Real-Time Video-Based Posture Monitoring Measurement of Back Angles Using YOLOv8 and Edge Detection for Strength Training

Pei-Hsuan Hsieh*, Hyun Sub Choi, Hung-Yu Hsu

Department of Computer Science, National Chengchi University, Taipei, Taiwan, ROC

Received 04 December 2025; received in revised form 29 January 2026; accepted 30 January 2026

DOI: https://doi.org/10.46604/ijeti.2026.15942

Abstract

Workout-related lower back injuries are common during strength training and are often caused by improper posture, highlighting the need for real-time posture monitoring to support injury prevention and performance optimization. This study proposes a mobile-based computer vision approach for real-time quantification of back angles during workouts. The proposed method integrates YOLOv8 instance segmentation to isolate the trunk region and applies Canny edge detection for contour extraction. It then employs a geometric formulation to identify neck and back reference points for angle computation. This hybrid design enables robust trunk localization and stable angle estimation across dynamic exercise movements. The model is evaluated on six gym-recorded videos captured with a low-cost mobile camera, achieving a mean relative error of 6.49%, comparable to video-based biomechanical assessment methods. These findings indicate that the proposed approach provides an efficient and practical solution for real-time back-posture monitoring using mobile devices, supporting safer and higher-quality daily training.

Keywords: computer vision, YOLOv8, back angle measurement, strength training, sports injury prevention

1. Introduction

Workout injuries are common among both professional and amateur athletes and are often caused by improper form or posture. Severe injuries can impair performance or even end an athlete's career, with lower back pain being especially prevalent. Maintaining proper lumbar posture during training is therefore critical [1]. Even a two-degree deviation from optimal alignment can increase stress on the posterior spine by approximately 16%. This risk is particularly pronounced for athletes lifting heavy loads (60 - 80% of their one-repetition maximum), underscoring the importance of avoiding excessive spinal curvature. A study demonstrated that athletes with both good body posture (e.g., shoulder slope, scapula protrusion, and pelvic tilt) and good movement patterns experienced significantly fewer injuries [1]. However, no direct interaction was found between body posture and movement patterns, indicating that they influence injury risks independently. The study concludes that while both good body posture and high-quality movement patterns contribute to lowering injury frequency, they should be addressed separately in injury prevention strategies.

Another study proposed an algorithm that employed the OpenPose skeleton detection technique to identify key body points (head, neck, chest, and hips) from side-view images and classify three sitting postures (i.e., good posture, text neck posture, and L-posture during desktop computer work) [2]. The classification was achieved by calculating three angles (head-neck, neck-chest, and chest-hip) and applying weighted threshold values to improve accuracy, with the head-neck angle serving as the most critical factor. Experimental results demonstrated an overall classification accuracy of 97.06%. The study further

* Corresponding author. E-mail address: hsiehph@nccu.edu.tw

highlighted that the algorithm is compatible with both professional cameras and smartphone images, underscoring its accessibility and flexibility for practical implementation. These findings highlight the importance of focusing on both posture correction and movement quality improvement in amateur athletes to reduce injury risks effectively.

Traditional posture monitoring methods, such as mirrors or personal trainers, are limited in both accuracy and accessibility. Nowadays, advanced technological interventions in prior studies have explored different strategies to conduct pose estimation and detect high-speed motion, such as geometric, statistical models, HPE (Human Pose Estimation) systems, and IMU-based (Inertial Measurement Unit) wearable systems [3]. Several issues have been identified in pose estimation, such as occlusions, poor lighting, complex body orientations, and dynamic backgrounds [3].

However, due to limited annotated datasets, privacy regulations, and ethical considerations, most existing models cannot provide real-time processing or be effectively adapted to different sports and variations in environmental conditions to calculate an athlete's back angle during training [4]. It is therefore crucial for this study to focus on developing an evaluation form, preventing injury, and enhancing performance by providing real-time posture monitoring. Particularly, individual athletes can benefit from a personalized intelligent training coach during strength training.

Leveraging computer vision, the proposed approach isolates the trunk, applies pre-processing and edge detection, and calculates key points to determine back angles from single images or video frames. YOLOv8, with its high inference speed and instance-segmentation capability, enables accurate and efficient trunk detection in diverse lighting and motion conditions [5]. Gaussian kernel smoothing and grayscale conversion reduce image noise and simplify feature extraction [6][7], while Canny edge detection ensures precise identification of trunk boundaries [8][9]. The model provides immediate feedback, allowing athletes to correct their posture during training. Such real-time monitoring can substantially reduce the risk of injury while optimizing performance, offering a practical tool for both professional and amateur athletes [10].

This study aims to develop a model for the real-time measurement of back angles during workouts. Unlike previous research that often relies on static datasets or high-computational HPE models, which frequently struggle with real-time processing and privacy constraints, this study introduces a streamlined, edge-aware model optimized for strength training. By combining proper posture correction with movement training, the model could further strengthen injury prevention programs for athletes, although each factor independently influences injury risk. A cost-effective and easy-to-implement approach for detecting body posture is also valuable not only for athletes but also for the general population, as it requires only a simple camera setup, making it highly accessible for everyday use.

Overall, this study makes the following three core technical contributions: (1) Adaptive real-time pipeline: Employs YOLOv8 instance segmentation to isolate the trunk, enabling high-speed inference in dynamic or low-light conditions. (2) Probabilistic edge refinement: Implements a probabilistic thresholding approach to distinguish stable structural boundaries from noise, providing a reliable trunk contour for angle calculation that outperforms standard gradient-based filters. (3) Accessible biometric feedback: Translates complex vision data into real-time back-angle measurements, offering a cost-effective, camera-based alternative to expensive mobile devices, wearables, or manual coaching for personalized injury prevention.

2. Related Work

The following literature review begins by discussing YOLOv8-based techniques for real-time object detection and background removal. Then, it examines image processing methods, including Gaussian kernel smoothing and grayscale conversion. Finally, edge detection techniques are reviewed, followed by a summary.

2.1. Real-Time Object Detection and Background Removal Using YOLOv8

YOLO ("You Only Look Once") is a real-time object detection model that transformed the field by reframing detection as a single regression problem from image pixels to bounding box coordinates and class probabilities. YOLO adopts a unified,

end-to-end design, unlike traditional multi-stage methods such as region-based R-CNN detectors [11] or architectures relying on deformable convolutional modules [12]. This design enables fast inference while maintaining high detection accuracy. In addition, this design allows real-time image processing and makes YOLO-based models particularly suitable for latency-sensitive computer vision applications [5] [13].

YOLO-based models have been widely adopted in real-world scenarios such as urban theft detection and retail security monitoring, where low latency and stable inference are essential [14]. The models have also been applied to industrial inspection and quality control tasks, including surface defect detection in manufacturing environments [5], fabric inspection in textiles [15], and agricultural disease detection [16].

Recent YOLO variants demonstrate strong potential for efficient deployment on resource-constrained edge and embedded devices [17]. Unlike classical foreground extraction methods such as GrabCut [18], YOLOv8 effectively separates foreground objects from background regions to automate background removal. Due to its high inference speed, prior studies confirm the practicality of YOLOv8 for real-time deployment under dynamic and unconstrained environments [19][20].

Recent studies increasingly highlight real-time object detection as a core requirement in safety monitoring, surveillance, and human-object interaction systems [13]. YOLOv8 enhancements demonstrate high resilience to illumination changes, occlusions, and complex backgrounds, significantly improving the clarity, accuracy, and robustness of downstream tasks [18]. These downstream tasks include contour extraction, geometric measurement, and posture analysis. By isolating target subjects and suppressing irrelevant visual information, the model offers a fully automated and learning-based solution. The model has demonstrated effectiveness in mobile and assistive applications, including Android-based object detection for visually impaired users [19] and voice-integrated real-time detection systems [20].

Compared with earlier YOLO variants (YOLOv1 to YOLOv7), YOLOv8 integrates features specifically designed for instance segmentation, enabling not only accurate object detection and standard bounding box generation but also the creation of precise, pixel-level segmentation masks for each object instance [5]. The model accomplishes this through an anchor-free, center-based approach, enhanced convolutional modules, and a decoupled head, which collectively contribute to superior performance in complex environments [5]. While the more recent YOLOv9 and YOLOv10 variants show improved performance in specific contexts, such as challenging agricultural scenarios [21], their empirical validation remains limited compared with YOLOv8, which is a mature, stable, and well-documented model.

2.2. Image Processing with Gaussian Kernel Smoothing and Grayscale

Gaussian kernel smoothing is generally employed as an image pre-processing step in vision-based measurement tasks to suppress high-frequency noise and stabilize gradient estimation for enhancing smoothness prior to edge detection [22]. This is achieved through the convolution of the input signal with a kernel function that is typically isotropic with radial symmetry, allowing a Gaussian function to assign weighted averages to neighboring values while preserving structural information and suppressing noise [6]. The degree of smoothing is controlled by the kernel's bandwidth, with larger bandwidths producing more extensive smoothing effects but causing edges to become less sharp [6]. By suppressing high-frequency noise, Gaussian smoothing improves edge continuity and reduces false gradients from motion or lighting changes, leading to enhanced reliability of subsequent contour extraction and geometric analysis [22]. It serves as a crucial pre-processing step to ensure stable and consistent edge detection under real-world workout environments (e.g., motion blur, lighting variations).

Image grayscale processing reduces the dimensionality of the data matrix, accelerates computations, and removes redundant color information [7]. In this study, the input images were converted to grayscale using the standard OpenCV implementation, which follows a weighted RGB formulation to combine the red, green, and blue (R, G, B) channels into a single intensity channel with values ranging from 0 (black) to 255 (white), where higher values indicate brighter pixels [7].

While color information provides negligible discriminative value for geometry-driven posture analysis, grayscale conversion stabilizes gradient estimation and optimizes computational efficiency, which is particularly advantageous for real-time and mobile vision systems [7].

OpenCV provides many well-established computer vision algorithms and pretrained models, allowing for the hybrid integration of classical image-processing techniques with deep-learning-based detection frameworks. By taking advantage of optimized CPU and GPU implementations, OpenCV supports reproducible and low-latency execution, which is critical for real-time vision analysis [7][22]. In OpenCV color conversion, grayscale employs a weighted method [7], assigning different weights to the R, G, and B channels to produce a grayscale map with distinct intensity levels. This weighted grayscale conversion further emphasizes pixel brightness and salient structural features while reducing computational complexity. Such simplification benefits downstream tasks, including edge detection, object recognition, and image segmentation [22].

2.3. Edge Detection Techniques

Edge detection performance is generally influenced by lighting conditions, object intensity similarities, edge density, and image noise [22]. A novel approach based on photography has been empirically verified for its effectiveness [22]. The performance of edge detection methods is therefore highly application-dependent and sensitive to pre-processing and parameter selection, particularly with respect to noise suppression and threshold determination[9][22]. To achieve stable and reliable edge extraction in real-world vision tasks, it is necessary to adopt both adaptive and well-established edge detection techniques and appropriate pre-processing.

The Canny edge detection algorithm implemented in OpenCV is a widely used multi-scale method for identifying image edges. It consists of four main stages: (1) image denoising using a Gaussian smoothing, (2) computation of gradient amplitude and direction, (3) non-maximum suppression to eliminate non-edge pixels and retain only the most salient edges, and (4) hysteresis thresholding using dual thresholds, i.e., high and low thresholds, to refine edge selection [8].

To reduce threshold sensitivity for the Canny edge detector, a probabilistic approach has been developed to automatically determine hysteresis thresholds, enhancing reliability across diverse imaging scenarios [9]. This approach employs conditional probability to evaluate edge likelihood by monitoring pixel behavior within an “instability zone,” defined as a gradient range where edge classification varies significantly with threshold changes [9]. This zone functions as a statistical filter where the probabilistic approach distinguishes between stable structural features and transient, noise-induced image artifacts. By analyzing edge maps across multiple candidate thresholds, the approach identifies statistical patterns in pixel behavior and adapts thresholding to image-specific characteristics.

This probabilistic approach enables dynamic, content-aware threshold selection, significantly enhancing the precision and robustness of edge detection, although its performance remains inherently context-dependent [9]. While gradient-based operators such as Sobel and Prewitt offer computational simplicity, they often lack the edge-thinning and noise-suppression capabilities provided by the Canny algorithm. Therefore, the refined Canny edge detector provides superior contour definition for posture analysis compared to standard gradient-based filters, which is essential for the precise structural analysis required in this study.

2.4. Summary

By reviewing existing state-of-the-art (SOTA) approaches in sports activity analysis, including wearable-based systems and vision-based human pose estimation methods, several limitations are identified within the specific context of this study—namely, the need for a non-invasive and mobile-friendly solution suitable for self-guided gym training. A gap therefore exists for a lightweight, vision-based posture monitoring framework that (1) avoids reliance on skeletal keypoints or wearable sensors,

(2) supports real-time operation on standard mobile hardware, and (3) provides quantitative and interpretable back-angle estimation across diverse exercises and loading conditions. This study addresses this gap by reframing trunk angle estimation as a segmentation- and geometry-driven problem, leveraging YOLOv8 instance segmentation and contour-based analysis to balance accuracy, robustness, and practical deployability.

3. Proposed Methodology

The overall model development procedure for this study is presented in Fig. 1. The model is developed with an emphasis on real-time feasibility, hardware independence, and deployment practicality, while maintaining satisfactory accuracy in posture monitoring, particularly in back-angle measurement for injury prevention and performance monitoring.

By adopting a vision-based experimental design, this study evaluates a proposed real-time model for back-angle estimation during strength training. The proposed model integrates YOLOv8-based object detection with classical image pre-processing techniques to achieve accurate and robust back-angle estimation from monocular side-view images and videos captured from either left- or right-facing viewpoints. Model development comprises the following stages: (1) training YOLOv8 models for background removal and trunk localization using annotated side-view human images; (2) converting images to grayscale to reduce the computational cost associated with Gaussian smoothing; (3) extracting geometric features from trunk contours via unsupervised Canny edge detection with data-driven threshold selection to measure back angles; and (4) comparing the detected angles with ground-truth measurements obtained using a standardized geometric procedure, enabling quantitative assessment using mean absolute error (MAE) and mean relative error.

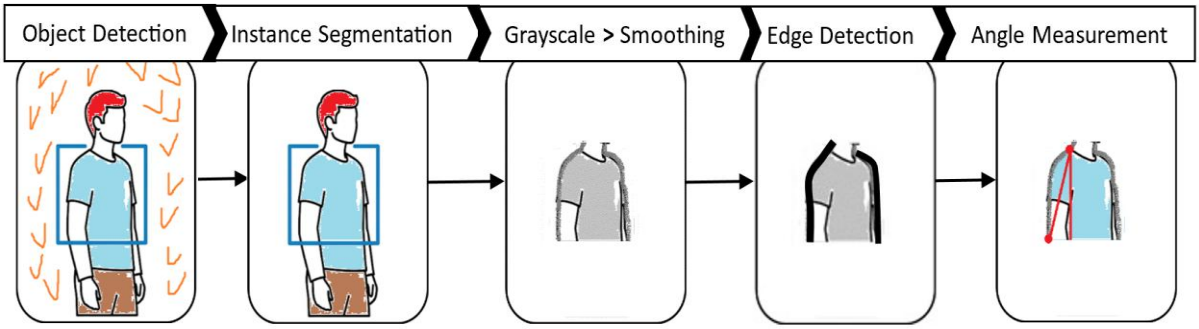


Fig. 1 Model development procedure

3.1 Dataset Construction and Annotation

Before model training, about 300 side-view human images were collected from online sources, such as Shutterstock.com and Alamy.com. Three criteria were adopted for image selection: (1) clear side-view visibility of the human trunk; (2) minimal occlusion of the upper body; and (3) sufficient resolution for contour extraction. The valid dataset covers a wide range of body movements, body types, and postures, providing the variability necessary for robust training of posture monitoring and accurate back-angle estimation.

Among the collected images, about 240 were annotated using the makesense.ai tool to define the trunk region with blue bounding boxes spanning from below the neck to above the hips. These annotated images were then used for model training. To ensure robustness under realistic workout conditions and prevent data leakage, all real workout video data were excluded from the training set, and performance evaluation was conducted exclusively on these unseen workout frames.

The annotated images were divided into training, validation, and test sets using an 8:1:1 ratio [23], following standard practice for fine-tuning pretrained deep-learning models. The base YOLOv8 model, pretrained on the COCO dataset, was then trained to recognize the human upper body.

3.2 Model Training and Background Removal

Image backgrounds can create unwanted noise or interfere with trunk detection results if not removed. YOLOv8 was employed twice in this study. The YOLOv8 instance segmentation model, specifically YOLOv8n-seg, provided effective background removal for both images and videos by generating accurate masks while maintaining real-time performance, an essential feature for workout scenarios in this study. Then, the YOLOv8 object detection model was applied to detect the upper body from the processed images (i.e., after background removal).

Following background removal, a YOLOv8 object detection model was applied to localize the upper body region within the segmented foreground. To enhance robustness in unconstrained workout environments, instance segmentation and upper-body detection were decoupled, ensuring background noise does not interfere with precise trunk localization.

The training parameters were set as follows: epochs = 60, imgsz (image size) = 640, flipud (vertical flip probability) = 0.0, fliplr (horizontal flip probability) = 0.5, scale = 0.5, hsv_h (hue adjustment) = 0.015, hsv_s (saturation adjustment) = 0.7, and hsv_v (value adjustment) = 0.4. These settings follow YOLOv8 default recommendations and prior studies, balancing convergence stability and overfitting avoidance for a medium-sized custom dataset.

3.3 Image Preprocessing and Edge Detection

In a real workout environment (Fig. 2(a)), one of the researchers performed weightlifting in a standing posture using a 22.7 kg barbell (originally measured as 50 lb) (i.e., barbell_50). Gaussian smoothing [6] was first applied to the recorded video frames (Fig. 2(b)) to reduce unwanted sharp edges and noise while preserving essential image information, preparing the frames for subsequent Canny edge detection. It should be noted that the images are converted to grayscale [7] (Fig. 2(c)) to reduce file size and computational complexity. These pre-processing steps improved processing efficiency and enhanced the quality of subsequent analyses.

(a) A sample video frame during a workout using a 50 lb barbell (i.e., barbell_50)

(b) After removing the image background and then applying Gaussian smoothing

(c) Applying grayscale

Fig. 2 Sequential image preprocessing steps

Rather than manually setting low and high thresholds, which can vary significantly between images, this study employed the Canny edge detection algorithm [8] with an unsupervised, probability-based approach [9] to dynamically adjust thresholds according to each image's characteristics. Thresholds were derived directly from the gradient magnitudes of the grayscale image [7] [22]. Gradient magnitudes, representing intensity changes between adjacent pixels, were first computed for each pixel and then flattened into a one-dimensional array. This array provided a distribution of gradient values, enabling data-driven and adaptive threshold selection.

In addition, to capture only the most significant edges, the high threshold was set to the 99th percentile of gradient values, sufficient to outline the upper contour of the human body for this application. The low threshold was defined as 50% of the high threshold [8], allowing weaker edge pixels to connect with stronger ones, ensuring continuous edges while filtering out noise. Overall, this unsupervised thresholding strategy enabled the Canny algorithm to adapt automatically to varying image conditions, balancing edge sensitivity and noise reduction. Consequently, it achieved robust and flexible edge detection while consistently preserving essential features.

3.4 Angle Computation and Coordinate Definition

After detecting the edge of a person's trunk using the Canny edge detector, a person's uprightness or slight posture skew was determined by extracting key reference points from the trunk contours. Each contour point is represented by its image-plane coordinates (x, y), which are defined relative to the top-left corner of the video frame. For the set of contour points within the trunk region, *min_x* and *max_x* denote the minimum and maximum horizontal coordinates, while *min_y* and *max_y* denote the minimum and maximum vertical coordinates, respectively. As illustrated by the gray area in Fig. 3, which is obtained from the YOLOv8 trunk detection model, the bounding limits of the trunk region of interest (ROI) are defined by these extrema.

All coordinates are calculated relative to the video frame's origin (0,0) at the very top-left corner. All contour points within the ROI were iterated to identify two anatomically significant reference points: the neck point and the back point. These points then serve as the primary markers for determining a person's orientation (facing left or right) and for computing the trunk inclination angle. For example, if a person faces left, the back point is obtained by selecting the contour point closest to the bottom-right point (*max_x*, *max_y*) from the contours. The person's neck point is then identified as the point closest to the top-left point (*min_x*, *min_y*).

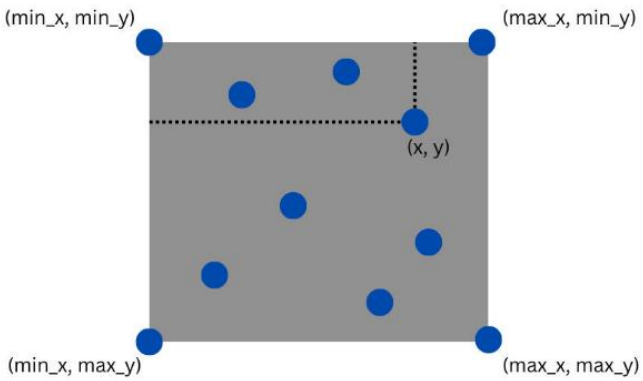


Fig. 3 The importance of the back point when the user is facing left

To robustly select the neck and back points, the contour point with the highest importance value was selected, and an importance-based geometric score was defined for each contour point in the following formula (1). The first term represents the normalized horizontal position of the point within the ROI, while the second term represents the normalized vertical position. The importance values varied across three possible posture scenarios, with corresponding formula adjustments for each. To account for the body's greater height relative to its width, a coefficient of 2 was applied in the formula to ensure that the identified point was positioned closer to the neck. This weighting biases the selection toward points closer to the upper body, ensuring that the identified neck point is positioned near the top of the trunk rather than along lateral edges. The contour point with the highest importance value is selected as the target point under the corresponding posture scenario. Similar formulations are used for other facing orientations by adjusting the normalization terms accordingly.

$$Importance = (x - x_{\min}) / (x_{\max} - x_{\min}) + 2 \times (y - y_{\min}) / (y_{\max} - y_{\min}) \quad (1)$$

Once the neck and back points were identified, the trunk inclination angle (θ) was calculated using basic planar geometry and the arctangent ($\tan^{-1}$) function. To measure the orientation, the neck and back points are treated as the vertices of a right-angled triangle. In this geometric configuration, the vertical distance between the two points serves as the opposite side, while the horizontal distance represents the adjacent side. The included angle θ is then calculated as the following formula (2). Then, as illustrated in Figs. 4 and 5, this angle represents the trunk inclination relative to the horizontal axis and serves as the quantitative measure of back posture used throughout the study.

$$\theta = \arctan(opposite / adjacent) \tag{2}$$

(a) A person facing right

(b) A person facing right with a calculated back angle

Fig. 4 During a workout using a barbell

(a) A person facing left

(b) A person facing left with a calculated back angle

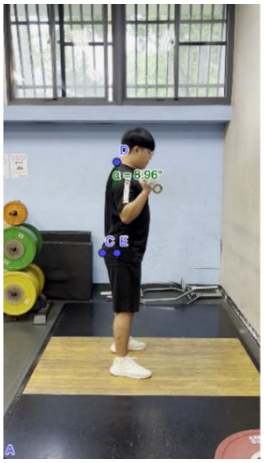
Fig. 5 During a workout using a pair of dumbbells

3.5 Ground-Truth Annotation and Evaluation Protocol

To establish ground truth for evaluating the proposed model, back angles were measured from selected video frames using GeoGebra Classic [24], which provides a protractor tool for computing the included angle once three points are defined. The procedure consisted of three steps: (1) uploading the selected video frames into GeoGebra Classic, (2) manually marking three points on each image - the neck point, the back point, and a third point forming a right-angled triangle (defined by the *x*-coordinate of the neck point and the *y*-coordinate of the back point), and (3) allowing GeoGebra Classic to automatically calculate the angle, which was used as the ground truth, based on these points. As stated earlier, all recorded workout video frames used for evaluation were excluded from the training dataset to ensure unbiased performance assessment. The same pre-processing, detection, and evaluation pipeline was applied consistently across all experiments to ensure reproducibility.

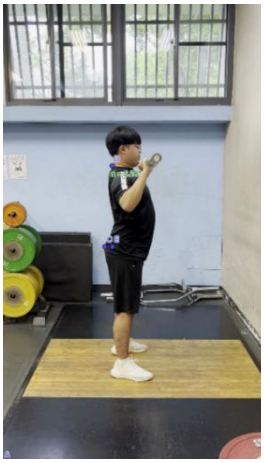
4. Experimental Results

To comprehensively evaluate the model's accuracy in measuring back angles, six video clips were recorded in the school gymnasium. Each clip featured the researcher, who filmed himself using a smartphone while performing one of six different workout types: barbell, dumbbell, seated chest press, squat, triceps pushdown, and weightlifting, with variations in posture, background, and orientation. Two frames with successful angle detections were selected from each video clip. For these frames, the ground-truth neck and back points were manually labeled using GeoGebra Classic, which then provided the corresponding back angle (left). These derived angles were compared with the angles detected by the proposed model (right), and the summary results are presented in Tables 1 and 2. As illustrated in Figs. 6-17, a total of 12 paired frames are presented.



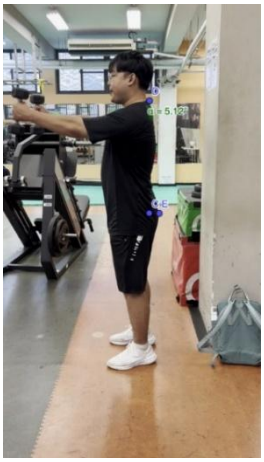
(a) Labeled by GeoGebra (b) Detected by the model

Fig. 6 During a workout using a 50-lb barbell (barbell_50)



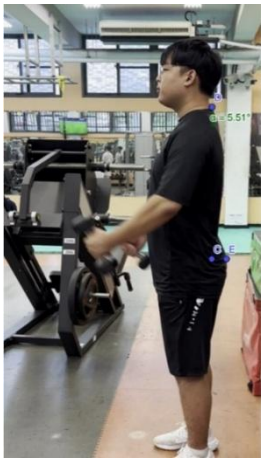
(a) Labeled by GeoGebra (b) Detected by the model

Fig. 7 During a workout using a 200-lb barbell (barbell_200)



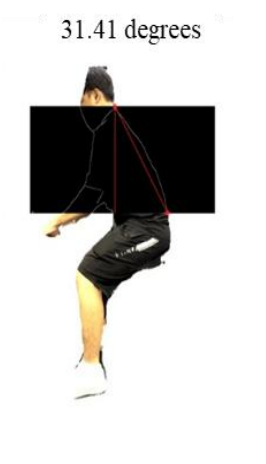
(a) Labeled by GeoGebra (b) Detected by the model

Fig. 8 During a workout using a pair of dumbbells weighing 50 lb in total (dumbbell_50)



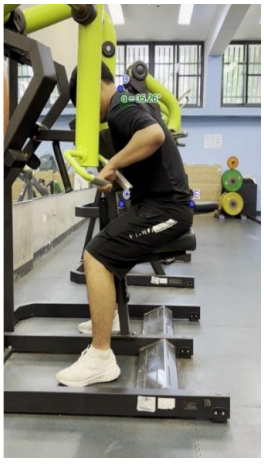
(a) Labeled by GeoGebra (b) Detected by the model

Fig. 9 During a workout using a pair of dumbbells weighing 200 lb in total (dumbbell_200)



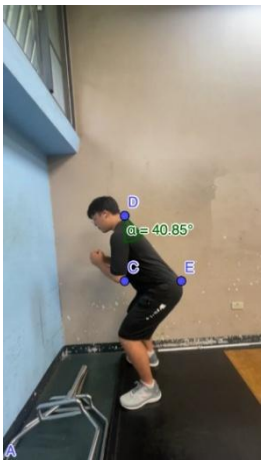
(a) Labeled by GeoGebra (b) Detected by the model

Fig. 10 During a workout using a 50-lb seated chest press (seated_chest_press_50)



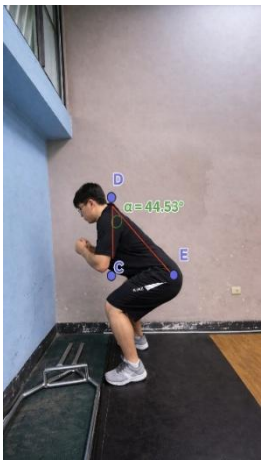
(a) Labeled by GeoGebra (b) Detected by the model

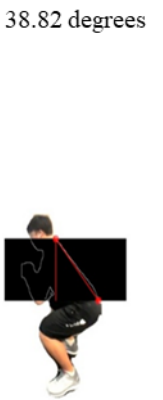
Fig. 11 During a workout using a 100-lb seated chest press (seated_chest_press_100)



(a) Labeled by GeoGebra (b) Detected by the model

Fig. 12 During a squat workout with a total load of 50 lb (squat_50)



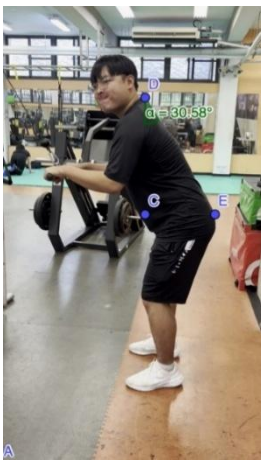


(a) Labeled by GeoGebra (b) Detected by the model

Fig. 13 During a squat workout with a total load of 100 lb (squat_100)

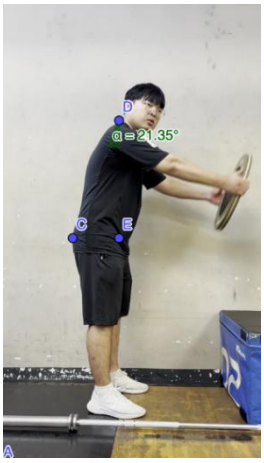
(a) Labeled by GeoGebra (b) Detected by the model

Fig. 14 During a workout using a 50-lb tricep pushdown (tricep_pushdown_50)



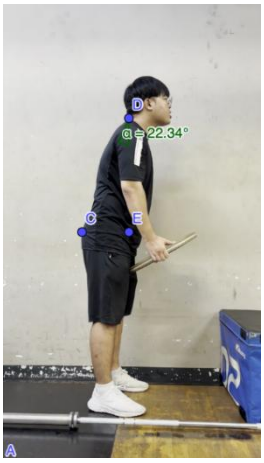
(a) Labeled by GeoGebra (b) Detected by the model

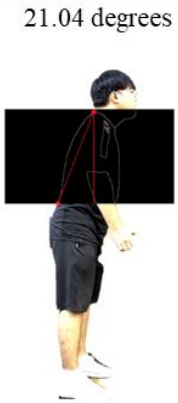
Fig. 15 During a workout using a 100-lb tricep pushdown (tricep_pushdown_100)



(a) Labeled by GeoGebra (b) Detected by the model

Fig. 16 During a workout using a 50-lb weight lifting (weight_lifting_50)





(a) Labeled by GeoGebra (b) Detected by the model

Fig. 17 During a workout using a 100-lb weight lifting (weight_lifting_100)

The mean relative error was rounded to two decimal places in both tables, and its measurement is *(a-b)/a,* where *a* is the labeled value obtained by GeoGebra Classic and *b* is the detected angle. Table 1 shows that the total detected angle sum (262.59) obtained from the developed model was close to the labeled sum (280.33), showing only an overall 6.49% relative difference and confirming high consistency across exercises. The MAE values are also reported in degrees in the Table. For performance comparison, the absolute relative error was used as the primary accuracy metric. Notably, upper-body exercises yielded a higher mean error rate (13%) than whole-body exercises (3%). However, the mean relative error is more useful than the absolute one for identifying experimental conditions that produce higher or lower estimation errors. The signed mean relative error reflects estimation bias: negative values indicate that trunk angles are overestimated, whereas positive values correspond to underestimation. This value indicates the direction of estimation bias. Notably, the symbols p and q were used respectively to label lighter and heavier loads in each workout type, allowing the identification of the synthesized results in the angle comparison. As presented in the table, although the upper-body workouts produced a higher mean relative error than the whole-body workouts, the lighter loads in the upper-body workouts generally resulted in smaller errors compared with the whole-body workouts, and vice versa.

Table 1 Model performance by body parts (labeled angles vs. detected angles)

Workout Type	Labeled Angle (°)	Detected Angle (°)	MAE*	Mean Relative Error	Angle Comparison
barbell_50 (p)	8.96	8.96	0.00	0	p_50 < q_200
barbell_200 (q)	5.28	2.58	2.70	0.51	
dumbbell_50 (p)	5.12	4.34	0.78	0.15	p_50 = q_200
dumbbell_200 (q)	5.51	4.67	0.84	0.15	
seated_chest_press_50 (p)	31.08	31.41	0.33 (-)	0.01 (-)	p_50 < q_100
seated_chest_press_100 (q)	35.26	27.09	8.17	0.23	
Upper Body	91.21	79.05	12.16	0.13 (or 0.1333)	p < q
squat_50 (p)	40.85	39.95	0.90	0.02	p_50 < q_100
squat_100 (q)	44.53	38.82	5.71	0.13	
tricep_pushdown_50 (p)	29.97	28.76	1.21	0.04	p_50 > q_100
tricep_pushdown_100 (q)	30.58	30.89	0.31 (-)	0.01 (-)	
weight_lifting_50 (p)	21.35	24.08	2.73 (-)	0.13 (-)	p_50 > q_100
weight_lifting_100 (q)	22.34	21.04	1.30	0.06	
Whole Body	189.62	183.54	6.08	0.03 (or 0.0321)	p > q
Sum	280.83	262.59	18.24	0.06 (or 0.0649)	Upper > Whole

* Each value was obtained by subtracting the labeled angle from the detected one. Negative values are presented with a minus sign to assist in interpreting the table.

Table 2 shows that heavier loading weights resulted in higher average relative errors (13%) compared with lighter loads (1%). Upper-body exercises with heavier loads yielded the highest mean relative error (25%), whereas whole-body exercises with lighter loads produced the lowest (0.3%). Upper-body exercises with lighter loads (1%) also produced fewer errors than whole-body exercises with heavier loads (7%). Within both lighter and heavier weight conditions, exercises exhibited distinct angle magnitudes, with squat_100 producing the largest trunk angles and barbell_200 the smallest. Across both load levels, upper-body exercises consistently showed greater angular deviations than whole-body movements, suggesting higher flexibility and motion variability in isolated upper-body actions.

Table 3 presents the ablation study results evaluating the role of each component in the proposed back-angle estimation pipeline. In the initial trials (Trials 1 and 2), instead of employing YOLOv8 segmentation, traditional approaches such as bounding-box-only detection and OpenPose-based systems were adopted, resulting in severe background interference or inaccurate trunk localization. These results demonstrate the necessity of learning-based foreground isolation for robust posture analysis [11]. When Gaussian smoothing or Canny edge detection was excluded (Trials 3 or 4), unstable gradients or the

absence of explicit boundary extraction were observed, confirming the importance of noise suppression and contour-based geometry for reliable angle computation [6]. As a single component was removed, the pipeline failed to reliably extract trunk boundaries or geometric reference points, preventing successful back-angle estimation.

Table 2 Model performance by loading weights (labeled angles vs. detected angles)

Workout Type	Labeled Angle (°)	Detected Angle (°)	MAE*	Mean Relative Error	Angle Comparison
barbell_50 (p)	8.96	8.96	0.00	0	dumbbell > weight_lifting > tricep_pushdown > squat > seated_chest_press > barbell
dumbbell_50 (p)	5.12	4.34	0.78	0.15	
seated_chest_press_50 (p)	31.08	31.41	0.33 (-)	0.01 (-)	
Upper Body	45.16	44.71	0.45	0.01 (or 0.0100)	
squat_50 (p)	40.85	39.05	1.80	0.04	
tricep_pushdown_50 (p)	29.97	28.76	1.21	0.04	
weight_lifting_50 (p)	21.35	24.08	2.73 (-)	0.13 (-)	
Whole Body	92.27	91.89	0.28	0.003 (or 0.0030)	
Sum (Lighter Weight)	137.33	136.60	0.73	0.01 (or 0.0053)	Upper > Whole
barbell_200 (q)	5.28	2.58	2.70	0.51	barbell > seated_chest_press > dumbbell > squat > weight_lifting > tricep_pushdown
dumbbell_200 (q)	5.51	4.67	0.84	0.15	
seated_chest_press_100 (q)	35.26	27.09	8.17	0.23	
Upper Body	46.05	34.34	11.71	0.25 (or 0.2543)	
squat_100 (q)	44.53	38.82	5.71	0.13	
tricep_pushdown_100 (q)	30.58	30.89	0.31 (-)	0.01 (-)	
weight_lifting_100 (q)	22.34	21.04	1.30	0.06	
Whole Body	97.45	90.75	6.70	0.07 (or 0.0688)	
Sum (Heavier Weight)	143.50	125.09	18.41	0.13 (or 0.1283)	Upper > Whole

* Each value was obtained by subtracting the labeled angle from the detected one. Negative values are presented with a minus sign to assist in interpreting the table.

Table 3 Ablation study results

Model	YOLOv8 Segmentation	Gaussian Smoothing	Grayscale	Canny Edge	Results
Trial 1	x	x	✓	✓	Background interference
Trial 2	x	✓	✓	✓	Inaccurate trunk detection
Trial 3	✓	x	✓	✓	No noise suppression
Trial 4	✓	✓	✓	x	No explicit boundary extraction
This Study	✓	✓	✓	✓	Expected and satisfactory

Finally, the complete pipeline of this study achieved expected and satisfactory performance, indicating that accurate real-time video-based back-angle estimation depends on the integrated use of YOLOv8 segmentation, Gaussian smoothing, grayscale conversion, and Canny edge detection, effectively combining deep-learning-based segmentation with classical image-processing techniques.

Table 4 compares the proposed model against state-of-the-art (SOTA) posture and pose estimation approaches identified in the literature review. The comparison focuses on sensing modality, real-time capability, and average estimation error to validate the model's performance and technical contributions. By combining YOLOv8-based instance segmentation with edge-based geometric analysis, the model estimates trunk angles without the need for skeletal keypoints or wearable sensors. It achieves a mean relative error of approximately 6.49% while maintaining real-time performance on a mobile device. These comparative analysis results demonstrate a favorable balance between accuracy and ease of deployment, making the model well-suited for real-time posture monitoring and injury prevention in strength training.

Table 4 SOTA comparative analysis results

Method	Modality	Real-time	Mean Error (%)	Citation
IMU-based Wearables	Sensor (IMU)	Yes	About 5 ~ 8%	[6]
Traditional HPE / OpenPose	Vision (RGB, Keypoints)	Limited	About 10 ~ 15%	[2-4]
Multimodal Pose	Vision + Sensor	No	About 8 ~ 12%	[25]
Proposed Model (YOLOv8 + edges)	Vision	Yes	About 6.49%	This Study

Notably, the computational cost of the proposed model is low relative to IMU-based wearables (low to medium), traditional HPE/OpenPose (high to very high), and multimodal pose systems (very high) [10]. A definitive benchmark of execution cost, however, was excluded from this comparison due to the high variability in mobile device hardware, operating systems, and edge computing capabilities [17]. Hardware constraints such as battery health and memory capacity can significantly impact both computational efficiency and the resulting user experience, particularly regarding latency [17]. The algorithm exhibits constant FLOPs (floating point operations), whereas its execution latency is hardware-dependent [17]. To avoid device-specific bias and emphasize model generalizability, this study adopts the mean relative error as a hardware-independent robustness metric.

5. Discussion

Personal training is commonly conducted in gym environments where lighting conditions and background scenes are highly variable. Moreover, most existing mobile fitness applications primarily provide instructional content and rarely offer automated assistance for correcting workout posture. As independent training without professional coaching has become increasingly common, accurate measurement of back angle during workouts has emerged as a critical factor for injury prevention, particularly for lower-back-related injuries.

Among existing technological approaches for back-angle measurement using mobile devices, wearable sensor–based systems, vision-based HPE methods relying on skeletal keypoints, and hybrid multimodal approaches are the most widely adopted. Wearable IMU-based solutions can achieve high accuracy in joint angle and trunk motion estimation; however, they require dedicated hardware, precise sensor placement, and frequent calibration, which limits scalability and user acceptance in everyday training environments [10]. Multimodal systems that combine vision and wearable sensors improve robustness further; however, the systems introduce additional system complexity and are often unsuitable for real-time deployment on mobile or edge devices [25].

In the context of gym-based training, “pose” refers to posture monitoring models derived from pose estimation techniques. This study initially adopted OpenPose as a baseline; however, it lacks a dedicated back landmark, which is essential for reliable trunk angle computation [2]. Therefore, the study shifted from keypoint-based pose estimation toward an object-boundary-driven strategy following background removal. The core objective was to develop a real-time back-angle estimation model that does not rely on skeletal keypoints but instead leverages explicit body contours. By integrating YOLOv8-based instance segmentation with edge-based contour extraction and geometric analysis, the proposed model directly estimates trunk angles without relying on skeletal keypoints or wearable sensors. This design choice represents a fundamental departure from prior vision-based posture analysis methods and addresses known limitations of keypoint instability, occlusion, and landmark ambiguity in strength-training environments.

Early experiments applied the GrabCut algorithm [18] for foreground extraction, followed by edge detection using the Canny operator [8]. While Sobel filtering [26] and GrabCut-based segmentation were explored; these approaches produced unstable boundaries under complex gym backgrounds and motion, leading to unreliable angle estimation. These limitations motivated the adoption of YOLO-based models, specifically YOLOv8 instance segmentation, which offers cleaner foreground isolation and more consistent trunk contours under diverse lighting and motion conditions.

Moreover, to reduce noise sensitivity in the Canny edge detector, the high-low hysteresis thresholds were tuned to balance detail preservation and noise suppression. Through iterative refinement of the extracted contour map, the neck and back points can be reliably identified. In addition, to improve the robustness of geometry-driven landmark identification for the neck and back points, an importance-based geometric scoring formula was introduced to estimate anatomical landmarks after determining trunk orientation (left- or right-facing). This formulation effectively reduced extreme miscalculations and produced point estimates closely aligned with true anatomical locations. The horizontal weighting factor within the formula was empirically calibrated using an anatomy-informed evaluation process, and experimental results confirmed its suitability across multiple exercise scenarios.

As shown in Table 3, the proposed model achieves a mean relative error of approximately 6.49%, which offers a practical balance between accuracy and deployment feasibility for real-time, video-based posture monitoring on mobile devices in gym environments. The model performs best for whole-body exercises involving lighter loads, which are prevalent in typical gym settings. In contrast, upper-body exercises with heavier weights occasionally led to less accurate trunk detection. Nevertheless, even under these challenging conditions, the YOLOv8-based approach maintained stable detection performance across diverse movements and load levels, demonstrating robustness for video-based posture monitoring.

Another distinguishing contribution of this study lies in its explicit analysis of performance across exercise types and loading conditions. Comparative results in Tables 1 and 2 show that whole-body exercises consistently yield lower relative errors than upper-body exercises, and lighter loads produce more accurate angle estimates than heavier loads. While these trends align with biomechanical expectations, given greater motion variability and localized flexibility in upper-body training, they are rarely quantified in prior posture estimation studies. By reporting error distributions across exercise categories and load levels, this work provides a more nuanced evaluation than the aggregate accuracy metrics typically used in related research.

With continued advances in sports and posture-monitoring technologies, the proposed model shows strong potential for integration into mobile or vision-based wearable systems aimed at reducing back injuries, particularly in weightlifting contexts [10]. Accuracy could be further improved by training a more robust YOLOv8-based trunk detection model using a larger and more diverse dataset. Moreover, with lightweight optimization techniques [27], the framework could be extended to real-time human action recognition on mobile phones or embedded camera platforms. It should be noted that in this study, the robustness claim refers to the stability and consistency of the proposed model across different exercise types and loading conditions, rather than large-scale statistical generalization.

Overall, this study's novelty lies not in proposing an effective pose estimator, but in reframing back-angle estimation for real-time, mobile, and non-intrusive deployment. This perspective differentiates the proposed approach from YOLO-based studies that focus primarily on object localization or classification, as well as from HPE-based posture systems that depend on dense skeletal keypoints. By demonstrating consistent accuracy across diverse exercises and loading conditions, this study advances the SOTA toward practical, vision-based posture monitoring for injury prevention in strength training.

6. Conclusions

This study successfully developed a real-time, video-based posture monitoring model for estimating back angles during strength training by integrating YOLOv8 instance segmentation, contour extraction, and Canny edge detection. Using only a single mobile camera, the proposed approach provides a non-intrusive and accessible alternative to traditional sensor-based systems. The model achieved a mean relative error of approximately 6.49% while maintaining real-time performance on mobile devices, demonstrating reliable geometric estimation under limited computational resources. These results confirm the feasibility of accurate, efficient, and practical back-posture monitoring using low-cost vision-based methods.

The main conclusions and technical findings are summarized as follows:

(1) Accurate estimation was achieved through stable identification of neck and back points extracted from Canny-generated contours. Iterative refinement of contour selection significantly improved trunk localization stability, enabling accurate back-angle computation with a mean relative error of approximately 6.49% in real-time settings.

(2) The proposed importance-based geometric formulation, incorporating topmost-bottommost point selection and orientation-based tie-breaking, enhanced the robustness of trunk bounding box detection. Experimental results confirmed improved back-angle consistency across different exercises and loading conditions.

(3) YOLOv8-based instance segmentation enabled superior background removal and contour extraction compared with traditional methods, eliminating the need for wearable sensors or predefined skeletal landmarks. Camera placement at approximately sternum height (e.g., seated_chest_press and tricep_pushdown) was also found to maximize estimation accuracy and robustness.

Despite these strengths, several limitations remain. The evaluation was conducted on a limited dataset, highlighting the need for large-scale validation. User orientation was treated as a predefined input due to low classification accuracy in preliminary experiments, indicating the need for improved orientation classifiers trained on larger datasets. In addition, the geometric formulation is optimized for upright or slightly forward-leaning postures and may require extension to support non-standard body positions. Camera placement also remains a practical constraint, which future work could address through perspective correction or 3D reconstruction techniques. Overall, the proposed model offers a strong foundation for real-time feedback, injury prevention, and intelligent coaching systems, demonstrating that effective posture monitoring can be achieved without expensive sensors or laboratory-grade equipment.

Conflicts of Interest

The authors declare no conflict of interest.

Statement of Ethical Approval

For this type of study, a statement of human rights is not required.

Statement of Informed Consent

For this type of study, informed consent is not required.

References

[1] M. Kritz, J. Cronin, and P. Hume, “The Bodyweight Squat: A Movement Screen for the Squat Pattern,” Strength and Conditioning Journal, vol. 31, pp. 76-85, 2009.

[2] W. Kim, J. Sung, D. Saakes, C. Huang, and S. Xiong, “Ergonomic Postural Assessment Using a New Open-Source Human Pose Estimation Technology (OpenPose),” International Journal of Industrial Ergonomics, vol. 84, article no. 103164, 2021.

[3] T. Fukushima, P. Blauberger, T. Guedes Russomanno, and M. Lames, “The Potential of Human Pose Estimation for Motion Capture in Sports: A Validation Study,” Sports Engineering, vol. 27, article no. 19, 2024.

[4] M. H. B. Uçar, S. Solak, A. O. Jimale, H. Ünal, and S. Eken, “A Comprehensive Survey on Pose Estimation and Tracking in Sports: Methodologies, Datasets, Challenges, and Future Directions,” Artificial Intelligence Review, vol. 59, article no. 54, 2026.

[5] M. Hussain, “YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection,” Machines, vol. 11, no. 7, article no. 677, 2023.

[6] A. M. Wink and J. B. T. M. Roerdink, “Denoising Functional MR Images: A Comparison of Wavelet Denoising and Gaussian Smoothing,” IEEE Transactions on Medical Imaging, vol. 23, no. 3, pp. 374-387, 2004.

[7] Q. Huang, C. Zhou, and G. Zhang, "Color Code Structure Design and Recognition using OpenCV," The 2022 2nd International Conference on Computer Graphics, Image and Virtualization, IEEE Press, Chongqing, China, pp. 28-33, 2022.
[8] Z. Xu, B. Xue, and G. Wu, "Canny Edge Detection Based on OpenCV," The 2017 13th IEEE International Conference on Electronic Measurement & Instruments, IEEE Press, Yangzhou, China, pp. 53-56, 2017.
[9] R. Medina-Carnicer, R. Muñoz-Salinas, E. Yeguas-Bolivar, and L. Díaz-Más, "A Novel Method to Look for the Hysteresis Thresholds for the Canny Edge Detector," Pattern Recognition, vol. 44, no. 6, pp. 1201-1211, 2011.
[10] L. Jenkins and R. Weerasekera, "Sport-related Back Injury Prevention with a Wearable Device," Biosensors and Bioelectronics: X, vol. 11, article no. 100202, 2022.
[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Press, Columbus, OH, USA, pp. 580-587, 2014.
[12] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, et al., "Deformable Convolutional Networks," Proceedings of the 2017 IEEE International Conference on Computer Vision, IEEE Press, Venice, Italy, pp. 764-773, 2017.
[13] S. S. More and R. Bansode, "FCN-YOLOS: An Effective Deep-Learning Model for Real-Time Object Detection," Journal of Field Robotics, vol. 42, no. 8, pp. 4053-4074, 2025.
[14] A. S. Hameed, T. M. Hasan, and R. Khaji, "Real Time Classification of Retail Theft Utilizing YOLO Algorithm," Ingénierie des Systèmes d'Information, vol. 30, no. 6, pp. 1517, 2025.
[15] W. Song, D. Lang, J. Zhang, M. Zheng, and X. Li, "Textile Defect Detection Algorithm Based on the Improved YOLOv8," IEEE Access, vol. 13, pp. 11217-11231, 2025.
[16] X. Wang and J. Liu, "Vegetable Disease Detection Using an Improved YOLOv8 Algorithm in the Greenhouse Plant Environment," Scientific Reports, vol. 14, article no. 4261, 2024.
[17] X. Liu, T. Wang, J. Yang, C. Tang, and J. Lv, "MPQ-YOLO: Ultra Low Mixed-Precision Quantization of YOLO for Edge Devices Deployment," Neurocomputing, vol. 574, article no. 127210, 2024.
[18] C. Li, X. Zhao, and H. Ru, "GrabCut Algorithm Fusion of Extreme Point Features," The 2021 International Conference on Intelligent Computing, Automation and Applications, IEEE Press, Nanjing, China, pp. 33-38, 2021.
[19] S. S. More, N. Patil, V. B. Lobo, N. Shet, D. Goswami, and P. Rane, "Empowering the Visually Impaired: YOLOv8-Based Object Detection in Android Applications," Procedia Computer Science, vol. 252, pp. 457-469, 2025.
[20] I. Singh, N. Singh, G. Singh, and P. Krishna, "YOLOv8 Based Object Detection with Custom Dataset and Voice Command Integration," The 2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), IEEE Press, Erode, India, pp. 374-381, 2025.
[21] C. Liu, Y. Shen, F. Mu, H. Long, A. Bilal, X. Yu, et al., "Detection of Surface Defects in Soybean Seeds Based on Improved YOLOv9," Scientific Reports, vol. 15, article no. 12631, 2025.
[22] E. Nadernejad, S. Sharifzadeh, and H. Hassanpour, "Edge Detection Techniques: Evaluations and Comparisons," Applied Mathematics and Sciences, vol. 2, no. 31, pp. 1507-1520, 2008.
[23] H. Bichri, A. Chergui, and M. Hain, "Investigating the Impact of Train/Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets," International Journal of Advanced Computer Science and Applications, vol. 15, no. 2, pp. 331-339, 2024.
[24] A. Or, "Measuring Angles Using Protractor," https://www.geogebra.org/m/fMnsdbzv, accessed in 2025.
[25] C. Ji, Y. Zhong, and M. Gao, "Multimodal Fusion Approach for Sports Injury Prevention and Pose Keypoint Detection," PLoS One, vol. 20, no. 8, article no. e0327911, 2025.
[26] Q. Chang, X. Li, Y. Li, and J. Miyazaki, "Multi-directional Sobel Operator Kernel on GPUs," Journal of Parallel and Distributed Computing, vol. 177, pp. 160-170, 2023.
[27] D. Q. Vu, N. T. H. Le, and J. C. Wang, "(2+1)D Distilled ShuffleNet: A Lightweight Unsupervised Distillation Network for Human Action Recognition," The 2022 26th International Conference on Pattern Recognition, IEEE Press, Montreal, Quebec, Canada, pp. 3197-3203, 2022.

Copyright© by the authors. Licensee TAETI, Taiwan. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC) license (https://creativecommons.org/licenses/by-nc/4.0/).