Pose-Based Binary Shooting Posture Classification Using YOLO-Pose and SVM

Warakorn Luangluewut
Department of Electrical Engineering
Kasetsart University
Bangkok, Thailand
Data Communication Division
Defence Technology Institute
Nonthaburi, Thailand
warakorn.luan@ku.th

Phunsak Thiennviboon*
Department of Electrical Engineering
Kasetsart University
Bangkok, Thailand
phunsak.t@ku.th
*Corresponding Author

Kittakorn Viriyasatr
Data Communication Division
Defence Technology Institute
Nonthaburi, Thailand
kittakorn.v@dti.or.th

Chayayot Saerejittima
Simulation Systems and Virtual Training Division
Defence Technology Institute
Nonthaburi, Thailand
chayayot.s@dti.or.th

Gunthorn Nathong
Testing and Evaluation Division
Defence Technology Institute
Nonthaburi, Thailand
gunthorn.n@dti.or.th

Kittituch Thanompongchart
Testing and Evaluation Division
Defence Technology Institute
Nonthaburi, Thailand
kittituch.t@dti.or.th

Piyarose Maleecharoen
Simulation Systems and Virtual Training Division
Defence Technology Institute
Nonthaburi, Thailand
piyarose.m@dti.or.th

Naris Channum
Simulation Systems and Virtual Training Division
Defence Technology Institute
Nonthaburi, Thailand
naris.c@dti.or.th

Jedsada Kraikhow
Simulation Systems and Virtual Training Division
Defence Technology Institute
Nonthaburi, Thailand
jedsada.k@dti.or.th

Prakorn Pratoomma
Simulation Systems and Virtual Training Division
Defence Technology Institute
Nonthaburi, Thailand
prakorn.p@dti.or.th

***Abstract*—Human pose estimation has been widely applied in various fields such as sports performance assessment and health data analysis. Firearm training can also benefit from this technology, as correct shooting postures improve both accuracy and trainee safety. This study presents a pose-based approach for binary classification of standing shooting postures, determining whether a posture is correct or incorrect. The proposed method employs a deep learning-based Human Pose Estimation (HPE) model that extracts key body points (keypoints) and their confidence scores using YOLO-pose models. These extracted features are combined into a feature vector and classified using a Support Vector Machine (SVM) with different kernel functions, namely: Linear, Polynomial, RBF, and Sigmoid. Experimental results show that the Linear and RBF SVMs combined with YOLOv8n-pose or YOLOv11n-pose models achieve accuracies and F1-scores above 80%. The highest performance, reaching 87.34% Accuracy and 87.27% F1-score, is achieved by the Linear SVM using YOLOv11n-pose features extracted from rear-view camera images. Therefore, the combination of practical YOLO-pose feature extraction with the highly interpretable Linear SVM establishes the proposed method as a promising, efficient, and interpretable real-time solution for shooting posture evaluation.**

***Index Terms*—Binary classification, human pose estimation, pose based classification, real time system, shooting posture classification, support vector machine, YOLO pose model.**

I. Introduction

This paper presents an experimental study on the development of an application for detecting shooting postures among military cadets. The primary objective is to evaluate whether the cadets adopt the correct shooting posture, which is a crucial factor influencing shooting performance [1]. The research focuses on classifying standing shooting postures, as shown in Fig. 1, into “correct” and “incorrect” categories.

Previous studies have attempted to classify postures using wearable sensors attached to the body [2], [3]. These sensors provide positional data in the form of vectors [4], which are then used for posture classification. For example, Hai Li [2] employed Support Vector Machines (SVM) [5], [6] for classification, achieving high accuracy. However, a major drawback of this approach is its practical complexity, as it requires physical sensor attachment to the user.

Recent advances in deep learning-based Human Pose Estimation (HPE) have enabled body joint detection directly from images, eliminating the need for body-worn sensors [7]. This approach simplifies data collection and enhances usability. Among existing models, OpenPose [8], [9] is one of the most widely used frameworks for human-pose estimation. Nevertheless, OpenPose suffers from limitations such as slow inference speed and error accumulation from multi-stage processing [10]–[13]. Recent YOLO-pose models [10] have been developed to address these issues, offering faster, single-stage inference and improved efficiency for real-time applications.

Therefore, this study aims to combine the strengths of previous approaches to develop a more effective method for classifying shooting postures and determining whether they are correct or incorrect. The paper is organized as follows. The background material of this work is briefly introduced in Section II. The research methodology is described in Section III. Results and discussion are provided in Section IV. Finally, the conclusion and future work are in Section V.

II. Background

In recent years, deep learning-based Human Pose Estimation (HPE) has been widely studied for detecting anatomical keypoints on the human body [7], [14]. Several models, including MoveNet [15], OpenPose [16], and YOLO-pose [10], have been proposed and adopted. Among them, YOLO-pose is particularly suitable for real-time applications owing to its single-stage detection and high inference speed [10], [11]. This study employs two YOLO-pose variants, YOLOv8n-pose and YOLOv11n-pose, to evaluate their effectiveness in keypoint detection for shooting posture classification. The extracted keypoints are used to determine whether a standing posture is correct or incorrect through a Support Vector Machine (SVM) classifier [5], [6], with multiple kernel functions [17] tested to identify the best configuration.

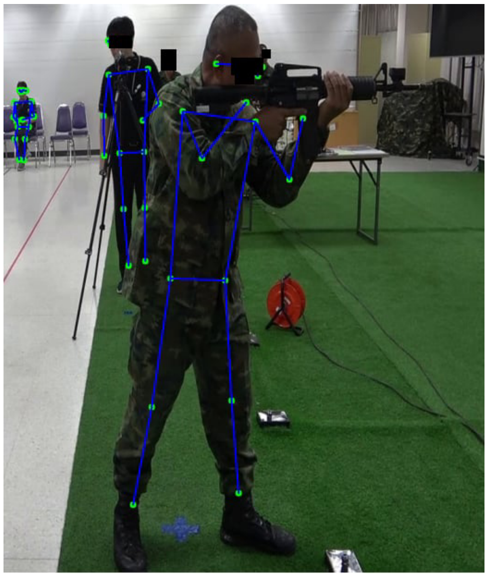

Fig. 1. Example of HPE output using the YOLOv8n-pose model.

A. *Related Work in Posture Classification*

Various approaches have been proposed for motion or posture classification, ranging from wearable motion sensors attached to the body [2], [3] to deep learning-based Human Pose Estimation (HPE) models that detect keypoints directly from images [8], [9]. Algorithms such as the Support Vector Machine (SVM) [5], [6] are commonly used for their balance between computational efficiency and accuracy. However, studies such as [2] relied on inertial measurement unit (IMU) sensors rather than deep learning-based HPE, limiting practicality since sensors must be physically attached to the user for each session.

Deep learning-based HPE [7], [14] extracts body-joint features and represents them as two-dimensional coordinates (x, y). Each detected point, called a *keypoint*, varies in number depending on the model, and most HPE systems also output a *confidence score* indicating the reliability of each detection. When multiple individuals appear in an image, the model associates keypoints for each person to form separate skeletal representations, as illustrated in Fig. 1.

B. *YOLO-Pose Models*

For pose detection, this study employs models that are real-time, lightweight, and efficient. Prior research [10], [11] has shown that YOLO-pose is well suited for such tasks due to several advantages:

- **Single Forward Pass:** Detects bounding boxes and poses of multiple individuals in one inference step.
- **No Complex Post-Processing:** Eliminates multi-stage processing, yielding stable runtime and low latency.
- **End-to-End Design:** Processes an image and directly outputs final detections.
- **Model Scaling:** Allows depth, width, and resolution adjustment to fit various hardware and accuracy needs.

Trained on the COCO keypoints dataset, YOLO-pose predicts 17 keypoints representing major body joints (Fig. 1). In this study, two variants, YOLOv8n-pose and YOLOv11n-pose, are used for keypoint detection and confidence estimation. Their implementation within the overall framework is described in Section III.

C. *YOLOv8-pose vs. YOLOv11-pose*

Successive versions of the YOLO-pose model [10] have been continuously developed to improve speed and accuracy. Among them, YOLOv8-pose [13] is one of the most widely studied and adopted variants for pose estimation. YOLOv11-pose [18], introduced more recently, enhances YOLOv8-pose with several notable architectural upgrades. Specifically, the backbone has been updated from C2f to C3k2, reducing parameters while enabling deeper and more efficient feature extraction. The addition of C2PSA and EFPN modules further decreases parameter count and improves inference speed.

As summarized in Table I [19], [20], YOLOv11-pose achieves a lower parameter count and higher inference speed than YOLOv8-pose, demonstrating its improved computational efficiency.

TABLE I
COMPARISON OF YOLOV8N-POSE AND YOLOV11N-POSE MODELS

Model	Size (pixels)	Speed CPU / GPU (ms)	Params (M)	FLOPs (B)
YOLO11n-pose	640 × 640	52.4 ± 0.5 / 1.7 ± 0.0	2.9	7.4
YOLOv8n-pose	640 × 640	131.8 / 1.18	3.3	9.2

D. Support Vector Machine (SVM)

In prior studies on motion classification, Hai Li [2] proposed a posture classification method based on the Support Vector Machine (SVM), which achieves an excellent balance between accuracy and computational efficiency, making it suitable for real-time applications. The SVM decision function for binary classification [5], [6] is expressed as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{1}$$

$$\hat{y} = \begin{cases} 1 \ (\textit{Correct posture}), & \text{if } f(\boldsymbol{x}) \geq \textit{threshold} \\ 0 \ (\textit{Incorrect posture}), & \text{otherwise.} \end{cases} \tag{2}$$

As shown in (1)–(2), the kernel function $K(\boldsymbol{x}_1, \boldsymbol{x}_2)$ plays a key role in mapping the input vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n$ into higher-dimensional spaces to achieve linear separability. Several commonly used kernel types are summarized in Table II. The kernel function is controlled by several hyperparameters: the **regularization parameter** C (controlling misclassification penalty), and the **kernel-specific parameters** γ, c, and d. γ (gamma) defines the scaling factor for Polynomial, RBF, and Sigmoid kernels; d (degree) sets the polynomial degree; and c (coefficient, or $coef0$) is the independent term in the Polynomial and Sigmoid kernels. The linear kernel uses only the regularization parameter C.

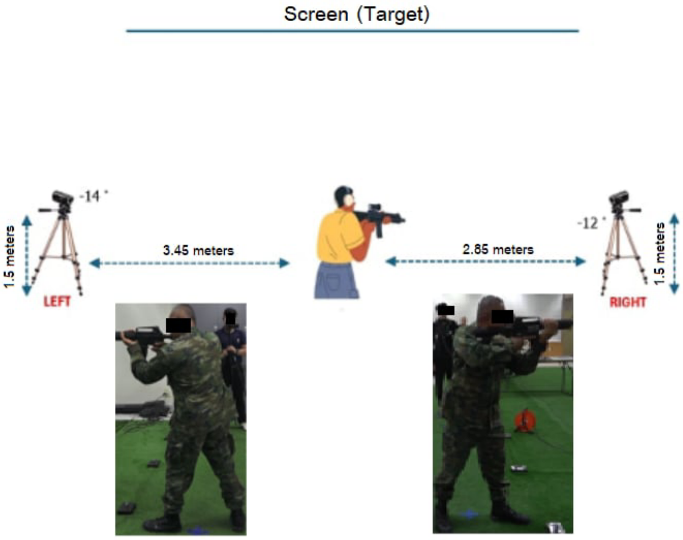

Fig. 2. Illustration of the data collection setup showing two cameras (left: rear view, right: front view), the demonstrator, and the target screen, along with example images from both camera views.

III. RESEARCH METHODOLOGY

A. Data Collection and Data Preparation

During the data collection stage, shooting postures were captured through continuous video recordings using two cameras positioned on the left and right sides of the demonstrators. The right-side camera was placed 2.85 m from the subject, while the left-side camera was positioned 3.45 m away. An example of the shooting postures and the camera placement setup is shown in Fig. 2, where the screen visible in the figure represents the target toward which the demonstrators aimed. Each video recording began with a demonstrator walking from the entrance to the shooting stage, after which he assumed either a correct or incorrect shooting posture. The demonstrator then maintained the posture for approximately 30 s before the end of the recording.

TABLE II
COMMON KERNEL FUNCTIONS IN SUPPORT VECTOR MACHINES (SVMS)

Kernel Type	Formula
Linear	$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1 \cdot \boldsymbol{x}_2$
Polynomial	$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\gamma\, \boldsymbol{x}_1 \cdot \boldsymbol{x}_2 + c)^d$
Gaussian / RBF	$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\gamma \,\lVert \boldsymbol{x}_1 - \boldsymbol{x}_2 \rVert^2)$
Sigmoid	$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \tanh(\gamma\, \boldsymbol{x}_1 \cdot \boldsymbol{x}_2 + c)$

Next, images were extracted from every frame of the recorded videos. Each image was processed using the YOLOv8n object detection model to identify objects labeled as "Person." For each image, only the largest detected "Person" instance, corresponding to the demonstrator, was manually selected and labeled as either *True* or *False*. A *True* instance represents a correct shooting posture captured from the portion of a video in which the demonstrator maintained a correct posture, whereas a *False* instance represents an incorrect shooting posture obtained from any portion of a video recorded for an incorrect posture. The selected regions, referred to as "cropped images," were then extracted and compiled to form the experimental dataset. The same procedure was applied to all video frames from both the left and right cameras. An example of this data preparation process is shown in Fig. 3.

Finally, all selected cropped images were organized into two independent 4-fold datasets, one for the Left camera (Left dataset) and one for the Right camera (Right dataset). Each fold contained two subsets, where each subset consisted of either *True* or *False* images of the same demonstrator captured from the corresponding camera view. Within the same fold, two subsets contained images of the same demonstrator, except for the first fold of the Left dataset. Importantly, subsets belonging to different folds included images from different demonstrators to ensure subject-independent evaluation. These two 4-fold datasets were then used separately for 4-fold cross-validation [21], [22]. The sample distribution across folds, binary classes (*True* and *False*), and camera views is summarized in Table III.

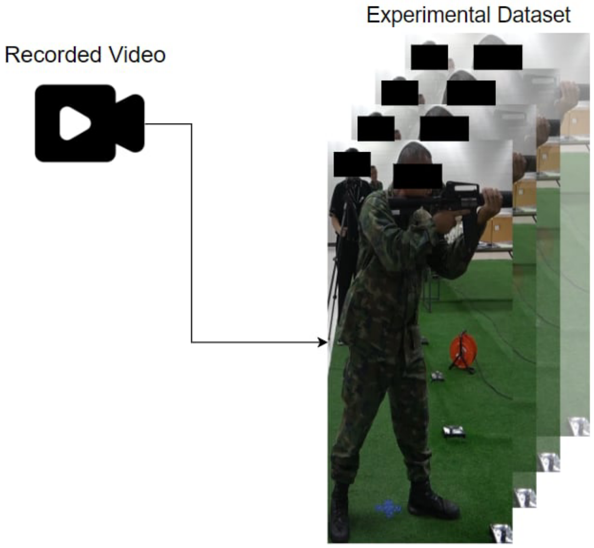

Fig. 3. Example of the data preparation process, in which the largest detected "Person" object, corresponding to the demonstrator, is extracted, labeled (*True* or *False*), and compiled into an experimental dataset of cropped images.

B. Proposed Method

The proposed method consists of three main steps, as illustrated in Fig. 4.

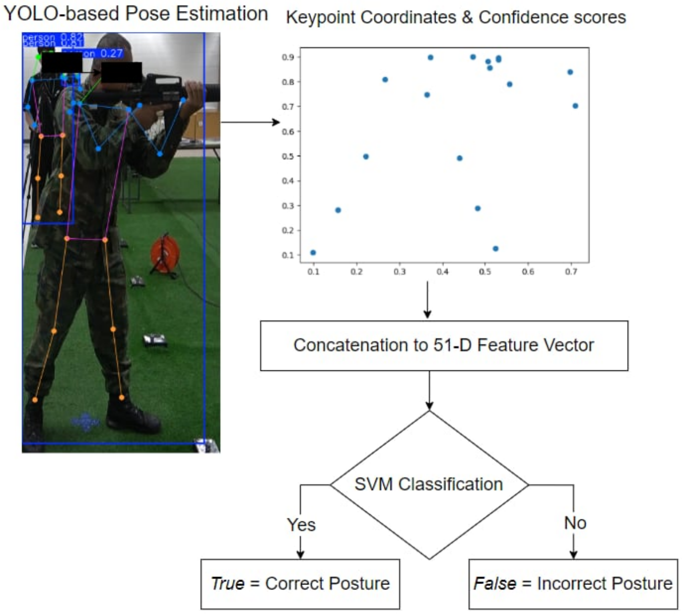
YOLO-based Pose Estimation
Keypoint Coordinates & Confidence scores
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Concatenation to 51-D Feature Vector
SVM Classification
Yes
No
True = Correct Posture
False = Incorrect Posture

Fig. 4. Illustration of the proposed method, comprising three main steps: YOLO-based pose estimation, 51-dimensional feature vector construction, and SVM-based posture classification.

- First, a cropped image of an individual is processed using the YOLO-pose model, which outputs 17 keypoints representing the body joints. Each keypoint contains three elements: two spatial coordinates (x, y) and one confidence score.
- Second, all x-coordinates, y-coordinates, and confidence scores from the 17 keypoints are concatenated to form a 51-dimensional (51-D) feature vector.
- Finally, the 51-D feature vector is fed into a support vector machine (SVM) classifier to determine whether the shooting posture in the cropped image is correct (*True*) or incorrect (*False*).

TABLE III
SAMPLE DISTRIBUTION BY FOLD AND BINARY CLASS IN THE LEFT AND RIGHT SHOOTING POSTURE DATASETS

Fold Index	**Right Dataset**		**Left Dataset**	
	True	***False***	***True***	***False***
1	560	809	916	962
2	594	782	614	898
3	567	223	528	794
4	620	749	666	848

In this experiment, cropped images were obtained from either the left or right camera, and four SVM kernels described in Section II-D were evaluated. The YOLO-pose models were selected for their advantages in real-time performance and high inference speed. Specifically, the YOLOv8n-pose model was chosen due to its widespread adoption in recent studies, while the YOLOv11n-pose model was included as the latest version of the YOLO-pose family.

C. Performance Metrics

To evaluate the performance of the proposed method, several standard performance metrics were computed based on the confusion matrix [23], [24]. The definition of the confusion matrix for the shooting posture classification task is presented in Table IV. The performance metrics used in this study include Precision, Recall, Accuracy, and F1-Score, as defined in (3)–(6). In this experiment, particular emphasis was placed on Accuracy and F1-Score, as these metrics together reflect both overall correctness and the balance between Precision and Recall, which are critical aspects for evaluating the classification model's performance.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

IV. RESULTS AND DISCUSSION

The proposed method was evaluated using 4-fold cross-validation on the Right and Left datasets, as described in Section III-A. Due to limitations in data collection and preparation, it was not feasible to construct a combined feature vector

TABLE IV
CONFUSION MATRIX FOR SHOOTING POSTURE CLASSIFICATION

Actual / Predicted	**Correct Posture**	**Incorrect Posture**
Correct Posture	TP (True Positive)	FN (False Negative)
Incorrect Posture	FP (False Positive)	TN (True Negative)

TABLE V
CONFUSION MATRICES FROM 4-FOLD CROSS-VALIDATION USING YOLOV8N-POSE AND SVM

SVM	**Right Dataset**				**Left Dataset**			
Kernel	***TP***	***FN***	***TN***	***FP***	***TP***	***FN***	***TN***	***FP***
Linear	2181	160	1856	707	2674	50	2739	763
Polynomial	2122	219	1840	723	1887	837	2747	755
RBF	2196	145	1852	711	2689	35	2731	771
Sigmoid	1180	1161	1005	1558	2724	0	47	3455

that integrates both camera views. Therefore, performance was evaluated separately for each dataset.

Tables V and VI present the confusion matrices from the binary classification of shooting postures, which included the classes *True* (correct) and *False* (incorrect), using the YOLOv8n-pose and YOLOv11n-pose models, respectively. In both cases, the YOLO-pose models extracted keypoints and their corresponding confidence scores from cropped images in the Left and Right datasets, which were then combined into feature vectors for SVM classification, as described in Section III-B. Four SVM kernels (Linear, Polynomial, RBF, and Sigmoid) were tested using the default hyperparameter settings of the *scikit-learn* SVC class [25], summarized in Table VII. The resulting performance metrics (Precision, Recall, Accuracy, and F1-Score) are reported in Table VIII.

Overall, the performance metrics obtained from the Left dataset are substantially higher than those from the Right dataset, except for the Polynomial SVM with the YOLOv8n-pose model. This discrepancy likely arises because keypoints extracted from the left camera are more reliable (less noisy) than those from the right camera. The right camera captured the front view, which includes more complex visual details that could confuse the YOLO-pose models, whereas the rear-view images from the left camera offer simpler body outlines, particularly of the arms, legs, and torso, possibly sufficient for assessing shooting posture correctness. The relatively lower performance of the Polynomial SVM with the YOLOv8n-pose model may stem from overfitting, since the Linear SVM using the same features already achieved high performance, suggesting that the Polynomial SVM might be unnecessarily complex for this dataset.

TABLE VI
CONFUSION MATRICES FROM 4-FOLD CROSS-VALIDATION USING YOLOV11N-POSE AND SVM

SVM	**Right Dataset**				**Left Dataset**			
Kernel	***TP***	***FN***	***TN***	***FP***	***TP***	***FN***	***TN***	***FP***
Linear	2095	246	1846	717	2700	24	2738	764
Polynomial	2195	146	1849	714	2632	92	2747	755
RBF	2139	202	1836	727	2690	34	2734	768
Sigmoid	1180	1161	1005	1558	2724	0	10	3492

TABLE VII
HYPERPARAMETER SETTINGS OF SVM MODELS

Kernel Type	**Hyperparameters**[a]
Linear	$C = 1.0$
Polynomial[b]	$C = 1.0, \text{degree} = 3, \text{coef0} = 0.0$
Gaussian / RBF[b]	$C = 1.0$
Sigmoid[b]	$C = 1.0, \text{coef0} = 0.0$

[a]For all cases, class_weight = ‘None’, [b]gamma = ‘scale’

Comparing YOLOv8n-pose and YOLOv11n-pose, their overall performance is comparable across all SVM kernels, except for the previously noted Polynomial case. However, YOLOv11n-pose offers superior real-time efficiency due to its lower GFLOPs (Table I), achieved through a redesigned backbone that reduces computational complexity while preserving keypoint extraction accuracy. The combination of YOLOv11n-pose (Left view) and the Linear SVM attained the highest performance, 87.34% Accuracy and 87.27% F1-score, demonstrating robustness and suitability for real-time implementation.

Across all SVM kernels, except the Polynomial case with YOLOv8n-pose (Left view), the Linear, RBF, and Polynomial SVMs produced similar results, with the Polynomial SVM performing slightly lower on average. Since the Linear SVM achieved performance comparable to the RBF SVM while offering greater interpretability, it is preferred for this task. This interpretability enhances model transparency in the classification stage following YOLO-pose feature extraction.

Overall, these results show that shooting-posture classification can be effectively achieved using YOLO-pose feature extraction with SVM classification. Although the SVM hyperparameters were not extensively optimized and the datasets remain preliminary, the performance, particularly 87.34% Accuracy and 87.27% F1-score from the Linear SVM with YOLOv11n-pose (Left view), demonstrates that the proposed approach is both feasible and promising for future work.

V. CONCLUSION AND FUTURE WORK

This study investigated the feasibility of a pose-based motion classification method combining YOLO-pose and Support Vector Machine (SVM) models for binary classification of standing shooting postures as either correct (*True*) or incorrect (*False*). Video recordings were captured from two simultaneous views, front (Right) and rear (Left), and processed separately into four-fold cross-validation datasets for each view. Results show that the Left (rear-view) dataset outperforms the Right (front-view) dataset, likely due to more reliable keypoints extracted from the rear view. An exception occurred with the Polynomial SVM using YOLOv8n-pose, possibly due to overfitting. Overall, YOLOv8n-pose and YOLOv11n-pose produced similar performance, though YOLOv11n-pose achieved higher real-time efficiency because

TABLE VIII
COMPARISON OF YOLOV8N-POSE AND YOLOV11N-POSE PERFORMANCE USING 4-FOLD CROSS-VALIDATION AND SVM CLASSIFICATION

Dataset	SVM Kernel	YOLOv8n-pose Model				YOLOv11n-pose Model			
		Precision(%)	Recall(%)	Accuracy(%)	F1-Score(%)	Precision(%)	Recall(%)	Accuracy(%)	F1-Score(%)
Right	Linear	75.52	93.17	82.32	83.42	74.50	89.49	80.36	81.31
Right	Polynomial	74.59	90.65	80.79	81.84	75.46	93.76	82.46	83.62
Right	RBF	75.54	93.81	82.54	83.69	74.63	91.37	81.06	82.16
Right[a]	Sigmoid[a]	43.10	50.41	44.56	46.47	43.10	50.41	44.56	46.47
Left	Linear	77.80	98.16	86.94	86.80	77.94	99.12	87.34	87.27
Left	Polynomial	71.42	69.27	74.43	70.33	77.71	96.62	86.40	86.14
Left	RBF	77.72	98.72	87.05	86.97	77.79	98.75	87.12	87.03
Left	Sigmoid	44.08	100.00	44.51	61.19	43.82	100.00	43.91	60.94

[a] The identical performance values for “Right–Sigmoid” in the YOLOv8n-pose and YOLOv11n-pose models were reviewed during result validation and are not typographical errors.

of its lower GFLOPs. Among all tested SVMs, the Linear and RBF SVMs delivered the best performance, with the Linear SVM achieving 87.34% Accuracy and 87.27% F1-Score using the YOLOv11n-pose (Left view), making it the preferred choice for its simplicity and interpretability. Despite the limited dataset and lack of extensive hyperparameter tuning, the results are promising. Future work will expand the dataset, include varied shooting postures, optimize SVM hyperparameters, and integrate the system into a real-time firearm training application.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Defence Technology Institute (DTI) and Armed Forces Academies Preparatory School (AFAPS) for their support and collaboration in this project.This work was supported by the project titled ”Research and Development to Enhance the Efficiency of Virtual Shooting Range Simulator,” funded by the National Research Council of Thailand (NRCT) under the fiscal year 2025 research and innovation program (Contract No. N23A680785).

REFERENCES

[1] M. Krawczyk-Suszek, B. Martowska, and R. Sapuła, “Analysis of the stability of the body in a standing position when shooting at a stationary target—a randomized controlled trial,” *Sensors*, vol. 22, no. 1, p. 368, 2022.

[2] H. Li, H. J. Yap, and S. Khoo, “Motion classification and features recognition of a traditional chinese sport (baduanjin) using sampled-based methods,” *Applied Sciences*, vol. 11, no. 16, p. 7630, 2021.

[3] S. Chen and R. R. Yang, “Pose trainer: correcting exercise posture using pose estimation,” *arXiv preprint arXiv:2006.11718*, 2020.

[4] A. Bonfiglio, D. Tacconi, R. M. Bongers, and E. Farella, “Effects of imu sensor-to-segment calibration on clinical 3d elbow joint angles estimation,” *Frontiers in Bioengineering and Biotechnology*, vol. 12, p. 1385750, 2024.

[5] V. Jakkula, “Tutorial on support vector machine (svm),” *School of EECS, Washington State University*, vol. 37, no. 2.5, p. 3, 2006.

[6] M. A. Chandra and S. Bedi, “Survey on svm and their application in image classification,” *International Journal of Information Technology*, vol. 13, no. 5, pp. 1–11, 2021.

[7] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, “Deep learning-based human pose estimation: A survey,” *ACM computing surveys*, vol. 56, no. 1, pp. 1–37, 2023.

[8] A. Singh, A. Bevilacqua, T. L. Nguyen, F. Hu, K. McGuinness, M. O’Reilly, D. Whelan, B. Caulfield, and G. Ifrim, “Fast and robust video-based exercise classification via body pose tracking and scalable multivariate time series classifiers,” *Data Mining and Knowledge Discovery*, vol. 37, no. 2, pp. 873–912, 2023.

[9] Z. Zhao, S. Kiciroglu, H. Vinzant, Y. Cheng, I. Katircioglu, M. Salzmann, and P. Fua, “3d pose based feedback for physical exercises,” in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 1316–1332.

[10] D. Maji, S. Nagori, M. Mathew, and D. Poddar, “Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2637–2646.

[11] S.-T. Tsai, Z.-R. Wu, P.-H. Lin, C.-H. Chen, W. Chien, and Y.-C. Chang, “Comparative analysis of real-time multi-person pose detection in electrical industrial safety scenarios using yolov8-pose and openpose,” in *2024 IEEE 4th International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)*. IEEE, 2024, pp. 327–330.

[12] M. Elnady and H. E. Abdelmunim, “A novel yolo lstm approach for enhanced human action recognition in video sequences,” *Scientific Reports*, vol. 15, no. 1, p. 17036, 2025.

[13] S. Cai, H. Xu, W. Cai, Y. Mo, and L. Wei, “A human pose estimation network based on yolov8 framework with efficient multi-scale receptive field and expanded feature pyramid network,” *Scientific Reports*, vol. 15, no. 1, p. 15284, 2025.

[14] Q. Dang, J. Yin, B. Wang, and W. Zheng, “Deep learning based 2d human pose estimation: A survey,” *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 663–676, 2019.

[15] G. Goyal, F. Di Pietro, N. Carissimi, A. Glover, and C. Bartolozzi, “Moveenet: Online high-frequency human pose estimation with an event camera,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4024–4033.

[16] S. Qiao, Y. Wang, and J. Li, “Real-time human gesture grading based on openpose,” in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, 2017, pp. 1–6.

[17] A. Patle and D. S. Chouhan, “Svm kernel functions for classification,” in *2013 International conference on advances in technology and engineering (ICATE)*. IEEE, 2013, pp. 1–9.

[18] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, “Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors,” *arXiv e-prints*, pp. arXiv–2411, 2024.

[19] Ultralytics. (2023) Explore ultralytics yolov8. Accessed: 2025-10-31. [Online]. Available: https://docs.ultralytics.com/models/yolov8

[20] ——. (2024) Ultralytics yolo11. Accessed: 2025-10-31. [Online]. Available: https://docs.ultralytics.com/models/yolo11

[21] D. Berrar *et al.*, “Cross-validation.” 2019.

[22] M. W. Browne, “Cross-validation methods,” *Journal of mathematical psychology*, vol. 44, no. 1, pp. 108–132, 2000.

[23] S. Sathyanarayanan and B. R. Tantri, “Confusion matrix-based performance evaluation metrics,” *African Journal of Biomedical Research*, vol. 27, no. 4S, pp. 4023–4031, 2024.

[24] A. Hay, “The derivation of global estimates from a confusion matrix,” *International Journal of Remote Sensing*, vol. 9, no. 8, pp. 1395–1398, 1988.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in python,” *The Journal of Machine Learning Research*, vol. 12, no. null, p. 2825–2830, Nov. 2011.