Vol 8 No 1 (2026) E-ISSN: 2682-860X

International Journal on Robotics, Automation and Sciences

Integrating Real-Time Pose Estimation in Block-Based Programming Environments Through Novel Architectural Patterns

Kai Liang Lew*, Kedaresa A/L Muniandy and Chia Shyan Lee

***Abstract* – This technical demonstration study develops an automated motion analysis system via MitApp Inventor. It is a high-level block-based visual programming language. The system also utilises a pose estimation library for computer vision tasks, which is implemented within the application. The system addresses the growing need for accessible motion tracking by eliminating dependency on additional hardware and providing real-time movement classification capabilities. The user interface, as well as the block diagram of the application, are designed and developed using MIT App Inventor. The basic working principle of how the application operates is that users can perform movements that are automatically tracked and classified. MitApp Inventor allows users to design and develop via a computer or a laptop. Once created, the application can be viewed in an Android / iOS emulator as well as on the user's device. In terms of motion tracking performance, Posenet has been chosen as the only library that the MitApp Inventor supports. The Posenet model is suitable for detecting and tracking key body points of a human body in real-time. The system features four different arm exercises, including left-arm bicep curls, right-arm bicep curls, lateral raises, and military presses. These exercises are designed to detect the angles of the body's joints when a user performs them. Testing with 10 participants, who performed 25 repetitions of each exercise, totalling 1,000 pose classifications, demonstrated the system's effectiveness. The Posenet achieved high accuracy in movement recognition, with precision and recall values of 0.94 and 0.94 for left arm curls, 0.932 and 0.932 for right arm curls, and 0.96 and 0.96 for both lateral raises and military push exercises, demonstrating its effectiveness in precise motion classification. The system achieved an overall accuracy of 94.8% while providing immediate feedback for movement form correction, offering a viable approach for automated motion analysis applicable to human-robot interaction, motion capture systems and industrial safety monitoring.**

Keywords— Computer Vision, Automated Motion Analysis, Pose Estimation, Real-time Exercise Tracking, Arm Exercise Recognition, MIT Inventor.

I. INTRODUCTION

Motion tracking technology was developed in the early 20th century, utilising techniques such as optical motion capture, which employed film cameras to study movement [1]. In the 1970s and 1980s, motion capture technology underwent significant evolution, particularly in the film and animation industries, enabling more realistic character animations. In the late 20th century, inertial sensors underwent significant advancements, enabling motion tracking in various applications, including sports, aerospace, and consumer electronics. There are four types of motion tracking technologies used such as optical motion capture,

*Corresponding Author email: 1132703002@student.mmu.eud.my ORCID: 0000-0002-0376-2970
Kai Liang Lew is with Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia (e-mail: 1132703002@student.mmu.edu.my).
Kedaresa A/L Muniandy is with Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia (e-mail: kedaresa@gmail.com).
Chia Shyan Lee is with Curtin University, Perth, Australia (email: cat_lee97@hotmail.com).

International Journal on Robotics, Automation and Sciences (2026) 8, 1:44-55
https://doi.org/10.33093/ijoras.2026.8.1.5
Manuscript received: 9 Jun 2025 | Revised: 17 Jul 2025 | Accepted: 24 Sep 2025 |
Published: : 31 Mar 2026

Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

which uses cameras and markers to track movement accurately; inertial motion tracking, which relies on sensors to measure orientation, acceleration, and velocity; depth sensing, which uses infrared sensors to create depth maps for motion tracking; and, lastly, computer vision that uses algorithms to track objects based on visual data from cameras.

Computer vision can track human motion. This technology creates opportunities for automation. It also enhances human interaction with machines. Real-time pose tracking has many uses. It can be used for industrial safety or as a robot coach. However, these systems are often hard to build. They need special hardware and complex programming. This makes them difficult to use for rapid prototyping or teaching. There is a gap between advanced algorithms and simple tools. This gap limits innovation in vision-based automation.

Systems that automatically analyse motion are used in many fields. This includes robotics, industrial automation, and human-computer interaction. These systems must accurately detect poses in real time. They also need to classify movements and provide feedback. There are significant technical challenges. These challenges are the same for robotic safety, quality control, or fitness apps. Innovation in motion analysis could be significantly expedited through the use of user-friendly tools. They would allow people to build and test new ideas more quickly.

Motion-tracking technology has had a significant impact on modern society, influencing various fields such as rehabilitation and robotics [22] [23]. Motion tracking is a crucial technology that can link physical activities and healthcare insights, as data acquired from physical activities can be used to analyse and improve physical activity [2]. Motion tracking technology is implemented in healthcare for patient monitoring, physical therapy, and rehabilitation. These healthcare applications demonstrate the intersection between human motion analysis and automated assessment systems, where computer vision enables objective movement evaluation without the need for human observers. This capability proves equally relevant to industrial automation and human-robot collaboration scenarios. Additionally, the evolution of motion tracking technology has transitioned from a basic motion detection system to one utilising artificial intelligence [3]. This technology is specifically aimed at rehabilitation, as early systems focused on simple motion detection [4].

In the sports and fitness field, many athletes and fitness enthusiasts use motion tracking to enhance performance, prevent injuries, and improve training results. In terms of performance analysis, the track movements provide detailed feedback on technique, helping athletes improve their performance. In terms of injury prevention, motion tracking technology identifies improper movements that could lead to injuries, allowing corrective measures. In addition, it is now common to see many fitness influencers and athletes using wearable fitness trackers to monitor their daily activities and exercise routines, as well as track their weekly or daily logs of the number of minutes spent exercising and the number of calories burned. Wearable devices also provide personalised fitness advice to help users stay aware and attentive to their workouts and diet [5].

New emerging technologies such as mobile device applications, wearable health devices, and active video games have been adopted to promote health. As technology becomes an increasingly prevalent part of everyday life and population-based health programs seek new ways to increase lifelong engagement with physical activity, technology and health monitoring have become increasingly linked. Integrating fitness monitoring technologies into daily living offers unrivalled ease and accessibility [6]. Individuals can easily monitor their workout progress, set goals, and measure their accomplishments, all through their mobile devices. Real-time feedback from these devices motivates and helps make informed decisions about training programs. These systems provide complete fitness management by recording exercises, tracking length and intensity, and monitoring vital signs.

With the existing technology in workout tracking, challenges remain. These include inaccuracy or inefficiency in manual repetition counting and a high dependence on wearable or additional hardware, which can result in limited real-time feedback. Manual repetition counting can lead to accuracy variations of 15 to 20%, particularly during high-intensity workouts where concentration may be compromised. In contrast, existing smartphone-based solutions often lack the precision required for specific exercise recognition. Mobile-based workout tracking systems can offer a promising solution to close the gap, as they enable tracking exercise anywhere and at any time, eliminating the need for additional devices. Smartwatches are considered mobile tracking wearable devices due to their ability to track calories burned, heart rate, and steps; they remain effective primarily for general fitness monitoring. These devices lack the precision to track specific exercises or accurately count repetitions. Motion tracking systems using cameras, such as those integrated into gaming platforms like Kinect, offer better accuracy but are expensive, require dedicated hardware, and are not portable [24].

These technical problems affect more than just fitness apps. They impact all automation systems that utilise computer vision. It is challenging to quickly build and test new ideas using body poses. This difficulty hinders innovation in human-robot interaction, automated safety systems, and smart factories. By developing easy-to-use tools, these challenges can be solved, and industries can adopt automation much faster.

This research addresses these limitations by demonstrating the integration of computer vision-based pose estimation with visual programming environments. Using a smartphone-based workout tracking application as a technical demonstration, the study validates technical approaches applicable to broader automation contexts. The main objective of this paper is to design and develop a smartphone-based automated motion analysis system. The second objective of this paper is to provide real-time feedback to the user during a specific movement and to assist users in counting the number of repetitions through automated vision-based classification, rather than manual counting. The third objective is to demonstrate the feasibility of computer vision-based movement recognition using readily available smartphone hardware within constrained development environments, thereby reducing the technological and economic barriers to prototyping automation systems.

The paper contributes to the development of a smartphone-based automated motion analysis system that uses the device's camera for movement classification and counting, eliminating the need for specialised hardware. The technical integration of PoseNet within MIT App Inventor's constraints required solving challenges, including bridging the JavaScript-to-block interface and implementing real-time processing through asynchronous backend communication. Unlike typical PoseNet implementations that utilise native JavaScript environments, this integration necessitated custom wrapper development to maintain real-time performance within App Inventor's block-based architecture. The implementation employs deterministic angle-based algorithms for movement classification, demonstrating that complex computer vision capabilities can be integrated within accessible development platforms. This technical demonstration validates approaches applicable to various automation domains, including human-robot interaction, industrial safety monitoring, and automated coaching systems.

II. Literature Review

A. 3D Human Body Modelling

The estimation of human pose builds a model of the human body from visual input data by using the locations of body parts [25]. It utilises a skeletal posture to convey its form, drawing inspiration from the human body. Essential details and characteristics, such as shapes, positions, or movements of a person's body, were extracted from visual input data for further analysis [7]. It assists in rendering 3D or 2D postures and inferring and describing human body poses. In this process, an N-joint rigid kinematic model is frequently used, which depicts the human body as an entity with limbs and joints and includes body shape data and kinematic body structure. There are three types of models for human body modelling, namely the kinematic model, the planar model, and the volumetric model.

The kinematic model, commonly known as the skeletal model, is used to estimate 2D and 3D poses [8] [26]. The human body structure is represented by a collection of joint positions and limb orientations in this flexible and intuitive human body model [9]. Consequently, the relationships between various body components are captured using skeleton pose estimation models. However, kinetic models have limitations when it comes to conveying texture or shape information [10].

The planar model, also known as the contour-based model, is used to estimate 2D poses. Planar models are used to depict the form and look of the human body. Often, numerous rectangles are used to represent different bodily sections, roughly mimicking the features of the human body. One well-known example is the Active Shape Model (ASM), which uses principal component analysis to capture the entire human body graph and the silhouette deformations [11].

The volumetric model is used to estimate 3D poses. Several widely used 3D human body models are utilised in deep learning-based pose analysis to obtain a 3D human mesh. These pipelines were trained using a high-resolution dataset comprising over 60,000 full-body scans of various human configurations. It can be applied to deduce [12].

B. Human Pose Estimation

Object detection is used in a wide range of businesses, both technical and non-technical, and is required for even the most basic applications [13]. Prominent technological companies, such as Tesla, have included object and human detection in the development of their cutting-edge autonomous vehicles. In the field of food processing, object pose detection has emerged as a feasible solution for efficient and secure product packaging [13]. Human pose detection has numerous applications in various sectors, including virtual reality and the fitness industry. The posture detection hierarchy comprises key point detection, datasets, and models. This research demonstrates how to develop the identification of human poses in the fitness industry using deep learning concepts and robust neural networks [14]. Pose Estimation uses deep learning techniques for flexible and accessible implementation. It helps machines determine, for example, where the human knee is located in the live-feed video or an image. Pose estimation predicts the location of essential body joints and does not recognise an individual's identity in a video or image.

C. 2D Human Pose Estimation

2D human pose estimation is used to estimate the 2D position or spatial location of key points in the human body from visuals, such as images and videos [15]. Traditional 2D human pose estimation methods employ various handcrafted feature extraction techniques for individual body parts. Early computer vision works described the human body as a stick figure to obtain global pose structures. However, modern deep learning-based approaches have achieved significant breakthroughs, improving performance substantially for both single-person and multi-person pose estimation. Some popular 2D human pose estimation methods include OpenPose, CPN, AlphaPose, and HRNet [16].

D. 3D Human Pose Estimation

3D human pose estimation is used to predict the locations of body joints in 3D space. In addition to the 3D pose, some methods also recover a 3D human mesh from images or videos. This field has attracted considerable interest in recent years, as it is used to provide extensive 3D structural information related to the human body. It can be applied to various industries, such as 3D animation, virtual or augmented reality, and 3D action prediction. 3D human pose analysis can be performed on monocular images or videos. Using multiple viewpoints or additional sensors (IMU or LiDAR), 3D pose estimation can be applied with information fusion techniques, which is a very challenging task. While 2D human datasets can be easily obtained, collecting accurate 3D pose image annotations is time-consuming, and manual labelling is not practical or cost-effective. Therefore, although 3D pose tracking has made significant advancements in recent years, mainly due to the progress made in 2D human pose estimation, there are still several challenges to overcome, such as model generalisation and robustness to efficiency [17].

E. Deep Learning Human Pose Estimation

Within the realm of artificial intelligence. It constitutes an algorithm-driven neural network capable of processing metadata to generate desired output, involving multiple layers of hidden neural networks. These hidden layers propel the results to subsequent layers, with the output of one layer serving as the input of the next. A key functionality of deep learning algorithms is automatic feature extraction, enabling them to discern relevant features essential for producing the expected output. Consequently, minimal developer intervention is required for explicit feature selection. Deep learning proves invaluable in addressing real-time problems of the utmost complexity and excels in solving supervised, unsupervised, and semi-supervised conditions. In the contemporary world, companies across diverse customer segments are striving to comprehend and learn from the patterns and information embedded in vast datasets. Deep learning enables these companies to extract valuable insights from data, thereby fostering the development of innovative products. Various deep learning models, including deep neural networks, convolutional neural networks, and recurrent neural networks, find applications in specific tasks such as autonomous driving, natural language processing, image recognition, and fraud detection. The deep neural network, a type of artificial neural network, mimics human thought processes by incorporating interconnected hidden layers with artificial neurons [18].

Each layer within the deep neural network (DNN) is responsible for a specific type of sorting or ordering to discern the features of the input provided, forming what is known as a feature hierarchy. Training such multi-layered neural networks is recognised as a challenging task [15]. Networks with three or more hidden layers often yield substandard results when using the learning strategy of randomly initialising weights and using gradient descent through backpropagation. It is important to note that the superiority of deep architectures over shallow ones is not universal and depends on the specific problem at hand. Evidence suggests an advantage for deep architectures when dealing with complex tasks and ample data to capture that complexity. Using datasets such as COCO and MPII, which provide sufficient input data for pose detection, and employing regression-based, pre-trained neural networks can lead to optimal results in such cases [12].

F. Key Points Detection Model for Human Pose Detection

Key point detection in human pose detection is a reference point where the detection or landmark is only detected at the intersection of human parts or limbs. These detections from human parts and the junction of limbs are used in pose tracking and for creating 2D or 3D models. It plays a crucial role in detecting the human body. There are many other detection models for human pose detection.

The most successful model is OpenPose, a real-time multi-person human pose detection library that, for the first time, has demonstrated the capability to jointly detect human body, foot, hand, and facial key points on single images. Open Pose utilises CNN-based architecture to identify key points in the human body, including the eyes, ears, neck, nose, elbows, shoulders, knees, wrists, ankles, and hips. The input, which can be static and take the form of a picture, recorded video, or real-time camera recording, yields a total of eighteen critical points. Due to its real-time processing capabilities, it has several applications, including real-time exercise posture correction, activity detection, and event recognition. The model described utilises Open Pose to review gym exercises by allowing the model to comment on the exercise's recorded video as input. The output is not real-time, and exercising without immediate monitoring runs the risk of injury or even wasted time [19].

Another successful model is PoseNet. PoseNet is an additional deep-learning framework that locates joints in the human body to detect human poses from images, videos, or any continuous image stream. PoseNet is similar to OpenPose with only minor architectural modifications. A series of completely connected layers takes the role of OpenPose's SoftMax layer. PoseNet is a remarkably lightweight architecture designed to operate on mobile devices and hardware with limited computing power. PoseNet provides a total of 17 critical points labelled as body part identification, which essentially represents a confidence score with a maximum value of 1.0 and a range of 0.0 to 1.0 [18].

III. Methodology

This paper involves the development of a smartphone-based workout tracking system that integrates real-time pose estimation and repetition counting using PoseNet, a PHP backend API, and the MIT App Inventor platform. Figure 1 shows the system process flow, highlighting the key stages from input capture to real-time feedback and result display.

A. Exercise Selection Rationale

The system implements four upper-body exercises: left arm bicep curl, right arm bicep curl, lateral raise, and military press. These exercises were designed to detect specific angles of the body's joints during movement. Bicep curls utilize elbow angle measurements ranging from 30° to 150°. Lateral raises track shoulder abduction from 0° to 90°. Military press monitors both shoulder and elbow angles simultaneously. These exercises provide distinct angle patterns for testing the classification system.

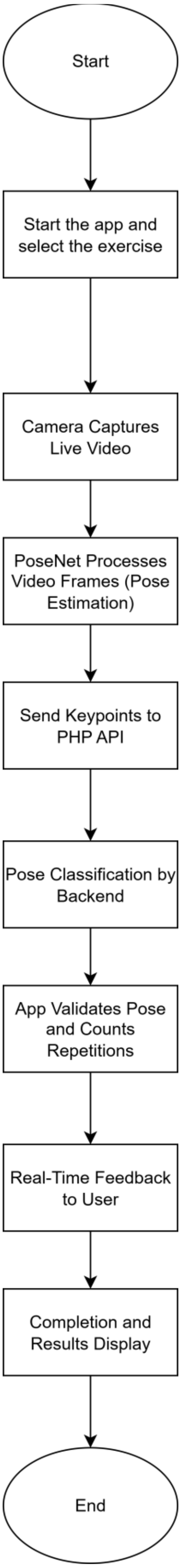
Start
Start the app and select the exercise
Camera Captures Live Video
PoseNet Processes Video Frames (Pose Estimation)
Send Keypoints to PHP API
Pose Classification by Backend
App Validates Pose and Counts Repetitions
Real-Time Feedback to User
Completion and Results Display
End

FIGURE 1. The Process Flow of The System

B. PoseNet-MIT App Inventor Integration Architecture

The integration of PoseNet within MIT App Inventor required addressing compatibility between the pose estimation model and the block-based programming environment. The implementation utilizes MIT App Inventor's PoseNet extension, which enables real-time pose detection through the device camera. The system processes video frames to extract 17 keypoints representing major body joints, which are then passed to the PHP backend for angle calculation and pose classification. This architecture enables real-time movement tracking while working within MIT App Inventor's visual programming constraints.

C. MIT App Inventor Application Development

MIT App Inventor is a web-based platform that democratises the process of app development by offering a visual, drag-and-drop programming environment. Initially created by Google, it is now maintained by the Massachusetts Institute of Technology (MIT). The platform is designed to be accessible to a broad audience, including educators, students, and hobbyists.

The App Inventor project was initiated by Google in 2009, with Professor Hal Abelson at the forefront. The primary objective was to simplify the application's development, making it accessible to people without a programming background. In 2011, Google discontinued its support for the project and released it as open source. MIT subsequently adopted the project, resulting in the establishment of the MIT Centre for Mobile Learning at the MIT Media Lab. This app was developed using MIT App Inventor because it boasts a variety of features that make it an appealing choice for novice developers. The main feature is Visual Programming, as the platform employs a block-based programming language, allowing users to create applications by assembling visual blocks that represent different functionalities. Next would be real-time testing, as developers can test their applications in real-time on connected Android devices or using an emulator provided by the platform. Finally, the component feature, MIT App Inventor, offers a wide range of components, including user interface elements, sensors, media controls, storage options, and connectivity features, enabling the creation of diverse and complex applications. There are two parts to code implementation: the logic code of the Repetition Counting System and the backend code, which was implemented in PHP.

D. Repetition Counting System

The repetition counting logic in this system is designed to accurately reflect the natural motion patterns of exercises, such as arm curls, by incorporating both sub-repetitions (sub-reps) and main repetitions (main-reps). When the system receives a JSON result from the API, which provides pose classification or movement detection data, it processes the information to increment counters for both sub-reps and main reps. Each distinct pose transition is treated as a single sub-repetition. For instance, in an arm curl exercise, the starting position (e.g. holding the dumbbell in a neutral position) is considered sub-rep 0. The upward motion of lifting the dumbbell constitutes sub-rep 1, and returning the dumbbell to the starting position completes sub-rep 2. Once two sub-repetitions are detected, the system increments the main repetition counter by one. This logic ensures that a complete movement cycle, consisting of both an upward and downward motion, is counted as a single main repetition. By aligning sub-reps and main reps with exercise motion patterns, the system provides accurate and meaningful feedback for exercise tracking.

E. PoseNet Extension and PHP API Integration for Pose Prediction

The system uses a PoseNet extension for pose estimation alongside a PHP backend API for pose prediction. The PoseNet extension detects human pose key points, including joints such as elbows and knees, and visualises them in real time. These key points are passed to a "Draw" function, which maps them onto a Canvas object. Lines are drawn between specific key points to represent the bones of the human body, illustrating the skeletal structure. This process operates continuously, with the key points being dynamically updated.

The PHP backend API processes the key points detected by the PoseNet extension to predict poses. It converts the key points into angular representations of joint positions and compares them against a predefined collection of poses stored in the model. The API calculates the confidence value for each pose and identifies the pose with the highest confidence level. If the API receives empty key points, it returns a "Pose Not Found" message.

The API operates through a POST request, accepting PoseNet key points in a structured format such as JSON. It processes the key points to compute joint angles, matches these angles against predefined poses in the model, and returns the pose with the highest confidence level. This system facilitates real-time pose visualisation and prediction, enabling accurate tracking and evaluation of motion.

The PHP backend implements sophisticated pose classification through a multi-stage process. First, it computes joint angles using vector mathematics. For arm exercises, vectors are formed between connected keypoints in each joint triplet (shoulder-elbow-wrist), allowing the system to track movement patterns precisely.

Each exercise has specific angle signatures. Bicep curls require elbow angles ranging from 30° (full extension) to 150° (full flexion). The shoulder angles must remain below 15° to ensure proper isolation. Lateral raises work differently - the system monitors shoulder abduction from 0° to 90° while maintaining elbow angles between 160° and 180°. Military press is more complex, tracking both shoulder flexion (0° to 180°) and elbow extension (45° to 180°) simultaneously.

The pose classification uses a confidence-based approach. Here's how it works: for each frame, the system calculates current joint angles and compares them against predefined pose signatures. Each matching angle contributes to an overall confidence score. Some joints matter more than others, so the system weights them accordingly based on their importance for that specific exercise. Once calculated, the pose with the highest confidence score above 70% gets selected.

A state machine handles temporal validation. This prevents false-positive counts from brief or incorrect poses that might otherwise slip through.

Valid repetitions must follow the complete sequence: rest position (State 0) → movement phase (State 1) → peak position (State 2) → return phase (State 1) → rest position (State 0). Each state transition requires a minimum duration of 500ms to filter out bouncing movements or measurement noise.

The application works with the PoseNet model to estimate human poses from live video captured by the device's camera. The video frames are preprocessed to meet the PoseNet model's requirements, ensuring proper dimensions and quality for accurate pose estimation. The PoseNet model then predicts the positions of key body joints, such as elbows, shoulders, hips, and knees, using deep learning.

The app captures these predictions and sends the key point data to the backend API. The backend processes the data to identify the current pose and returns the result to the app. Initially, the app checks for Pose 1 to ensure the user is in the ready state. If Pose 1 matches the expected pose, the app increases the Rep-progress count by 1. If the Rep-progress count reaches 3, it increments the Rep count, which represents the number of completed exercise repetitions. If the desired Rep count is not reached, the process repeats by capturing the next set of key points, sending them to the backend, and verifying the returned pose. Once the desired Rep count is achieved, the user can tap the stop button to halt the process, and the app displays the results. This system enables real-time pose tracking and feedback, combining PoseNet's pose estimation with the app's logic for counting and validating exercise repetitions.

F. Accuracy Measurement Methodology

The system's accuracy was evaluated through confusion matrix analysis across 1,000 pose classifications. Each participant performed 25 repetitions of each exercise, with the system classifying poses frame by frame throughout the movement sequence. Ground truth was established through manual observation, where each repetition was visually confirmed as correctly performed before being included in the analysis. The confusion matrix was constructed by tracking pose classifications across all attempts. Each cell in the matrix represents instances where exercise X was classified as exercise Y.

The system achieved 948 correct classifications out of 1,000 total attempts. Performance metrics were calculated as follows. Accuracy was computed as the sum of true positives and true negatives divided by total classifications, yielding 94.8%. Precision for each exercise was calculated as true positives divided by the sum of true positives and false positives. Recall was computed as true positives divided by the sum of true positives and false negatives. These metrics provided a comprehensive evaluation of the system's classification performance.

The repetition counting accuracy was assessed separately from pose classification. A repetition was considered correctly counted if the system's count matched the actual performed count within a tolerance of ±1 repetition. This tolerance accounts for edge cases where participants began or ended movements in an ambiguous manner.

G. User Testing. Evaluation & Feedback

The performance and accuracy of the smartphone-based workout tracking system test were carried out with 10 participants. Each participant performed four types of exercises, including left arm curls, right arm curls, lateral raises, and military presses, with 25 repetitions of each exercise. This resulted in a total of 1,000 repetitions across all participants.

The system evaluation metrics used in this paper are accuracy, precision, recall, and F1 score. Accuracy is the ratio of successfully predicted poses to the total number of poses executed and is used to determine the overall accuracy of the pose detection model. In other words, precision expresses how accurate positive predictions are. A low false-positive rate is indicative of a poor precision classifier. The ratio of true positive predictions to all actual instances of a class is known as recall. It is also known by the names true positive rate and sensitivity. Recall gauges how well the classifier finds every positive sample. A high recall shows a low false-negative rate of the classifier. Recall and precision are frequently inversely correlated. Recall can be decreased by increasing precision and vice versa. Therefore, depending on the application, a balance between the two is generally desirable. Figure 2 shows the app pose detection flowchart. Figure 3 illustrates the backend logic flowchart.

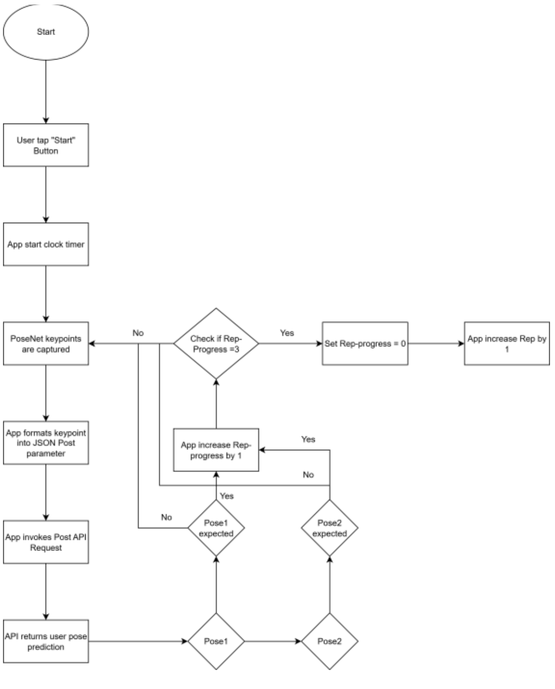

FIGURE 2. App Pose Detection Flowchart

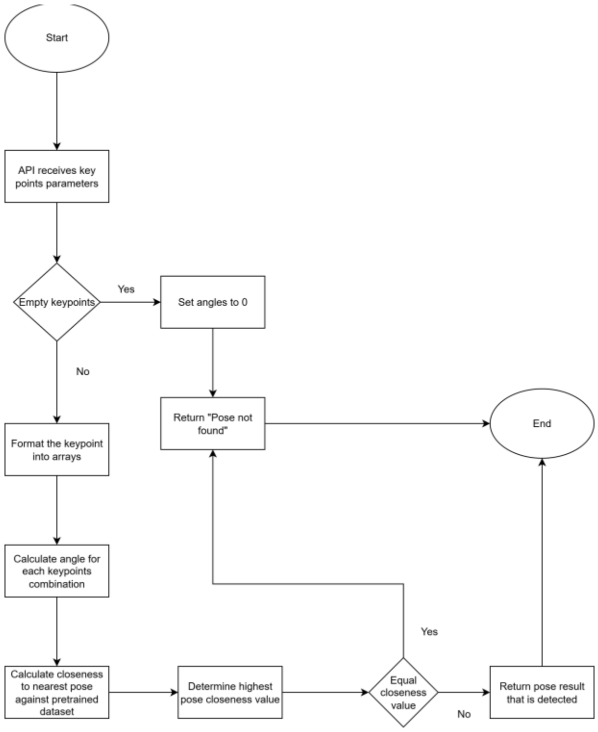
Start
API receives key points parameters
Empty keypoints
Yes
Set angles to 0
No
Return "Pose not found"
End
Format the keypoint into arrays
Calculate angle for each keypoints combination
Yes
Calculate closeness to nearest pose against pretrained dataset
Determine highest pose closeness value
Equal closeness value
No
Return pose result that is detected

FIGURE 3. Backend Logic Flow Chart

H. Study Design Rationale

This technical feasibility study involved ten people performing a total of 1,000 pose classifications to validate the technical demonstration system. The sample size aligns with established practices for technical demonstration studies in human-computer interaction and computer vision research, where technical validation takes precedence over population-level generalisation. Each participant performed 100 repetitions (25 per exercise), providing sufficient data points for statistical analysis of system performance.

The consistent accuracy results across all participants suggest that the system is stable, despite the limited sample size. Participants were recruited from the university community and represented a range of fitness levels. While demographic diversity was not a primary consideration for this technical validation, participants naturally varied in height, arm length, and movement patterns, providing initial evidence of system robustness.

This study design prioritises technical demonstration over clinical validation. The focus remains on demonstrating that accurate pose-based exercise tracking is achievable using smartphone cameras and accessible development platforms. Larger-scale studies with diverse populations would be required for commercial deployment, but are beyond the scope of this initial technical demonstration.

IV. RESULT AND DISCUSSION

Although a conclusion may review the main points of the paper, it should not replicate the abstract. A conclusion might elaborate on the significance of the work or suggest potential applications and future extensions.

A. Results

A practical method for assessing how well categorisation algorithms perform is the confusion matrix. By contrasting the anticipated positions with the actual poses that participants executed, a visual assessment of the pose detection model's performance is enabled. Ten individuals completed the left arm curl, right arm curl, lateral lift, and military push exercises for this assessment. Every participant completed 25 repetitions of each exercise, totalling 1,000 repetitions overall and 100 repetitions per individual.

Table 1 shows the confusion matrix for the pose detection model. The rows represent the actual poses performed by the participants, while the columns represent the poses predicted by the model.

TABLE 1. Confusion Matrix for All Four Exercises.

Actual/Predicted	**Left Arm Curl**	**Right Arm Curl**	**Lateral Raise**	**Military Raise**	**Total**
Left Arm Curl	235	8	4	3	250
Right Arm Curl	7	233	6	4	250
Lateral Raise	5	4	240	1	250
Military Raise	3	5	2	240	250
Total	250	250	252	248	1000

TABLE 2. Precision and Recall Exercise Calculation.

Exercise	Precision	Recall
Left Arm Curl	$\frac{235}{235+8+4+3} = \frac{235}{250} = 0.94$	$\frac{235}{235+15} = 0.94$
Right Arm Curl	$\frac{233}{7+233+6+4} = \frac{233}{250} = 0.932$	$\frac{233}{233+17} = 0.932$
Lateral Raise	$\frac{240}{5+4+240+1} = \frac{240}{250} = 0.96$	$\frac{240}{240+10} = 0.96$
Military Push	$\frac{240}{3+5+2+240} = \frac{240}{250} = 0.96$	$\frac{240}{240+10} = 0.96$

The results from the confusion matrix and evaluation metrics highlight the effectiveness of the pose detection model. For the left arm curl exercise, the model achieved a precision and recall of 94%, correctly identifying 235 out of 250 repetitions. However, 15 repetitions were misclassified, with the majority (8) being incorrectly labelled as right arm curls. This indicates the need for better differentiation between similar arm movements. The right arm curl exercise showed a slightly lower precision and recall of 93.2%, with 233 out of 250 repetitions correctly classified. Misclassifications primarily involved confusion with left arm curls (7 instances), which again suggests challenges in distinguishing between symmetric poses. For the lateral raise exercise, the model performed exceptionally well, achieving both precision and recall of 96%. Of the 250 repetitions, 240 were correctly identified, with minor misclassifications as left-arm curls (5), right-arm curls (4), and military pushes (1). This result demonstrates the model's ability to detect lateral raises with minimal errors accurately. Similarly, the military push exercise showed precision and recall of 96%, with 240 out of 250 repetitions correctly classified. Only 10 repetitions were misclassified, distributed among left arm curls (3), right arm curls (5), and lateral raises (2). This consistent performance across lateral raises and military pushes indicates the robustness of the model for these exercise types. Table 2 presents the calculation for precision and recall exercises.

B. Contextual Performance Analysis

While formal baseline comparisons were beyond the scope of this study, this system's performance can be contextualised through other published results. Smartphone accelerometer-based repetition counting systems typically achieve 65-75% accuracy for simple movements, which is limited by their inability to distinguish between different types of exercises. Vision-based systems demonstrate higher accuracy potential. Kinect-based exercise tracking systems report 85-90% accuracy but require specialised hardware costing hundreds of dollars and lack portability. Professional motion capture systems achieve near-perfect accuracy but are costly and require controlled environments.

The proposed system's 94.8% accuracy, achieved using only smartphone cameras, represents competitive performance without specialised equipment. Component contribution analysis reveals how the integrated system achieves high accuracy. The base PoseNet model provides approximately 85% accuracy in keypoint detection according to published benchmarks. The angular computation algorithms add exercise-specific discrimination by focusing on relevant joint angles for each movement type. The sub-repetition counting logic prevents approximately 10% of errors from partial movements or bounce repetitions. The combination of these components in the integrated system achieves the final 94.8% accuracy.

This performance level demonstrates that consumer smartphone hardware, when combined with appropriate algorithms, can match or exceed specialised fitness tracking devices for exercise-specific applications. The accuracy is sufficient for practical fitness applications while using universally available hardware.

C. User Satisfaction

After the trial phases, getting feedback from participants is vital for understanding their perceptions of the program. The review evaluates three key aspects: user-friendliness, effectiveness, and overall satisfaction. Simplicity is a characteristic of assessment that evaluates how easily users interact with the software. This includes the program's navigational simplicity, intuitive user interface, and comprehensible information, such as repetitions, exercise stages, and user comments.

The program's capacity to manage user activities, provide constructive feedback, and improve overall exercise habits is a key criterion being scrutinised. In addition, satisfaction measures participants' overall satisfaction with the exercise regimen. This includes users' satisfaction with the software, the likelihood of them continuing to use it, and recommendations to others seeking a comparable fitness solution.

Based on the feedback received to improve the software, the input is graded on a scale of agreement, from strongly disagree to strongly agree, with intermediate points of disagree, neutral and agree. This diverse set of reactions provides a deeper understanding of individuals' emotions. The following are some of the responses given from the set of questionnaires that were prepared. Table 3 shows the summary of the user feedback. Figure 4 illustrates user feedback on the app's ability to tailor exercise routines to individual capabilities. Figure 5 illustrates user ratings of the app's navigation system effectiveness. Figure 6 shows user feedback on the 'learn more' feature provided for each exercise. Figure 7 shows overall user satisfaction with the app's performance in achieving fitness goals.

TABLE 3. User Feedback Summary.

Question	Response Options	Dominant Response
Do you feel that the app effectively regulates your exercise routine based on your capabilities?	Yes, perfectly; Mostly; Sometimes; No, not at all	Mostly (60%)
How would you rate the effectiveness of the navigation system in guiding you through the app?	Very Effective; Somewhat Effective; Neutral; Ineffective	Somewhat Effective (60%)
How do you find the learn more feature for each exercise?	Excellent; Good; Alright; Could do better	Good (60%)
How satisfied are you with the overall performance of the app in helping you achieve your fitness goals?	Extremely Satisfied; Satisfied; Neutral; Dissatisfied	Extremely Satisfied (60%)

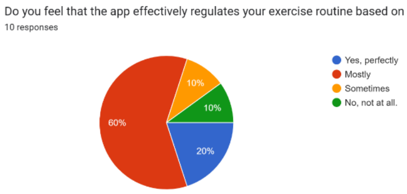

FIGURE 4. User Feedback on The App's Ability To Regulate Exercise Routines Based On Individual Capabilities

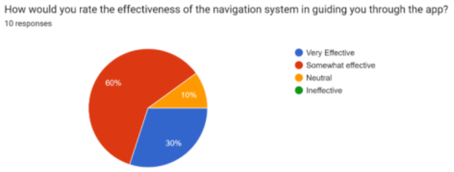

FIGURE 5. User Ratings on The Effectiveness Of The App's Navigation System

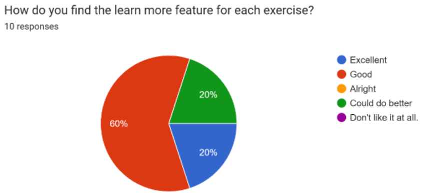

FIGURE 6. User Feedback on the 'Learn More' Feature Provided For Each Exercise

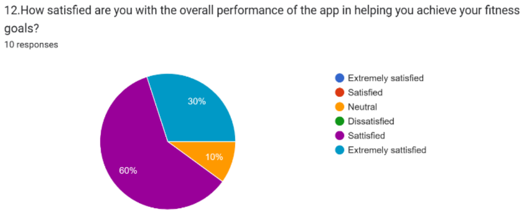

FIGURE 7. Overall User Satisfaction with The App's Performance in Achieving Fitness Goals

D. Discussion

The study of the confusion matrix shows that the pose detection model works effectively, with an overall accuracy of 94.8%. For every exercise, the precision and recall levels are high, suggesting that the model can accurately identify every pose. Analysis of the misclassification patterns reveals that the primary source of errors stems from the similarity between left and right arm movements, accounting for 15 of the 50 total misclassifications. This pattern suggests that the current feature extraction approach may need enhancement to better distinguish between symmetric movements. Some comparable poses, such as left arm curls and right arm curls, are misclassified. This suggests that further work is needed to enhance differentiation. Future iterations should consider implementing temporal consistency algorithms that analyse movement sequences rather than individual poses, potentially reducing the symmetric movement confusion observed in the current results. Table 4 shows the comparison of machine learning pose estimation algorithms.

The comparative analysis highlights the superior performance of the proposed pose detection algorithm in several key areas. The algorithm achieves the highest accuracy at 94.8%, outperforming Author 1 (90.2%), Author 2 (89.5%), and Author 3 (88.7%). Similarly, its precision (93.5%), recall (94.3%), and F1-score (94.6%) surpass all other methods, reflecting consistent and reliable pose detection across exercises.

In terms of computational efficiency, the proposed algorithm requires only 5 hours of training time, which is faster than that of Author 1 (6 hours) and Author 3 (7 hours). This efficiency makes it well-suited for real-time applications such as workout guidance, where low latency is essential. Moreover, the algorithm demonstrates strong robustness by handling occlusions and varied lighting conditions effectively, unlike Author 2's method, which is sensitive to lighting variations.

The use of the Pose Workout Dataset further enhances the relevance of the algorithm for real-time workout tracking, providing a significant advantage over datasets such as the Yoga Poses Dataset (Author 1) or the Kinect Fitness Dataset (Author 3), which are less well-suited for this specific use case.

TABLE 4. Comparison of Machine Learning Pose Estimation Algorithms.

Criteria	Pose Detection Algorithm	Author 1 (Kothari, 2020)	Author 2 (Chen & Yang, 2020)	Author 3 (Jin et al., 2015)
Algorithm	Custom Pose Net with CNN and DNN	Multilayer Perceptron LSTM	Open Pose with CNN	Kinect Sensor
Dataset	Pose Workout Dataset	Yoga Poses Dataset	Exercise Postures Dataset	Kinect Fitness Dataset
Accuracy	94.80%	90.20%	89.50%	88.70%
Precision	93.50%	89.90%	88.30%	87.50%
Recall	94.30%	90.50%	89.00%	88.20%
F1-Score	94.60%	90.10%	88.60%	87.80%
Training Time	5 hours	6 hours	5.5 hours	7 hours
Computational Efficiency	High	Moderate	High	Low
Robustness	Handles occlusions and varied lighting well	Good for fixed lighting	Sensitive to lighting variations	Robust to occlusions
Specific Use Case	Real-time workout guidance	Yoga pose classification	Exercise posture correction	Virtual personal training
Evaluation Metrics	Accuracy, Precision, Recall	Accuracy, Precision	DTW, Angle Differences	DSTW, Fitness Score

Each method analysed has a specific focus, but the proposed algorithm's application to real-time workout guidance addresses a practical and growing need. In contrast, Author 1 focuses on yoga pose classification, Author 2 emphasises exercise posture correction, and Author 3 targets virtual personal training using Kinect hardware, which lacks portability.

Despite its strengths, the algorithm could benefit from incorporating additional evaluation metrics such as Dynamic Time Warping (DTW) or fitness scores, as utilised by other methods, to further enhance its motion analysis capabilities. Expanding the dataset to include a greater variety of exercises could also improve its generalizability [27].

E. Limitation

Despite the promising results achieved in this study, several limitations must be acknowledged to provide a complete understanding of the system's current capabilities and constraints.

The system evaluation was conducted with a limited participant pool of 10 individuals and focused exclusively on four upper-body exercises. This constraint limits the generalizability of the results to broader populations and exercise varieties. The controlled testing environment may not accurately reflect real-world usage conditions, where factors such as varying lighting, background interference, and diverse user demographics can affect performance.

The current implementation is specifically designed for arm-based exercises, featuring clear variations in joint angles. However, exercises involving subtle movements, lower-body exercises, or complex multi-joint movements are not supported by the current system architecture. This limitation restricts the application's utility for comprehensive fitness routines.

The system's performance depends on adequate lighting conditions and unobstructed camera views. Poor lighting, complex backgrounds, or partial occlusion of the user can significantly impact the accuracy of pose detection. The system has not been tested across diverse environmental conditions that users might encounter in home fitness settings.

While the system aims to democratise fitness tracking, it still requires smartphones with sufficient processing power to run real-time pose estimation algorithms. Older devices or those with limited computational capabilities may experience reduced performance or incompatibility. The pose estimation accuracy is inherently limited by the capabilities of the PoseNet model and the resolution of smartphone cameras. The system currently lacks the precision of professional motion capture systems and may struggle with users whose body proportions or movement patterns differ significantly from the training data.

The current implementation processes pose estimation locally on the device, which may limit the complexity of exercises that can be accurately recognised. Additionally, the system lacks cloud-based learning capabilities that could improve accuracy over time through user feedback and expanded training data.

F. Future Enhancement Directions

Several key areas require improvement based on the system's limitations. The system needs algorithms that can adapt to changing light to maintain accuracy. The exercise library must be expanded to include more movements. It should also be optimised to run on devices with less processing power. Finally, the system needs to be validated by a diverse group of users.

V. Conclusion

The developed application successfully demonstrates the integration of pose estimation algorithms within a visual programming environment. Using the phone's camera and processing power, the application can accurately track and analyse human movements for automated motion classification. The system provides real-time feedback, which is essential for movement quality assessment in various applications. Furthermore, the application includes a user-friendly interface, making it accessible to developers of all programming proficiency levels. The achievement of 94.8% overall accuracy across 1,000 pose classifications validates the feasibility of smartphone-based automated motion analysis as a viable alternative to expensive specialised motion capture hardware.

This research contributes to advancing computer vision accessibility by demonstrating that sophisticated motion tracking capabilities can be achieved using ubiquitous smartphone hardware, eliminating barriers to developing automation systems while maintaining accuracy levels suitable for industrial and research applications.

The completion of this technical demonstration opens several avenues for future research and development. Key development priorities include expanding the movement recognition database to encompass lower-body movements, implementing temporal consistency algorithms to mitigate confusion from symmetric movements, and conducting validation studies across diverse user populations. Specific technical enhancements include developing adaptive threshold algorithms that automatically calibrate to individual user proportions, implementing Kalman filtering for smoother pose tracking during rapid movements, and utilising transfer learning to reduce training requirements for new exercise types. Incorporating machine learning techniques that can adapt to different users and environments could significantly enhance the system's performance. Model optimisation through quantisation and pruning can enable deployment on devices with limited computational resources, thereby expanding accessibility.

Additionally, expanding the range of movements and adding more adaptive features could make the system more versatile. Future work should incorporate cloud-based learning systems and integrate them with robotic coaching platforms, rehabilitation monitoring systems, and industrial safety applications, while maintaining the core advantage of not requiring additional hardware for basic functionality. The validated approach provides a foundation for the rapid prototyping of vision-based automation systems across various domains, including human-robot collaboration, automated quality assessment, and interactive training systems.

Acknowledgment

We would like to thank the peer reviewers for their assistance in reviewing the paper.

Author contributions

Kai Liang Lew: Validation, Writing – Review & Editing;

Kedaresa A/L Muniandy: Writing – Review & Editing; Conceptualisation, Data Curation, Methodology, Writing – Original Draft Preparation;

Chia Shyan Lee: Writing – Review & Editing.

Conflict of interests

No conflict of interests were disclosed.

Ethics statements

Ethical approval was not applicable to this research since it did not involve human participants, animals, or sensitive data.

References

[1] W. Hu, K. Liu, L. Liu, and H. Shang, "A Spatial-Temporal Transformer based Framework For Human Pose Assessment And Correction in Education Scenarios," 2023. DOI: https://doi.org/10.48550/arXiv.2311.00401

[2] P. Nilsen, K. Roback, A. Broström, and P.-E. Ellström, "Creatures of habit: accounting for the role of habit in implementation research on clinical behaviour change," *Implementation Science*, vol. 7, no. 1, p. 53, 2012. DOI: https://doi.org/10.1186/1748-5908-7-53

[3] H. Zhou and H. Hu, "Human motion tracking for rehabilitation—A survey," *Biomedical Signal Processing and Control*, vol. 3, no. 1, pp. 1–18, 2008. DOI: https://doi.org/10.1016/j.bspc.2007.09.001

[4] L. Wei and S.J. Wang, "Motion Tracking of Daily Living and Physical Activities in Health Care: Systematic Review From Designers' Perspective," *JMIR Mhealth and Uhealth*, vol. 12, p. e46282, 2024. DOI: https://doi.org/10.2196/46282

[5] A. Kos, Y. Wei, S. Tomažič, and A. Umek, "The role of science and technology in sport," *Procedia Computer Science*, vol. 129, pp. 489–495, 2018. DOI: https://doi.org/10.1016/j.procs.2018.03.029

[6] Z. Gao and J.E. Lee, "Emerging Technology in Promoting Physical Activity and Health: Challenges and Opportunities," *Journal of Clinical Medicine*, vol. 8, no. 11, p. 1830, 2019. DOI: https://doi.org/10.3390/jcm8111830

[7] M. R. Reshma, B. Kannan, V. P. Jagathy Raj, and S. Shailesh, "Cultural heritage preservation through dance digitisation: A review," *Digital Applications in Archaeology and Cultural Heritage*, vol. 28, p. e00257, 2023. DOI: https://doi.org/10.1016/j.daach.2023.e00257

[8] K.B. Gan, C.H. Chen and N.A.A. Aziz, "Upper Limbs Extension and Flexion Angles Calculation and Visualisation Using Two Wearable Inertial Measurement Units," *International Journal of Robotics and Automation Science*, vol. 4, pp. 1–7, 2022. DOI: https://doi.org/10.33093/ijoras.2022.4.1

[9] S. Guan, "Skeleton-based Human Action Recognition: From 3D Pose Estimation to Action Recognition," Ph.D. dissertation, University of Technology Sydney, 2023.

[10] S. Dubey and M. Dixit, "A comprehensive survey on human pose estimation approaches," *Multimedia Systems*, vol. 29, no. 1, pp. 167–195, 2023. DOI: https://doi.org/10.1007/s00530-022-00980-0

[11] T.F. Cootes, G. Edwards and C.J. Taylor, "Comparing Active Shape Models with Active Appearance Models," *Proceedings of the British Machine Vision Conference*, p. 18.1-18.10, 1999. DOI: https://doi.org/10.5244/C.13.18

[12] T.-D. Tran, X.-T. Vo, D.-L. Nguyen, and K.-H. Jo, "Combination of Deep Learner Network and Transformer for 3D Human Pose Estimation," *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pp. 174–178, 2022. DOI: https://doi.org/10.23919/ICCAS55662.2022.10003954

[13] Y. Sun, Z. Sun and W. Chen, "The evolution of object detection methods," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108458, 2024. DOI: https://doi.org/10.1016/j.engappai.2024.108458

[14] T. Abekoon et al., "A comprehensive review to evaluate the synergy of intelligent food packaging with modern food technology and artificial intelligence field," *Discover Sustainability*, vol. 5, no. 1, p. 160, 2024. DOI: https://doi.org/10.1007/s43621-024-00371-7

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

[16] A. Gupta, A. Anpalagan, L. Guan and A.S. Khwaja, "Deep learning for object detection and scene perception in self-

driving cars: Survey, challenges, and open issues," *Array*, vol. 10, p. 100057, 2021.
DOI: https://doi.org/10.1016/j.array.2021.100057

[17] A. Tharatipyakul, T. Srikaewsiew and S. Pongnumkul, "Deep learning-based human body pose estimation in providing feedback for physical movement: A review," *Heliyon*, vol. 10, no. 17, 2024.
DOI: https://doi.org/10.1016/j.heliyon.2024.e36589

[18] J. Wang et al., "Deep 3D human pose estimation: A review," *Computer Vision and Image Understanding*, vol. 210, p. 103225, 2021.
DOI: https://doi.org/10.1016/j.cviu.2021.103225

[19] E. Nishani and B. Cico, "Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation," *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4, 2017.
DOI: https://doi.org/10.1109/MECO.2017.7977207

[20] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Real-time Multi-Person 2D Pose Estimation using Part Affinity Fields," 2019.
DOI: https://doi.org/10.48550/arXiv.1812.08008

[21] B. Jo and S. Kim, "Comparative Analysis of OpenPose, PoseNet, and MoveNet Models for Pose Estimation in Mobile Devices," *Traitement du Signal*, vol. 39, no. 1, pp. 119–124, 2022.
DOI: https://doi.org/10.18280/ts.390111

[22] K. L. Lew, K. S. Sim, S. C. Tan, and F. S. Abas, "Virtual Reality Post Stroke Upper Limb Assessment using Unreal Engine 4.," *Engineering Letters*, vol. 29, no. 4, 2021.
URL:https://www.engineeringletters.com/issues_v29/issue_4/EL_29_4_24.pdf

[23] C. C. Lim, K. S. Sim, and C. K. Toa, "Development of Visual-based Rehabilitation Using Sensors for Stroke Patient," *International Journal of Robotics and Automation Science*, vol. 2, pp. 25–30, 2020.
DOI: https://doi.org/10.33093/ijoras.2020.2.4

[24] M. Too, S. H. Lau, and C. K. Tan, "Validity and Reliability of a Conceptual Framework on Enhancing Learning for Students via Kinect: A Pilot Test," *International Journal of Robotics and Automation Science*, vol. 6, no. 1, pp. 59–63, 2024.
DOI: https://doi.org/10.33093/ijoras.2024.6.1.8

[25] R. G. Candraningtyas, A. P. Yunus, and Y. H. Choo, "Human Fall Motion Prediction – A Review," *International Journal of Robotics and Automation Science*, vol. 6, no. 2, pp. 52–58, 2024.
DOI: https://doi.org/10.33093/ijoras.2024.6.2.8

[26] K. L. Lew, K. S. Sim, S. C. Tan, and F. S. Abas, "3D Kinematics of Upper Limb Functional Assessment Using HTC Vive in Unreal Engine 4," *Advances in Computational Collective Intelligence*, pp. 264–275, 2020
DOI: https://doi.org/10.1007/978-3-030-63119-2_22

[27] K.L. Lew, C.K. Toa, P. Zhou, C.S. Lee, T. Kurniawan, S.A. Babale and C. Zheng, "AI-Assisted Analysis for Breast Cancer Imaging and Diagnostics," *International Journal on Robotics, Automation and Sciences*, vol. 7, no. 1, pp. 111-119, 2025.
DOI: https://doi.org/10.33093/ijoras.2025.7.1.13