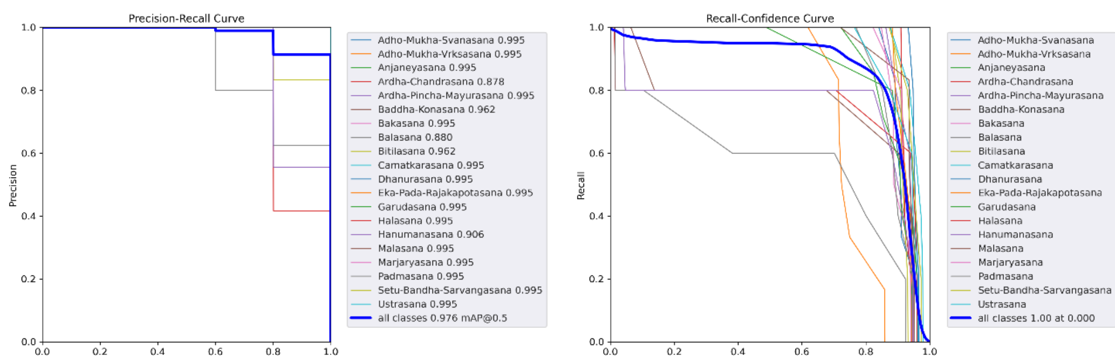



**Figure 3.** Training Data Evaluation Results

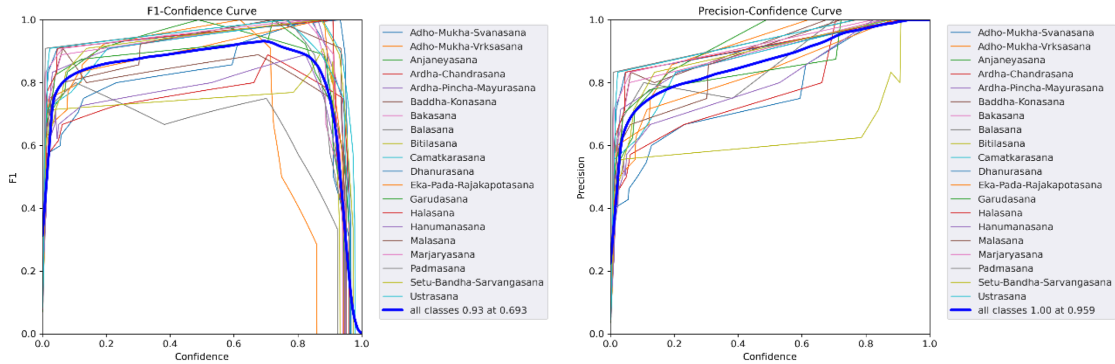
**Figure 4** shows each yoga pose's Precision-Recall Curve (a) and Recall-Confidence Curve (b). The Precision-Recall curve illustrates the relationship between Precision, the accuracy of positive predictions, and Recall, the model's ability to identify all positive instances [29]. The Recall-Confidence curve serves as a key evaluation metric for measuring the YOLOv5 model's efficiency and performance in multi-class object detection [30].




(a) (b)

**Figure 4.** (a) Precision-Recall Curve and (b) Recall-Confidence Curve

**Figure 5** presents the F1-Confidence curve (a) and Precision-Confidence curve (b) for various yoga poses. The F1-Confidence curve serves as a key performance indicator, providing a combined assessment of accuracy and recall in classifying various yoga poses [30]. The Precision-Confidence curve illustrates the importance of setting appropriate confidence thresholds in classification problems [31].




(a) (b)

**Figure 5.** (a) F1-Confidence Curve and (b) Precision-Confidence Curve

The values for each component of the confusion matrix, which contribute to the calculation of accuracy, are displayed in **Figure 6**. The matrix visualizes the model's performance by comparing predicted and actual class labels.

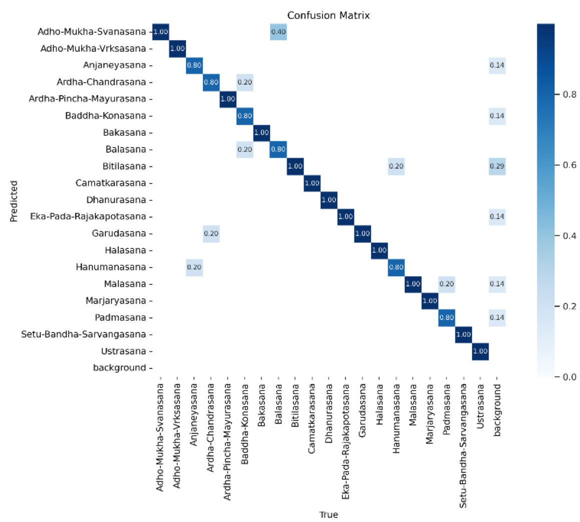



**Figure 6.** Confusion Matrix

Referring to (1), the accuracy formula is applied in this section to evaluate the model's performance.

$$Accuracy = \frac{(18,80)+(0,60)}{(18,80)+(0,99)+(1,00)+(0,60)} = \frac{19,4}{21,39} = 0,90696$$

With an accuracy rate of 0.90696, equivalent to 90%, the result indicates that the model demonstrates strong capability in detecting and classifying various yoga postures with high precision. Furthermore, the validation of the YOLOv5 model, as presented in **Figure 3**, **Figure 4**, **Figure 5**, and summarized in **Table 1**, the yoga posture classification system demonstrates excellent performance with an overall precision (P) of 0.942, recall (R) of 0.941, mAP@50 of 0.976, and mAP@50-95 of 0.866.

**Table 1.** Validation Results

| Class | Precision (P) | Recall (R) | mAP50 | mAP50-95 |
|---|---|---|---|---|
| *Adho Mukha Svanasana* | 0.895 | 1.000 | 0.995 | 0.854 |
| *Adho Mukha Vrksasana* | 1.000 | 0.800 | 0.995 | 0.853 |
| *Anjaneyasana* | 1.000 | 0.807 | 0.995 | 0.895 |

| Class | Precision (P) | Recall (R) | mAP50 | mAP50-95 |
|---|---|---|---|---|
| *Ardha Chandrasana* | 0.846 | 0.807 | 0.878 | 0.830 |
| *Ardha Pincha Mayurasana* | 0.951 | 1.000 | 0.995 | 0.801 |
| *Baddha Konasana* | 0.899 | 0.800 | 0.962 | 0.859 |
| *Bakasana* | 0.955 | 1.000 | 0.995 | 0.821 |
| *Balasana* | 0.957 | 0.800 | 0.880 | 0.823 |
| *Bitilasana* | 0.815 | 1.000 | 0.962 | 0.876 |
| *Camatkarasana* | 0.949 | 1.000 | 0.995 | 0.857 |
| *Dhanurasana* | 0.951 | 1.000 | 0.995 | 0.932 |
| *Eka Pada Rajakapotasana* | 0.914 | 1.000 | 0.995 | 0.956 |
| *Garudasana* | 0.869 | 1.000 | 0.995 | 0.840 |
| *Halasana* | 0.954 | 1.000 | 0.995 | 0.925 |
| *Hanumanasana* | 0.897 | 0.800 | 0.906 | 0.814 |
| *Malasana* | 0.963 | 1.000 | 0.995 | 0.851 |
| *Marjaryasana* | 0.957 | 1.000 | 0.995 | 0.848 |
| *Padmasana* | 0.979 | 1.000 | 0.995 | 0.892 |
| *Setu Bandha Sarvangasana* | 0.954 | 1.000 | 0.995 | 0.811 |
| *Ustrasana* | 0.979 | 1.000 | 0.995 | 0.967 |
| All | 0.942 | 0.941 | 0.976 | 0.866 |

Precision evaluates how accurately the model identifies true positive predictions in comparison to all positive predictions made. As presented in **Table 1**, certain classes, such as *Adho Mukha Vrksasana*, *Anjaneyasana*, and *Baddha Konasana*, achieved a perfect Precision score of 1.000, indicating zero errors in their positive predictions. On the other hand, some classes, like *Ardha Chandrasana*, had a lower Precision value of 0.846, highlighting inaccuracies in some of its predictions. The lower Precision for this class can attributed to variations in posture and inconsistent poses within the dataset, as shown in **Figure 7**. Overall, the average Precision for all classes reached 0.942, demonstrating that the model possesses a high level of accuracy in its predictions.

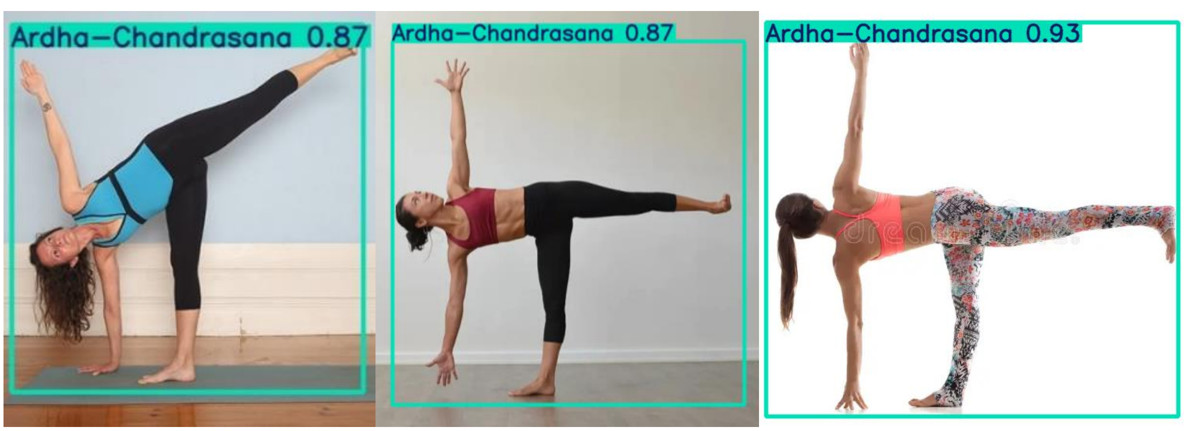



**Figure 7.** *Ardha Chandrasana*

As shown in **Table 1**, several classes, including *Adho Mukha Svanasana*, *Baddha Konasana*, *Bakasana*, and *Balasana*, achieved a perfect Recall score of 1.000. This demonstrates that the model successfully identified all positive instances within these classes without any misses. However, some classes recorded the lowest Recall value of 0.800, indicating that some positive instances were not detected. This discrepancy can be attributed to pose variations or challenging image angles, as seen in the *Adho Mukha Vrksasana* class, illustrated in **Figure 8**. Despite this, the overall average Recall score for all classes reached 0.941, highlighting the model's strong capability in identifying positive instances.


**Figure 8.** *Adho Mukha Vrksasana*

The Mean Average Precision at a 50% Intersection over Union (IoU) threshold (mAP@50) serves as a widely adopted metric for assessing object detection models, particularly in terms of accurately recognizing and localizing objects. In this study, the mAP@50 scores for each yoga posture consistently demonstrated high performance, with most classes achieving values close to or equal to 0.995. The high mAP@50 values, as presented in **Table 1**, indicate that the model possesses exceptional capability in detecting yoga postures, especially when the overlap tolerance between predictions and ground truth is moderately set. This reflects the model's strong generalization ability across the dataset.

At the class level, certain poses such as *Eka Pada Rajakapotasana* and *Ustrasana* achieved very high mAP@50-95 values of 0.956 and 0.967, respectively, highlighting the model's near-perfect ability to identify these poses. Conversely, some poses, including *Adho Mukha Vrksasana* (0.853), *Adho Mukha Svanasana* (0.854), and *Balasana* (0.823), showed relatively lower mAP@50-95 values. These lower scores may be attributed to visual similarities between poses or inconsistencies in the angles of images in the training dataset. Despite this, the mAP@50 values close to 1 indicate that the model can still detect these poses accurately with moderate overlap tolerances.

In **Figure 9**, the detection results for the *Ustrasana* pose are displayed with confidence scores of 0.94 and 0.97. With an impressive mAP@50-95 value of 0.967, the model demonstrates exceptional reliability in accurately identifying and classifying this pose. These detection results demonstrate that the model consistently identifies distinctive features of *Ustrasana*, such as the body's arch and hand positioning, even under varying lighting conditions or backgrounds. The high mAP score underscores that *Ustrasana* is among the poses that are relatively straightforward for the model to detect. This visualized detection supports the claim that the model can be widely implemented in applications requiring accurate recognition of yoga poses.

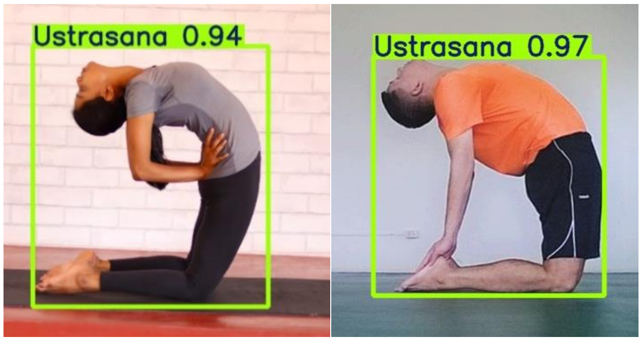



**Figure 9.** *Ustrasana*

In **Figure 10**, the detection results for the *Balasana* pose are presented, with confidence scores of 0.89 and 0.91. This indicates that, although this pose has a relatively lower mAP@50-95 value compared to some other poses, at 0.823, the model is still capable of detecting the pose with a fairly high level of confidence in certain cases. This suggests that despite challenges such as variations in image capture, the model can still deliver relevant and consistent results. With improvements to the dataset or further optimization, such as adding more data variation or fine-tuning training parameters, the accuracy for detecting the *Balasana* pose can be enhanced, thereby improving the overall performance of the system.


**Figure 10.** *Balasana*

Figure 11 illustrates the detection results of yoga postures using the YOLOv5 model, which successfully identifies 20 classes of yoga poses with confidence scores above 0.80 and no labeling errors. Each pose is highlighted with a bounding box and accompanied by a label indicating the name of the posture, as well as a confidence score that reflects the model's level of certainty in the detection.

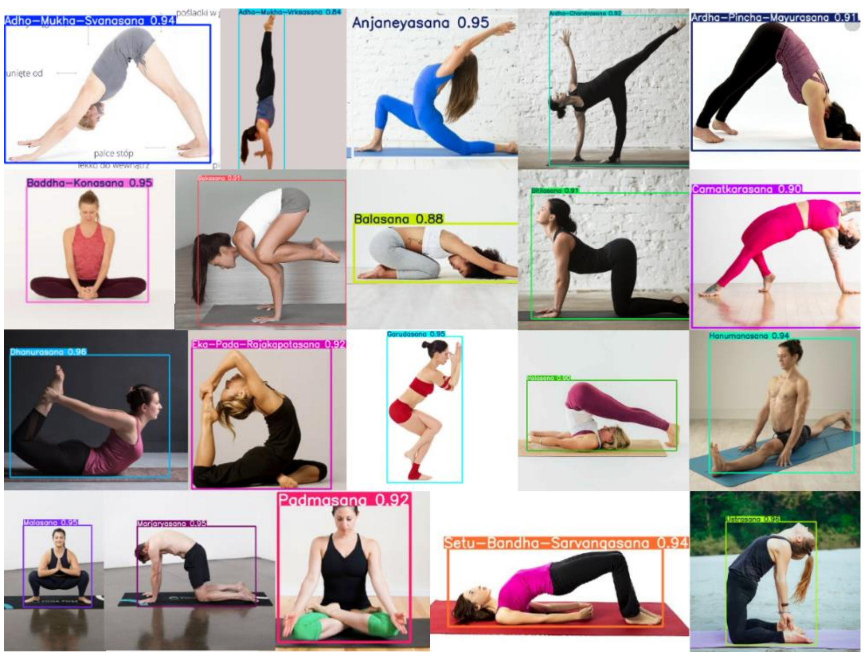



**Figure 11.** Detection results of 20 yoga postures

**Discussion**

In comparison to previous research employing the CNN method, which achieved an average accuracy of 87.85% and a classification accuracy of 99.37% using a smaller dataset consisting of 544 images divided into three yoga classes [32], this study demonstrates the superior capability of YOLOv5 in classifying yoga postures. Using a larger dataset, YOLOv5 attained a higher accuracy rate of 90% and demonstrated flawless labeling with 100% classification accuracy. These results underscore the robustness of YOLOv5 in accurately and efficiently identifying yoga postures, emphasizing its significant potential for integration into real-time systems.

Overall, these results confirm the capability of YOLOv5 to classify yoga postures with high efficiency and accuracy, making it a promising tool for real-time applications. The results demonstrate YOLOv5's robust performance in accurately and efficiently classifying yoga postures, showcasing its potential for integration into real-time systems. Addressing lower accuracy in specific poses may involve diversifying the dataset, adding more varied poses, or optimizing the data augmentation pipeline. Future applications might include AI-driven systems that deliver real-time feedback, empowering yoga practitioners to train independently while minimizing injury risks and improving overall practice quality.

## 4. Conclusion

The YOLOv5-based model demonstrated excellent performance in recognizing and classifying 20 types of yoga postures, achieving an accuracy of 90% along with an average precision of 0.942, recall of 0.941, and mAP@50 and mAP@50-95 scores of 0.976 and 0.866, respectively. These metrics indicate that the model is not only capable of detecting nearly all relevant yoga postures with minimal detection errors but also exhibits high predictive accuracy in effectively classifying the postures. Certain postures, such as *Adho Mukha Vrksasana* and *Balasana*, showed slightly lower mAP scores, possibly due to visual similarities between poses or limitations in the training dataset. These issues could be addressed by incorporating data augmentation and adding greater variation to the dataset. In conclusion, the model proves to be highly effective for detecting and classifying 20 yoga postures, offering efficient and accurate results. It has significant potential to support yoga practitioners in practicing postures independently, providing better feedback, reducing injury risks, and enhancing the quality of their training sessions.

## References:


[1] D. Kumar and A. Sinha, "Yoga Pose Detection and Classification Using Deep Learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3307, pp. 160–184, 2020, doi: 10.32628/cseit206623.

[2] S. K. Yadav, A. Agarwal, A. Kumar, K. Tiwari, H. M. Pandey, and S. A. Akbar, "YogNet: A two-stream network for realtime multiperson yoga action recognition and posture correction," *Knowl Based Syst*, vol. 250, p. 109097, 2022, doi: 10.1016/j.knosys.2022.109097.

[3] S. W. Mohammed, V. Garrapally, S. Manchala, S. N. Reddy, and S. K. Naligenti, "Recognition of Yoga Asana from Real-Time Videos using Blaze-pose," *International Journal of Computing and Digital Systems*, vol. 13, no. 1, pp. 1295–1304, 2022, doi: 10.12785/ijcds/1201104.

[4] A. Sharma, Y. Shah, Y. Agrawal, and P. Jain, "Real-Time Recognition of Yoga Poses Using Computer Vision for Smart Health Care a Preprint,", 2022.

[5] G. S. Irani, "Injuries / Harms Resulting from Incorrect Adjustments / Alignments Performed by Yoga Asana Practitioners," *Arch Pharm Pract*, vol. 11, no. 3, pp. 38–47, 2020.

[6] R. Gajbhiye, S. Jarag, P. Gaikwad, and S. Koparde, "AI Human Pose Estimation: Yoga Pose Detection and Correction," *Int J Innov Sci Res Technol*, vol. 7, no. 5, pp. 1649–1658, 2022.

[7] R. Wulanningrum, A. N. Handayani, and H. W. Herwanto, "Comparative Analysis of Yolov-8 Segmentation for Gait Performance in Individuals with Lower Limb Disabilities," *International Journal of Robotics and Control Systems*, vol. 5, no. 1, pp. 516–529, 2025, doi: 10.31763/ijrcs.v5i1.1731.

[8] B. Mahaur and K. K. Mishra, "Small-object detection based on YOLOv5 in autonomous driving systems," *Pattern Recognit Lett*, vol. 168, pp. 115–122, 2023, doi: 10.1016/j.patrec.2023.03.009.

[9] R. A. Asmara *et al.*, "YOLO-based object detection performance evaluation for automatic target aimbot in first-person shooter games," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 4, pp. 2456–2470, Aug. 2024, doi: 10.11591/eei.v13i4.6895.

[10] P. Verma and R. Sharma, "Enhancing Yoga Practice through Real-time Posture Detection and Correction using Artificial Intelligence : A comprehensive Review," *NeuroQuantology*, vol. 21, no. 6, pp. 1053–1059, 2023, doi: 10.48047/nq.2023.21.6.NQ23111.

[11] A. Upadhyay, N. K. Basha, and B. Ananthakrishnan, "Deep Learning-Based Yoga Posture Recognition Using the Y_PN-MSSD Model for Yoga Practitioners," *Healthcare*, vol. 11, no. 4, pp. 1–19, 2023, doi: 10.3390/healthcare11040609.

[12] S. K. Jaiswal and R. Agrawal, "A Comprehensive Review of YOLOv5: Advances in Real-Time Object Detection," *International Journal of Innovative Research in Computer Science and Technology*, vol. 12, no. 3, pp. 75–80, 2024, doi: 10.55524/ijircst.2024.12.3.12.

[13] V. Karnakar, "Object Detection in Autonomous Vehicle Using Yolov5," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 5, no. 10, pp. 3114–3119, 2023, doi: 10.56726/IRJMETS45668.

[14] A. K. Rajendran and S. C. Sethuraman, "Transfer Learning Based Yogic Posture Recognition System Using Deep Pre-trained Features," *SN Comput Sci*, vol. 5, no. 6, 2024, doi: 10.1007/s42979-024-03086-8.

[15] M. H. Hamir *et al.*, "Retail Product Object Detection using YOLOv5 for Automatic Checkout System in Smart Retail Environment," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 2, no. 2, pp. 182–195, 2026.

[16] A. P. D, "A Comparative Analysis of Object Identification Labelling Platforms: Basketball Perspective," *International Journal of Media and Networks*, vol. 2, no. 3, pp. 01–04, 2024, doi: 10.33140/ijmn.02.03.02.

[17] Q. Huang, L. Yang, H. Huang, T. Wu, and D. Lin, "Caption-Supervised Face Recognition: Training a State-of-the-Art Face Model Without Manual Annotation," *European Conference on Computer Vision*, vol. 12362, pp. 139–155, 2020, doi: 10.1007/978-3-030-58520-4_9.

[18] A. Zaidan, R. Fatrisna, A. Nur, and A. Prasetya, "Decision tree based algorithms for Indonesian Language Sign System (SIBI) recognition," *Applied Engineering adn Technology*, vol. 3, no. 2, pp. 86–101, 2024.

[19] A. N. Handayani, F. A. Pusparani, D. Lestari, I. M. Wirawan, A. P. Wibawa, and O. Fukuda, "Real-Time Obstacle Detection for Unmanned Surface Vehicle Maneuver," *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 765–779, 2023, doi: 10.31763/ijrcs.v3i4.1147.

[20] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, pp. 1–17, 2021, doi: 10.3390/f12020217.

[21] T. F. Dima and M. E. Ahmed, "Using YOLOv5 Algorithm to Detect and Recognize American Sign Language," *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, pp. 603–607, 2021, doi: 10.1109/ICIT52682.2021.9491672.

[22] R. Khanam and M. Hussain, "What is YOLOv5: A deep look into the internal features of the popular object detector," *arXiv preprint arXiv:2407.20892.*, pp. 3–10, 2024.

[23] S. Ray, K. Alshouiliy, and D. P. Agrawal, "Dimensionality reduction for human activity recognition using google colab," *Information*, vol. 12, no. 1, pp. 1–23, 2021, doi: 10.3390/info12010006.

[24] M. G. Naftali, J. S. Sulistyawan, and K. Julian, "Comparison of Object Detection Algorithms for Street-level Objects," *arXiv preprint arXiv:2208.11315*, 2022.

[25] R. Hakani *et al.*, "Optimizing UAV Detection Performance with YOLOv5 Series Algorithms," *International Journal of Microsystems and IoT*, vol. 2, no. 7, pp. 991–995, 2024.

[26] R. Padilla, S. L. Netto, and E. A. B. Da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," *International Conference on Systems, Signals, and Image Processing*, vol. 2020-July, no. July, pp. 237–242, 2020, doi: 10.1109/IWSSIP48289.2020.9145130.

[27] A. Soumya, C. K. Mohan, and L. R. Cenkeramaddi, "High Precision Single Shot Object Detection in Automotive Scenarios," *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 2, no. Visigrapp, pp. 604–611, 2024, doi: 10.5220/0012383100003660.

[28] Ž. Vujović, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.

[29] J. Miao and W. Zhu, "Precision–recall curve (PRC) classification trees," *Evol Intell*, vol. 15, no. 3, pp. 1545–1569, 2022, doi: 10.1007/s12065-021-00565-2.

[30] X. Wu and T. Li, "A deep learning-based car accident detection approach in video-based traffic surveillance," *Journal of Optics*, vol. 53, no. 4, pp. 3383–3391, 2024, doi: 10.1007/s12596-023-01581-4.

[31] D. F. Santos, "Advancing Skin Cancer Detection: Harnessing the Power of CNNs," *Res Sq*, 2024, doi: 10.21203/rs.3.rs-4137983/v1.

[32] W. Yang, "A Joint Network Based CNN for Yoga Pose Classification and Scoring," *Highlights in Science, Engineering and Technology*, vol. 23, pp. 161–167, 2022, doi: 10.54097/hset.v23i.3218.