Optimizing Vision-Based CVS Risk Assessment: Benchmarking Geometric and Deep Learning Architectures for Real-Time Applications

MATHUROS PANMUANG [1], CHONNIKARN RODMORN [2]
[1]Department of Educational Technology and Communications, Faculty of Technical Education
Rajamangala University of Technology Thanyaburi
39 Moo 1, Klong 6, Khlong Luang, Pathum Thani 12110
THAILAND

[2]Department of Applied Statistics, Faculty of Applied Science
King Mongkut's University of Technology North Bangkok
1518 Pracharat 1 Road, Wongsawang, Bangsue, Bangkok 10800
THAILAND

Abstract: - This study benchmarks geometric and deep learning architectures for vision-based assessment of behavioral risk factors associated with Computer Vision Syndrome (CVS). Eye blinking, head pose deviation, and body twist were the three behaviors assessed. With accuracies of 0.9034 for blink detection and 0.9037 for head pose estimation, the results demonstrate that Dlib offered the most dependable performance for facial analysis. For head pose analysis, the highest F1-score was 0.7896 at a yaw threshold of 11°, with stable performance across the 9°–11° range. MediaPipe showed greater practical suitability for real-time body twist detection, with processing speeds of 29.98–32.79 FPS, although its maximum accuracy was 0.4668. In contrast, the CNN-based model was constrained by low processing speed, ranging from 4.22 to 7.32 FPS, making it less suitable for immediate deployment. These findings indicate that effective CVS risk assessment requires a balance between detection reliability and computational efficiency. Dlib is more appropriate for accuracy-sensitive facial measurements, whereas MediaPipe offers better practical value for real-time posture and movement monitoring.

Key-Words: - Computer Vision Syndrome, CVS Risk Assessment, Behavioral Risk Detection, Dlib, MediaPipe, CNN.

Received: February 2, 2026. Revised: April 7, 2026. Accepted: June 9, 2026. Published: September 17, 2026.

1 Introduction

Computer and electronic device use has become routine in work and learning settings, increasing the need to monitor Computer Vision Syndrome (CVS), which involves both ocular symptoms and musculoskeletal discomfort. CVS-related risk behaviors should be monitored continuously because early detection can support prevention and timely user feedback. A previous study reported that a reduced blink rate is directly associated with the severity of CVS, [1]. However, CVS symptoms are not limited to the eyes; they may also include neck and shoulder discomfort associated with improper head orientation and body posture. Clinical head pose estimation has been noted as a useful approach for identifying indicators of human motor control and physiological abnormalities, [2]. Therefore, an effective CVS risk assessment system should not rely on a single visual cue but should analyze multiple behavioral factors, including blinking, head orientation, and upper-body movement.

Computer vision technology has evolved significantly from traditional geometric approaches, such as Dlib-based landmark detection, to neural-network-based methods and modern frameworks such as MediaPipe. For blink-related detection, [3], compared CNN, MediaPipe, and Dlib, reporting that Dlib can be limited under varying illumination and face-angle conditions, while CNN and MediaPipe provide more stable performance. For head pose estimation, [4], confirmed that MediaPipe algorithms are more robust against sudden movement and occlusion than Dlib. This observation is consistent with, [5], which found that MediaPipe is effective in capturing 3D facial landmarks during movement.

In parallel, machine learning and deep learning approaches have continued to advance. An SVM-based approach using glabellar length features was proposed, [6]. while advanced deep learning models, including EyeNet and Auto-Encoder Feature Union (AEFU), were introduced, [7], [8], to improve detection accuracy. Furthermore, model interpretability has become increasingly important in medical and health-related detection systems.

Although computer vision techniques have advanced rapidly, further comparative evaluation is still needed to determine which architecture is most appropriate for practical CVS risk assessment, particularly in resource-limited settings. Previous studies have often focused on detection accuracy, whereas latency, computational demand, and real-time feasibility have been examined less extensively. However, these factors are essential when a system is intended for deployment on standard consumer devices. Recent studies have increasingly highlighted this issue. A benchmark comparison of YOLO architectures showed that appropriately sized models can achieve high accuracy while maintaining real-time processing speed, [9]. This finding is consistent with evidence that Random Forest combined with MediaPipe features can outperform heavier deep learning models in both accuracy and speed, [10].

Building upon this research direction, the present study extends lightweight and real-time model benchmarking to the specific context of CVS risk assessment. Rather than reusing previously published datasets or experimental results, this work provides new task-specific benchmarking data for CVS-related behavioral detection, including blink detection, head pose estimation, and body twist detection. The study compares three primary architectures—Dlib, MediaPipe, and CNN—across these behavioral dimensions and evaluates their trade-offs in terms of accuracy, precision, recall, F1-score, and FPS. In addition, threshold sensitivity analysis is conducted to identify balanced configurations for real-time use. The main contribution of this study is therefore a systematic benchmarking and threshold optimization framework for selecting appropriate architecture for a practical, real-time CVS risk warning system on standard consumer devices.

The main contributions of this study are summarized as follows. First, this study presents a systematic benchmarking framework for CVS-related behavioral risk assessment by comparing Dlib, MediaPipe, and CNN across three detection tasks: blink detection, head pose estimation, and body twist detection. Second, new experimental benchmarking data are provided using task-specific datasets rather than reusing previously published datasets or experimental results. Third, the study introduces F1-score-based parameter optimization for Eye Aspect Ratio (EAR) Threshold, Yaw Threshold, Twist Threshold, and Lean Threshold parameters. Finally, the study evaluates both detection performance and real-time feasibility using classification metrics and FPS, supporting the selection of practical architecture for real-time CVS risk warning systems on standard consumer devices.

2 Related Works

This section reviews the evolution of vision-based monitoring systems, ranging from ocular fatigue detection to comprehensive posture analysis. It highlights the transition from traditional geometric approaches to modern deep learning frameworks and discusses the trade-offs between accuracy and computational efficiency in the context of Computer Vision Syndrome (CVS) and drowsiness detection.

2.1 Vision-based Systems for CVS Monitoring

The proliferation of digital devices has necessitated the development of non-intrusive systems to monitor visual fatigue. A previous exploratory study, [1], established the physiological foundation for these systems by investigating blink behavior in real-life settings. The study confirmed that a decrease in blink rate, approximately 9–17 blinks per minute was significantly correlated with the severity of CVS symptoms, thereby supporting the use of webcam-based monitoring as a reliable approach for risk assessment. Expanding on feature extraction, various eye-tracking features, including pupil size and saccade distance, were explored, [8]. An Auto-Encoder Feature Union (AEFU) algorithm was proposed, demonstrating that combining manually extracted features with deep learning representations significantly enhances fatigue detection accuracy compared to using raw signals alone.

In terms of application, several platforms have been proposed. A web-based platform called "iSVC" was developed to utilizing the Dlib library to calculate the EAR for preventing CVS, [11]. While effective, their system relied on fixed thresholds, which may lack adaptability. In the context of online learning, EyeNet, a lightweight model optimized through knowledge distillation

from ResNet-18, was proposed to monitor student fatigue in real time on resource-constrained devices, [7]. Nevertheless, many existing systems rely primarily on blink rate as a single indicator, with limited consideration of broader behavioral patterns such as head orientation and body posture, which are important for comprehensive CVS assessment.

2.2 Eye State Analysis Techniques

The methodology for detecting eye states has evolved from geometric calculations to advanced neural networks. The effectiveness of Support Vector Machines (SVM) using facial landmarks has been demonstrated for detecting frowning associated with strain, [6]. Novel facial features, such as glabellar length, were introduced to support strain-related detection. Similarly, accessibility has been focused on the implementation of a YOLOv5-based blink counter on a Raspberry Pi, [12]. This demonstrates that deep learning can be adapted for low-cost embedded systems.

However, the robustness of these algorithms varies under different environmental conditions. A comparative evaluation of CNN, MediaPipe, and Dlib was conducted, [3]. The findings showed that although Dlib performs well in controlled settings, it is highly sensitive to face angles and lighting variations. In contrast, CNN and MediaPipe exhibited superior robustness, maintaining high accuracy even in challenging lighting. To further mitigate false positives in real-time detection, A hybrid approach combining MediaPipe-based feature extraction with Fuzzy Logic was proposed, [13]. This approach effectively filters out noise that traditional binary classifiers may misinterpret. Despite these advancements, there remains a lack of systematic sensitivity analysis regarding optimal thresholding parameters for these algorithms, specifically within the context of CVS, where micro-expressions like partial blinks are common.

2.3 Head and Body Pose Estimation

Since CVS encompasses musculoskeletal symptoms, such as neck and shoulder pain, monitoring head and body posture is crucial. Although optoelectronic motion-capture systems are considered the gold standard for kinematic assessment, they are impractical for daily use. Video-based pose estimation algorithms, including OpenFace, MediaPipe, and 3DDFA_V2, have been validated against an optoelectronic baseline, [2]. The results demonstrated that modern RGB-based algorithms can achieve clinically acceptable accuracy, with an error of less than five degrees, thereby supporting their use in non-clinical settings.

Regarding algorithm robustness under challenging conditions, Dlib and MediaPipe were compared in scenarios involving facial occlusions and spontaneous movements, [4]. The results confirmed that MediaPipe maintains higher accuracy and stability than Dlib when the face is partially obstructed or moving rapidly. Furthermore, for full-body pose estimation, the depth-ambiguity problem inherent in 2D images has been addressed using an optimization method based on a humanoid model combined with MediaPipe Pose, [14]. This approach enables more accurate estimation of 3D joint angles and supports the analysis of complex body activities. While these studies validate the accuracy of pose estimation, few have explicitly benchmarked the computational cost (latency) of these methods when running concurrently with eye-tracking modules on standard consumer hardware.

2.4 Comparative Analysis of Architectures

A comparative evaluation of Convolutional Neural Networks (CNNs) and computer vision approaches using Dlib and MediaPipe was conducted for driver-assistance applications, [5]. The results indicated that although CNNs are powerful, MediaPipe provides a more efficient solution for extracting 3D facial key-point coordinates in real-time applications.

Most notably, transfer learning models, including VGG16 and ResNet50, were benchmarked against traditional machine learning classifiers, including Random Forest and XGBoost, using features extracted by MediaPipe, [10]. The results indicated that a Random Forest classifier trained on MediaPipe features achieved the highest accuracy of 97% and lower latency than heavier deep learning models, such as ResNet50. This finding challenges the assumption that "deeper is always better" and supports investigation into lightweight, geometry-based architecture for real-time risk assessment systems like CVS monitoring. Consequently, this study aims to bridge these gaps by conducting a comprehensive benchmarking of Geometric versus Deep Learning architectures, specifically focusing on the trade-offs between detection sensitivity and computational latency for multi-modal CVS risk factors.

3 Methodology

This study employs a systematic process to benchmark and optimize computer vision architectures for CVS risk assessment. The methodology consists of four components: dataset preparation, sensitivity analysis and threshold optimization, architecture benchmarking and model selection, and statistical performance evaluation.

3.1 Dataset Preparation and Experimental Setup

To simulate real-world Computer Vision Syndrome (CVS) risk scenarios, a custom dataset was constructed from video data collected from 10 volunteers, consisting of 5 males and 5 females. Each participant recorded one continuous video clip simulating computer usage behavior, with each clip lasting 1 minute, resulting in a total of 10 minutes of video footage. The recorded videos served as the common experimental source for three CVS-related behavioral detection tasks: blink detection, head pose estimation, and body twist detection.

For the CNN-based approaches, the recorded videos were cropped into task-specific static image datasets. Three separate image datasets were prepared according to the target behavior. The blink detection dataset consisted of two classes: closed eyes and open eyes. The head pose estimation dataset consisted of three classes: looking forward, looking left, and looking right. The body twist detection dataset included three classes: sitting straight, twisting left, and twisting right. Each class provided at least 3,000 images for model training and evaluation. The CNN datasets were then split into training, validation, and testing subsets at an 80:10:10 ratio. Training data were used to learn model parameters, validation data supported performance monitoring and overfitting control, and the test data were reserved for final evaluation. The testing subset was not used during model training, threshold tuning, or hyperparameter selection.

For the Dlib- and MediaPipe-based approaches, no model retraining or fine-tuning was performed. Dlib employed a pre-trained 68-point facial landmark predictor for blink detection and head pose estimation, while MediaPipe employed a pre-trained Face Mesh for facial landmark extraction and MediaPipe Pose for body landmark extraction. The original video streams were processed directly, and the extracted landmarks were converted into task-specific features, including the Eye Aspect Ratio (EAR) for blink detection, Yaw-angle-based indicators for head pose estimation, and the shoulder-depth-based Twist Index for body twist detection.

For Dlib and MediaPipe, the training subset was not used because these methods rely on pre-trained landmark detectors and rule-based thresholding rather than supervised model training. Their threshold-based parameters were selected using validation data and then evaluated using testing data. This design ensured a consistent evaluation protocol across all architectures while clearly distinguishing between trainable CNN-based models and pre-trained landmark-based approaches.

Ground-truth labels served as the reference for calculating the confusion matrix, accuracy, precision, recall, F1-score, and inference speed in frames per second (FPS) for each task. All experiments were run on a standard workstation with an AMD Ryzen 7 4800H processor operating at 2.90 GHz and 16 GB RAM, without a discrete GPU.

3.2 Sensitivity Analysis and Threshold Optimization

The objective of this section was to determine the optimal threshold parameters for the landmark-based detection tasks. Sensitivity analysis was conducted by systematically varying task-specific threshold values over predefined candidate sets. For each candidate configuration, precision, recall, and F1-score were measured using the validation subset. The configuration with the highest F1-score was selected, as this metric accounts for both false detections and missed events, making it suitable for imbalanced behavioral detection tasks.

Blink Detection Optimization: Blink detection relies on the eye aspect ratio (EAR) method for eye-closure analysis, [15]. Let $p_1, p_2, \ldots p_6$ denote the six eye landmarks. The EAR is defined as:

$$EAR = \frac{\|p2-p6\|+\|p3-p5\|}{2\|p1-p4\|} \quad (1)$$

A blink event is detected when the EAR value remains below a predefined threshold τ_{EAR} for a specified number of consecutive frames c. Instead of using a fixed empirical threshold, the EAR threshold and the consecutive-frame parameter were jointly optimized to maximize the balance between precision and recall. The candidate EAR threshold set was defined as $T_{EAR} = \{0.01, 0.02, 0.03, 0.05, 0.10, 0.15\}$, and the candidate consecutive-frame set was defined as $C = \{1, 2\}$. The optimal blink detection configuration was selected

by maximizing the F1-score on the validation subset, as shown in Equation (2):

$$(\tau_{EAR}^{*}, c^{*}) = \arg \max_{\tau \in T_{EAR},\, c \in C} \left(\frac{2 \cdot \text{Precision}(\tau,c) \cdot \text{Recall}(\tau,c)}{\text{Precision}(\tau,c) + \text{Recall}(\tau,c)} \right) \quad (2)$$

Head Pose Sensitivity: For head pose estimation, the yaw angle $\theta_y(t)$ was extracted from facial landmarks and compared against a Yaw Threshold τ_{yaw}. In this study, head orientation was classified into three classes: looking left, looking right, and looking forward. The decision rule is defined in Equation (3):

$$H(t) = \begin{cases} Left, & \theta_y(t) < -\tau_{yaw} \\ Right, & \theta_y(t) > \tau_{yaw} \\ Forward, & |\, \theta_y(t) \,| \leq \tau_{yaw} \end{cases} \quad (3)$$

The candidate Yaw Threshold set was defined as:

$$T_{yaw} = \{3°, 5°, 7°, 9°, 11°, 13°, 15°\} \quad (4)$$

The optimal Yaw Threshold was selected by maximizing the F1-score on the validation subset, as shown in Equation (5):

$$\tau_{yaw}^{*} = \arg \max_{\tau \in T_{yaw}} F1(\tau) \quad (5)$$

where $F1(\tau)$ denotes the F1-score obtained for each candidate Yaw Threshold τ. This formulation enables the optimal threshold to be selected based on the balance between false head-turn detections and missed head-turn events.

Body Twist Sensitivity and Twist Index Formulation: For body twist detection, MediaPipe Pose was used to extract the left and right shoulder landmarks. Let $p_{ls} = (x_{ls}, y_{ls}, z_{ls})$and $p_{rs} = (x_{rs}, y_{rs}, z_{rs})$denote the left and right shoulder landmarks, respectively. The 2D shoulder distance on the image plane is defined as:

$$D_{shoulder} = \sqrt{(x_{ls} - x_{rs})^2 + (y_{ls} - y_{rs})^2} \quad (6)$$

The Twist Index T_w is computed by normalizing the absolute relative-depth difference between the left and right shoulders by the 2D shoulder distance:

$$T_w = \frac{|z_{ls} - z_{rs}|}{D_{shoulder} + \epsilon} \quad (7)$$

where z_{ls} and z_{rs} are the relative depth values of the left and right shoulder landmarks, and ϵ prevents division by zero. A larger shoulder-depth difference indicates greater upper-body rotation. Dividing this value by $D_{shoulder}$ adjusts the index for differences in body size, camera distance, and image resolution. This normalization reduces the influence of subject scale, camera distance, and image resolution, allowing the Twist Index to represent relative body rotation rather than absolute pixel displacement.

A body twist event is detected when the Twist Index exceeds a predefined Twist Threshold τ_{twist}. In addition, a Lean Threshold τ_{lean} was considered to account for lateral upper-body displacement. The candidate threshold sets were defined as:

$$T_{twist} = \{0.01, 0.03, 0.05, 0.08, 0.10, 0.15, 0.20, 0.25\} \quad (8)$$

and

$$T_{lean} = \{0.05, 0.10, 0.15, 0.25\} \quad (9)$$

The optimal body twist configuration was selected by maximizing the F1-score on the validation subset:

$$\left(\tau_{twist}^{*}, \tau_{lean}^{*}\right) = \arg \max_{\tau_t \in T_{twist},\, \tau_l \in T_{lean}} F1(\tau_t, \tau_l) \quad (10)$$

where $F1(\tau_t, \tau_l)$ denotes the F1-score obtained from a given Twist Threshold $\tau_t$and Lean Threshold τ_l.

3.3 Architecture Benchmarking and Model Selection

After defining the threshold optimization procedure, the selected computer vision architectures were benchmarked across the three CVS-related behavioral detection tasks. For blink detection, Dlib, MediaPipe, and the CNN-based model were compared. Dlib and MediaPipe relied on landmark-based EAR computation, whereas the CNN model performed image-based classification between open-eye and closed-eye states.

For head pose estimation, Dlib, MediaPipe, and the CNN-based model were evaluated for three-class head orientation classification, consisting of looking forward, looking left, and looking right. Dlib and MediaPipe used landmark-based Yaw-angle estimation, while the CNN-based model classified cropped facial images into the corresponding head orientation classes.

For body twist detection, MediaPipe Pose and the CNN-based model were compared. Dlib was excluded from this task because it does not provide full-body posture landmarks. MediaPipe Pose estimated body landmarks and computed the Twist Index, whereas the CNN model classified body posture images into sitting straight, twisting left, and twisting right classes.

Model selection was performed using F1-score as the primary criterion because the behavioral classes were imbalanced and accuracy alone could overestimate performance. FPS was used as a secondary criterion to assess real-time feasibility. Therefore, the selected architecture for each task was the configuration that achieved the most appropriate trade-off between detection reliability and computational efficiency.

3.4 Statistical Performance and Real-Time Efficiency Analysis

To evaluate both detection reliability and real-time feasibility, each architecture was assessed using classification performance metrics and computational efficiency indicators. The experiments were conducted using the recorded video dataset and the independent testing subsets prepared for the CNN-based models. Although the experiments were conducted offline, the measured inference speed provides an indicator of the system's potential for real-time deployment on general-purpose hardware.

Classification Metrics: Since the behavioral classes were imbalanced, accuracy was reported together with precision, recall, and F1-score to provide a more complete evaluation of model performance. These metrics were calculated for each experimental task and are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

where TP, TN, FP, and FNrepresent true positives, true negatives, false positives, and false negatives, respectively. The F1-score was used as the primary criterion for threshold optimization and model selection because it provides a balanced measure between precision and recall, particularly in imbalanced behavioral detection tasks.

Head Pose Error Margin: For head pose estimation, the error analysis was based on the discrepancy between the annotated head-orientation labels and the predicted orientation classes. The error analysis was based on misclassified samples in each head pose class, including looking forward, looking left, and looking right. False positives, false negatives, precision, recall, and F1-score were reported for each class.

Computational Latency: Real-time feasibility was examined using inference speed in frames per second (FPS). For each architecture, FPS was calculated from the processing time measured across a continuous sequence of $N = 2000$ frames. FPS was calculated as follows:

$$FPS = \frac{N}{\sum_{i=1}^{N} T_i / 1000} \tag{15}$$

where $\boldsymbol{T_i}$ denotes the processing time of frame $\boldsymbol{i}$in milliseconds, and $\boldsymbol{N}$denotes the total number of processed frames. In this study, $\boldsymbol{N}$ was set to 2000 to obtain a stable measurement of inference speed. In addition, CPU utilization was monitored to assess computational resource consumption. The final model selection considered F1-score as the primary indicator of detection reliability and FPS as the secondary indicator of real-time feasibility, ensuring that the selected architecture was both sufficiently accurate and computationally efficient for general-purpose devices.

4 Results and Discussion

This study evaluated and compared the performance of three computer vision architectures: Dlib, MediaPipe, and CNN. The evaluation covered three CVS-related behavioral detection tasks: blink detection, head pose estimation, and body twist detection. The analysis focused on both detection reliability and real-time feasibility, using Accuracy, Precision, Recall, F1-score, and FPS as evaluation indicators. In line with the methodology, F1-score was used as the primary criterion for selecting balanced configurations because the behavioral classes were imbalanced, while FPS was used as a secondary criterion for assessing real-time applicability.

In the performance testing of the CNN-based models, CNN differed fundamentally from the landmark-based approaches used in Dlib and MediaPipe. CNN models were trained using task-specific image datasets, whereas Dlib and MediaPipe were evaluated using pre-trained landmark extraction frameworks and threshold-based decision rules. Therefore, CNN performance depended mainly on the training process, including the number of epochs and data augmentation techniques, while Dlib and MediaPipe performance depended mainly on the selected threshold parameters. The CNN values reported in this

section represent the results obtained from the best-performing trained models.

4.1 Parameter Sensitivity Analysis for Blink Detection

The blink detection performance of Dlib, MediaPipe, and the CNN-based model under varying EAR threshold values and consecutive-frame settings is compared in Table 1 (Appendix).

According to Table 1 (Appendix), the Dlib-based EAR approach provided the most balanced blink detection performance among the tested architectures. At an EAR threshold of 0.01 with one consecutive frame, the model obtained the highest accuracy. However, the best F1-score was obtained with an EAR threshold of 0.02 with one consecutive frame. Because it produced an accuracy of 0.9034, a precision of 0.6269, a recall of 0.1927, and an F1-score of 0.2948, this configuration was selected for blink detection. Increasing the consecutive-frame parameter to 2 generally reduced Recall and F1-score for Dlib, suggesting that blink events in this dataset occurred rapidly and could be missed when more consecutive frames were required. This finding aligns with [3], who noted that Dlib can provide accurate results in controlled environments but remains sensitive to environmental factors and threshold variations when parameters are not properly optimized.

MediaPipe achieved substantially higher FPS than Dlib, with frame rates above 60 FPS across all tested configurations. However, its Recall and F1-score remained very low across all EAR threshold settings, with the highest F1-score reaching only 0.0084. This indicates limited sensitivity to blink events in this dataset. Therefore, although MediaPipe is computationally efficient, its blink detection reliability was insufficient under the tested conditions. This finding is consistent with previous work showing that MediaPipe performs efficiently in real-time facial landmark detection, [5], However, detailed or subtle facial movements may remain challenging depending on the task and dataset.

The CNN-based model achieved an Accuracy of 0.8865 but produced a very low Recall of 0.0046 and an F1-score of 0.0084, with the lowest FPS among the compared methods at 4.22 FPS. This indicates that, in the present experimental setting, the CNN-based model showed limited suitability for real-time blink detection. The result suggests that image-based CNN classification may require larger or more diverse training data to improve generalization for rapid blink events. Overall, the Dlib-based EAR method provided the most balanced trade-off between detection reliability and real-time feasibility for blink detection.

4.2 Head Pose Estimation Performance Comparison

Table 2. (Appendix) presents the performance comparison of Dlib, MediaPipe, and the CNN-based model for head pose estimation. Dlib and MediaPipe were evaluated using candidate Yaw Threshold values ranging from 3° to 15°, while the CNN-based model was evaluated as an image-based three-class classifier without threshold adjustment. The results demonstrate how variations in the Yaw Threshold affected Accuracy, Precision, Recall, F1-score, and FPS

Table 2. (Appendix) presents the performance comparison for head pose estimation using different Yaw Threshold values. The results show that Dlib achieved the strongest overall performance among the compared architectures. The optimal Dlib configuration was found at a yaw threshold of 11° using the F1-score-based selection criterion specified in the technique. Dlib obtained an accuracy of 0.9037, precision of 0.8391, recall of 0.7456, F1-score of 0.7896, and FPS of 10.70 with this cutoff. With an accuracy of 0.9005 and an F1-score of 0.7874, the 9° threshold likewise performed similarly. According to these findings, a yaw threshold of 9° to 11° offers a good compromise between minimizing false detections from minute, normal head motions and identifying significant head rotations

MediaPipe achieved higher inference speed than Dlib, with FPS values of approximately 75–79 across the tested yaw thresholds. Its best F1-score was obtained at a yaw threshold of 3°, where Accuracy, Precision, Recall, F1-score, and FPS were 0.8511, 0.6028, 0.5463, 0.5732, and 76.30, respectively. MediaPipe is more suitable for applications where a high frame rate is required, while Dlib is better suited to tasks that require more reliable classification.

The CNN-based model's results were 0.8033 for accuracy, 0.2701 for precision, 0.3333 for recall, 0.2984 for F1-score, and 6.54 for FPS. Its F1-score and processing speed were still behind the best Dlib and MediaPipe settings, even if its accuracy was comparable to certain MediaPipe results. In this experiment, the CNN approach was therefore less suitable for real-time head pose estimation. A larger and more varied training dataset may be needed to improve three-class head orientation classification. This indicates that, in the present experimental setting, the CNN-based model

showed limited suitability for real-time head pose estimation. The result suggests that image-based classification may require larger or more diverse training data to improve generalization for reliable three-class head orientation estimation.

These findings support the importance of head pose estimation for CVS-related behavioral risk assessment. Previous studies have emphasized that head pose detection can provide useful information for assessing posture-related discomfort, including neck and back pain, [2], [4]. The present results further show that Dlib provides stronger classification reliability for head orientation based on yaw-angle estimation, whereas MediaPipe provides higher real-time efficiency.

This finding is consistent with previous work showing that MediaPipe is effective for real-time landmark detection, [5]. However, it may still have limitations in fine-grained head orientation analysis. Furthermore, the lower FPS of the CNN-based model supports previous observations that deep learning models may be more suitable for tasks that do not require immediate real-time responsiveness, [12].

From the perspective of benchmarking novelty, the results in Table 2 (Appendix) provide task-specific evidence for selecting lightweight computer vision architectures in CVS risk assessment. Rather than evaluating head pose estimation only by Accuracy, this study compares Dlib, MediaPipe, and CNN using Accuracy, Precision, Recall, F1-score, and FPS under a unified protocol. This supports the contribution of the study by providing new benchmarking evidence for real-time CVS-related head pose estimation.

4.3 Body Twist Detection Performance Comparison

Body twist detection is important for evaluating upper-body movement as a behavioral indicator related to posture risk in Computer Vision Syndrome (CVS). Such movement may be associated with neck, shoulder, and back discomfort during prolonged computer use or improper sitting posture. This study compares the performance of MediaPipe Pose and the CNN-based model for body twist detection. Dlib was excluded from this task because it provides facial landmarks but does not provide full-body posture landmarks. MediaPipe Pose was evaluated using different Twist Threshold and Lean Threshold values, whereas the CNN-based model was evaluated as an image-based body posture classifier. The results are shown in Table 3 (Appendix).

The results indicate that MediaPipe Pose achieved higher FPS than the CNN-based model across all tested threshold settings, indicating better real-time feasibility for body landmark analysis. Based on Accuracy alone, the highest value was 0.4668 at a Twist Threshold of 0.25 with a Lean Threshold of 0.05 or 0.25. However, model selection was based on the F1-score due to the unbalanced behavioral classes. The MediaPipe configurations with the highest F1-score were identified using a Twist Threshold of 0.10 and a Lean Threshold of 0.05 or 0.15. The accuracy, precision, recall, and F1-score for this arrangement were 0.4176, 0.3167, 0.2955, and 0.3057, respectively.

A Twist Threshold of 0.10 gave the most balanced MediaPipe result for body twist detection, particularly in terms of Precision and Recall. Higher Twist Threshold values, such as 0.25, improved Accuracy but reduced Recall and F1-score. This pattern suggests that stricter twist thresholds may fail to capture some body twist events. The results also show that changes in the Lean Threshold had a limited influence on the final classification performance compared with the Twist Threshold, indicating that the shoulder-depth-based Twist Index played a more direct role in body twist detection in this dataset.

The CNN-based model achieved an Accuracy of 0.4023, a precision of 0.3158, a recall of 0.3520, an F1-score of 0.3329, and a FPS of 7.32. Although it produced the highest F1-score in Table 3 (Appendix), its FPS was much lower than that of MediaPipe Pose. This result indicates that CNN may provide better classification balance for body posture images in the current dataset, but its low inference speed limits its suitability for real-time body twist detection. In contrast, MediaPipe Pose provided a more practical trade-off for real-time deployment because it maintained competitive classification performance while achieving approximately 30–32 FPS across the tested configurations.

Body twist detection is a relevant component of CVS risk assessment because upper-body movement and improper posture may contribute to neck, shoulder, and back discomfort. Body movement and posture-related behavior detection has been emphasized as important for preventing musculoskeletal discomfort and promoting long-term postural correctness, [4], [14]. In this study, MediaPipe Pose was considered more practical for real-time body twist detection because it offered full-body landmark extraction and maintained substantially higher processing speed than the

CNN-based model. This finding is consistent with previous work showing that MediaPipe is efficient for pose detection, particularly in scenarios where processing speed is important, [5].

Table 3 (Appendix) provides task-specific benchmarking evidence for body twist detection in CVS risk assessment. Unlike previous studies focusing mainly on facial behaviors or general pose detection, this study compares MediaPipe Pose and the CNN-based model under a unified protocol. It also examines Twist Threshold and Lean Threshold sensitivity, offering practical guidance for parameter selection in real-time CVS risk warning systems

5 Conclusion and Future Work

This study evaluated three computer vision architectures for CVS-related behavior detection: blink detection, head pose estimation, and body twist detection. The findings show that model suitability varies by task, particularly in the balance between detection accuracy and real-time feasibility.

For blink detection, the Dlib-based EAR approach gave the most balanced result. The selected setting was an EAR threshold of 0.02 with one consecutive frame, producing an Accuracy of 0.9034, a precision of 0.6269, a recall of 0.1927, an F1-score of 0.2948, and a FPS of 10.54. MediaPipe was faster, but its low Recall and F1-score suggest limited sensitivity to blink events in this dataset.

For head pose estimation, Dlib performed best overall. A yaw threshold of 11° produced an Accuracy of 0.9037, a precision of 0.8391, a recall of 0.7456, an F1-score of 0.7896, and a FPS of 10.70. The comparable result at 9° suggests that the 9°–11° range is appropriate for detecting meaningful head turns while reducing false positives.

For body twist detection, MediaPipe Pose was more practical for real-time use. It maintained approximately 31–32 FPS, with its best F1-score of 0.3057 at a Twist Threshold of 0.10 and a Lean Threshold of 0.05 or 0.15. Although the CNN-based model achieved a higher F1-score of 0.3329, its speed was limited to 7.32 FPS.

Overall, no single architecture consistently outperformed the others across all CVS-related behaviors. Dlib was more effective for facial landmark-based tasks, particularly blink detection and head pose estimation. MediaPipe Pose was more practical for real-time body twist detection because of its higher FPS and full-body landmark capability. CNN-based models showed potential for image-based posture classification but were limited by low inference speed. These findings support task-specific architecture selection for CVS-related behavior detection systems.

Future work should extend this benchmarking study toward an integrated real-time CVS risk assessment framework. Additional factors, including screen distance, vertical and horizontal viewing deviation, gaze direction, blink rate, eye movement patterns, and continuous usage duration, should be incorporated to support multi-factor risk assessment. Future studies should also examine session-based behavioral patterns over time and evaluate the framework in real-user environments with diverse lighting conditions, camera angles, user distances, and working postures. This would help assess robustness and practical applicability for real-time CVS risk warning systems.

Acknowledgement:
The authors sincerely thank the Department of Educational Technology and Communications, Faculty of Technical Education, Rajamangala University of Technology Thanyaburi, and the Department of Applied Statistics, Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, for providing facilities, equipment, and support during the conduct of this study.

References:

[1] I. Lapa, S. Ferreira, C. Mateus, N. Rocha, and M. Rodrigues, "Real-Time Blink Detection as an Indicator of Computer Vision Syndrome in Real-Life Settings: An Exploratory Study," Int. J. Environ. Res. Public. Health, vol. 20, no. 5, p. 4569, Mar. 2023, doi: 10.3390/ijerph20054569.

[2] Y. Hammadi, F. Grondin, F. Ferland, and K. Lebel, "Evaluation of Various State of the Art Head Pose Estimation Algorithms for Clinical Scenarios," Sensors, vol. 22, no. 18, p. 6850, Sep. 2022, doi: 10.3390/s22186850.

[3] J. M. Park, D. J. Hwang, Y. C. Hwang, J. M. Lee, H. Y. Shin, K. D. Jung, and C. Y. Go, "The Impact of Face Angle and Lighting Changes on Eye State Recognition Accuracy: A Comparative Evaluation of CNN, MediaPipe, and Dlib Performance," Int. J. Adv. Smart Converg., vol. 13, no. 4, pp. 210–221, Dec. 2024, doi: 10.7236/IJASC.2024.13.4.210.

[4] R. Yahya, R. Jailani, N. K. Zakaria, and F. A. Hanapiah, "Dynamic head pose estimation in varied conditions using Dlib and MediaPipe," Int. J. Electr. Comput. Eng. IJECE, vol. 15, no. 5, pp. 4581-4592, Oct. 2025, doi: 10.11591/ijece.v15i5.pp4581-4592.

[5] A. K. Reddy, I. A. Khan, N. C. T., L. L. Thampi, A. M. A., A. Kumar, and S. Kumar, "Smart Driver Assistance: Real-Time Drowsiness Detection Leveraging Facial Cues with MediaPipe and OpenCV," Jul. 19, 2024, Preprint - In Review. doi: 10.21203/rs.3.rs-4642662/v1.

[6] R. Kaur and A. Guleria, "Digital Eye Strain Detection System Based on SVM," in 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India: IEEE, Jun. 2021, pp. 1114–1121. doi: 10.1109/ICOEI51242.2021.9453085.

[7] Q. T. Le, D. C. Duong, M. H. Vu, T. V. Le, T. N. Doan, H. H. Dinh, and N. N. Nguyen, "Eye Strain Detection During Online Learning," Intell. Autom. Soft Comput., vol. 35, no. 3, pp. 3517–3530, 2023, doi: 10.32604/iasc.2023.031026.

[8] W. Sun, Y. Wang, B. Hu, and Q. Wang, "Exploration of Eye Fatigue Detection Features and Algorithm Based on Eye-Tracking Signal," Electronics, vol. 13, no. 10, p. 1798, May 2024, doi: 10.3390/electronics13101798.

[9] S. Intagorn, M. Panmuang, C. Rodmorn, and S. Pinitkan. "Evaluating the Performance of YOLO Architectures for Effective Gun and Knife Detection," Engineering Access, 11(2), 170–177. doi: 10.14456/MIJET.2025.16.

[10] P. Joshi, M. Adhikari, S. Shrestha, and S. Shaik, "Real-Time Driver Drowsiness Detection Using CNN, MediaPipe, and ML Classifiers," in SoutheastCon 2025, Concord, NC, USA: IEEE, Mar. 2025, pp. 589–594. doi: 10.1109/SoutheastCon56624.2025.10971270

[11] F. Vieira, E. Oliveira, and N. Rodrigues, "iSVC – Digital Platform for Detection and Prevention of Computer Vision Syndrome," in 2019 IEEE 7th International Conference on Serious Games and Applications for Health (SeGAH), Kyoto, Japan: IEEE, Aug. 2019, pp. 1–7. doi: 10.1109/SeGAH.2019.8882438.

[12] M. T. I. M. Carcellar, C. J. S. Tychuaco, and A. N. Yumang, "Self-Applicable Eye Strain Detection Through the Measurement of Blink Rate Using Raspberry Pi," in 2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia: IEEE, Aug. 2024, pp. 13–17. doi: 10.1109/IICAIET62352.2024.10730733.

[13] S. Dalve, I. Ramdasi, G. Kothawade, Y. Khadke, and M. Wete, "Real Time Prevention of Driver Fatigue Using Deep Learning and MediaPipe," Int. J. Innov. Res. Comput. Sci. Technol., vol. 11, no. 3, pp. 7–11, May 2023, doi: 10.55524/ijircst.2023.11.3.2.

[14] J.W. Kim, J.Y. Choi, E.J. Ha, and J.H. Choi, "Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model," Appl. Sci., vol. 13, no. 4, p. 2700, Feb. 2023, doi: 10.3390/app13042700.

[15] T. S. Jan Cech, "Eye-Blink Detection Using Facial Landmarks," in 21st computer vision winter workshop, 2016, [Online]. https://api.semanticscholar.org/CorpusID:35923299 (Accessed Date: December 14, 2025).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The research was funding by King Mongkut's University of Technology North Bangkok Contract no. KMUTNB-69-KNOW-08.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

APPENDIX

Table 1. Performance Comparison of Models for Blink Detection

Model Architecture	Threshold (EAR)	Consecutive Frames	Accuracy	Precision	Recall	F1-score	FPS
Dlib	0.01	1	0.9077	0.8095	0.1560	0.2616	9.58
	0.02	**1**	**0.9034**	**0.6269**	**0.1927**	**0.2948**	**10.54**
	0.03	1	0.8938	0.4746	0.1284	0.2021	10.52
	0.05	1	0.8894	0.3333	0.0550	0.0944	10.23
	0.10	1	0.8721	0.0200	0.0046	0.0075	10.44
	0.15	1	0.8524	0.0000	0.0000	0.0000	10.49
	0.01	2	0.8976	0.7273	0.0367	0.0699	10.38
	0.02	2	0.8981	0.5938	0.0872	0.1521	10.96
	0.03	2	0.8923	0.4167	0.0688	0.1181	10.83
	0.05	2	0.8918	0.3871	0.0550	0.0963	10.22
MediaPipe	0.01	1	0.8841	0.0400	0.0046	0.0083	63.90
	0.02	1	0.8832	0.0370	0.0046	0.0082	70.52
	0.03	1	0.8851	0.0000	0.0000	0.0000	70.78
	0.05	1	0.8856	0.0000	0.0000	0.0000	65.80
	0.10	1	0.8635	0.0000	0.0000	0.0000	69.01
	0.15	1	0.8827	0.0000	0.0000	0.0000	71.73
	0.01	**2**	**0.8865**	**0.0500**	**0.0046**	**0.0084**	**70.42**
	0.02	2	0.8846	0.0000	0.0000	0.0000	73.12
	0.03	2	0.8856	0.0000	0.0000	0.0000	72.48
	0.05	2	0.8861	0.0000	0.0000	0.0000	73.09
CNN	-	-	0.8865	0.0500	0.0046	0.0084	4.22

Source: created by the authors

Table 2. Performance Comparison of Models for Head Pose Estimation

Model Architecture	Threshold (Yaw)	Accuracy	Precision	Recall	F1-score	FPS
Dlib	3	0.4745	0.5025	0.6056	0.5492	10.57
	5	0.7030	0.5962	0.6823	0.6364	10.67
	7	0.8535	0.7312	0.7369	0.7340	10.74
	9	0.9005	0.8268	0.7516	0.7874	10.62
	11	**0.9037**	**0.8391**	**0.7456**	**0.7896**	**10.70**
	13	0.8965	0.8407	0.7105	0.7701	10.65
	15	0.8838	0.8335	0.6613	0.7375	10.71
MediaPipe	**3**	**0.8511**	**0.6028**	**0.5463**	**0.5732**	**76.30**
	5	**0.8559**	**0.6389**	**0.5152**	**0.5705**	**75.36**
	7	0.8232	0.5447	0.3972	0.4594	79.45
	9	0.8033	0.2678	0.3333	0.2970	76.86
	11	0.8033	0.2678	0.3333	0.2970	76.91
	13	0.8033	0.2678	0.3333	0.2970	76.14
CNN	-	0.8033	0.2701	0.3333	0.2984	6.54

Source: created by the authors

Table 3. Performance Comparison of Models for Body Twist Detection

Model Architecture	Threshold (Twist)	Threshold (Lean)	Accuracy	Precision	Recall	F1-score	FPS
MediaPipe	0.01	0.05	0.1362	0.2297	0.3447	0.2758	32.72
	0.03	0.05	0.1891	0.2851	0.3108	0.2974	32.06
	0.05	0.05	0.2724	0.3025	0.2978	0.3001	32.79
	0.08	0.05	0.3719	0.3065	0.2892	0.2976	31.93
	0.10	0.05	0.4176	0.3167	0.2955	0.3057	31.52
	0.15	0.05	0.4471	0.3055	0.2747	0.2893	31.93
	0.20	0.05	0.4588	0.2865	0.2497	0.2668	29.98
	0.25	0.05	0.4668	0.2765	0.2364	0.2549	31.14
	0.01	0.10	0.1362	0.2297	0.3447	0.2758	32.36
	0.03	0.10	0.1891	0.2851	0.3108	0.2974	31.85
	0.05	0.10	0.2724	0.3025	0.2978	0.3001	32.49
	0.08	0.15	0.3719	0.3065	0.2892	0.2976	31.62
	0.10	0.15	0.4176	0.3167	0.2955	0.3057	31.85
	0.15	0.15	0.4471	0.3055	0.2747	0.2893	32.29
	0.20	0.25	0.4588	0.2865	0.2497	0.2668	31.48
	0.25	0.25	0.4668	0.2765	0.2364	0.2549	30.87
CNN	-	-	0.4023	0.3158	0.3520	0.3329	7.32

Source: created by the authors