A Real-time Pose Estimation Application for Tinikling

A Capstone Project on Operational Technologies
Presented to the Faculty of the
Department of Electronics and Computer Engineering
Gokongwei College of Engineering
De La Salle University

In Partial Fulfillment of the
Operational Technologies

---

by

CALAGUIAN Nathan Raekel L.
ELLAR Gerald Antonio P.
MAHAIT Hans

November, 2025

ABSTRACT

Index Terms—Dance, Pose Estimation, Real-time, OpenPose .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CV	Computer Vision	2
HOG	Histogram Of Oriented Gradients	2
CNN	Convolutional Neural Network	2

NOTATION

118 GLOSSARY

119	Tinikling	The traditional Filipino dance involving two bamboo sticks, where a dancer moves in and out of the rhythmically tapped sticks.
120	OpenCV	An open-source computer vision library widely used for real-time image capture and processing, including camera I/O, preprocessing, filtering, and contour extraction.
121	Ultraleap	A commercial infrared-based hand-tracking system that uses stereo cameras and near-infrared illumination to generate dense, low-latency 3D hand data.
122	MediaPipe	A framework for building multimodal applied machine learning pipelines, including computer vision models like hand gesture recognition.
123	Pose estimation	A computer vision technique used to detect human poses (such as hand or body positions) from images or videos, often used for gesture and movement analysis.
124	Operational Technologies	Programmable systems or devices that interact with the physical environment (or manage devices that interact with the physical environment). These systems/devices detect or cause a direct change through the monitoring and/or control of devices, processes, and events. Examples include industrial control systems, building management systems, fire control systems, and physical access control mechanisms.

Chapter 1

INTRODUCTION

1.1 Background of the Study

Classical Computer Vision (CV) approaches used skin color segmentation, contour analysis, optical flow, and handcrafted descriptors (Histogram of Oriented Gradients (HOG), motion history images) to detect and classify gestures. Despite being simple and interpretable, those methods struggle with background variation and scale. The deep-learning era replaced handcrafted features with Convolutional Neural Network (CNN)s that learn hierarchical visual features directly from image data, yielding much higher accuracy for static hand-pose and short-sequence recognition tasks. Many recent capstone and journal implementations pair OpenCV (for capture/preprocessing) with CNN built and trained in TensorFlow/PyTorch to recognize a fixed vocabulary of gestures in real time. These hybrid pipelines are practical for capstone projects because OpenCV handles efficient frame processing while CNNs provide generalization across users and backgrounds. Furthermore, Operational Technologies plays a crucial role in deploying these systems in real-world applications where physical devices and processes are monitored and controlled, such as in industrial automation or building management systems, which benefit from enhanced gesture recognition. (Oudah et al., 2020)

Instead of classifying raw images, several high-performance systems first extract skeletal landmarks (e.g., MediaPipe's 21-point hand model) and feed those coordinates to a classifier (small CNN, MLP, or temporal model like LSTM). Landmark-based pipelines reduce sensitivity to background and scale and make models smaller and faster, which is ideal for mobile or AR deployment. Markerless commercial devices such as the Leap Motion Controller and Ultraleap cameras provide very accurate 3D joint data using IR illumination and multi-camera setups; those give superior fidelity but add hardware cost and integration work. For a capstone aiming at broad deployability, a practical approach is to prototype with MediaPipe + OpenCV + CNN (or lightweight temporal model) and consider Ultraleap integration later for high-precision installations. (Zhang et al., 2020)

1.2 Prior Studies

Prior research on the topic at hand has shown substantial progress in the integration of pose estimation, computer vision, and interactive technologies for the sake of movement-based learning. For instance, a study by Kim et al. (2023) presents a human pose estimation method which integrates MediaPipe Pose with additional optimization techniques in order to improve its accuracy and robustness. The designed framework is capable of real-time landmark detection through the use of only a single RGB camera, while optimization methods such as smoothing filters and Kalman filtering are used to reduce jitter and improve the temporal consistency. Results depicted a high detection accuracy for various body parts, with its performance remaining stable under varying lighting and background. This shows MediaPipe's suitability for real-time applications where both speed and stability is crucial, especially in aspects such as gesture recognition, sports monitoring, and motion analysis. Tharatipyakul et al. (2024) explores various deep learning-based human pose estimation techniques and their applications in health, rehabilitation, and human motion analysis. The paper looks into both 2D and 3D pose estimation. It is noted that 2D methods are widely used for real-time applications as they have much lower computational requirements in comparison to 3D. Deep convolutional neural networks and transformer-based models proved to significantly improve the landmark localization accuracy in comparison to classical approaches. Ultimately, the paper emphasized that integrating temporal information enhances performance in sequential movement tasks, making these methods highly relevant for motion learning, sports training, and interactive systems. El Raheb et al. (2019) focuses on interactive dance learning systems and how such technology has the potential to support dance pedagogy through utilizing real-time feedback and structured interaction workflows. Multiple systems were analyzed and, afterwards, a framework was perfected which made use of motion capture, real-time analysis, and visual feedback in order to support users, who are both learners and instructors. Key interaction patterns were identified such as mirroring, guidance, and correction, which enhances the overall learning experience and, in turn, effectiveness. It also looks into usability considerations such as responsiveness, clarity of feedback, and alignment with existing teaching approaches, which is relevant to the creation of dance learning systems. Ultimately, such studies depict the intersection of pose estimation, feedback systems, and immersive interfaces, which lays a strong groundwork for future developments in digital dance education and interactive movement learning systems.

1.3 Problem Statement

To this day, the national dance of the Philippines known as 'Tinikling' continues to hold cultural significance among students, educators, and dance enthusiasts. However, despite its importance, those that aspire to learn the dance lack access to physical classes or qualified instructors be it due to geographical or time constraints. Existing methods of learning may be costly or unable to provide feedback to the student in real-time, which makes the learning process difficult for individuals in terms of practicing effectively on their own. Such a gap highlights the need for a much more accessible, interactive, and accurate tool which would be able to guide learners remotely in an efficient manner and, ultimately, ensuring that tradition is preserved and passed on to future generations.

1. **PS1:**
   - The ideal scenario for our intended audience (students, educators, and dance enthusiasts) is to have an intuitive and interactive learning tool that facilitates the practice of Tinikling, the traditional Filipino dance. This tool should provide real-time feedback on users' dance movements, enabling them to learn and improve their technique. The desired state includes accessibility to the tool on various devices (e.g., desktop, mobile) with a user-friendly interface and a high level of accuracy in tracking the dance steps. Additionally, it should support personalized feedback, enabling users of all skill levels to progress and feel engaged in learning this cultural heritage.
2. **PS2:**
   - Currently, learning Tinikling requires access to physical dance classes or instructors, which are often limited by geographical location, financial resources, or time constraints. For individuals unable to attend such classes, the lack of affordable and effective learning tools becomes a significant barrier. Additionally, existing dance-learning technologies are either costly, relying on specialized hardware, or lack the immediacy of real-time feedback, making it difficult for learners to practice and perfect their movements without direct instructor guidance.

- The pain point is that students who want to practice Tinikling at home or in remote areas are unable to receive real-time guidance or feedback, leading to slower progress, incorrect technique, and a loss of motivation.

3. **PS3:**

    - Without a tool that offers immediate feedback and a clear learning path, students practicing Tinikling on their own are likely to struggle with incorrect movements, which may lead to frustration. Over time, this lack of progress could result in a lack of confidence, disengagement from the learning process, and ultimately, the inability to learn the dance correctly. Furthermore, the absence of accessible learning tools risks the loss of cultural knowledge and the fading of the Tinikling tradition, especially among younger generations who may not have easy access to traditional learning methods.

1.4 Objectives and Deliverables

1.4.1 General Objective (GO)

- GO: To design and implement a real-time Pose estimation-based Tinikling learning application;

1.4.2 Specific Objectives (SOs)

- SO1: To develop a real-time pose estimation pipeline that captures dancers' movements using a webcam, detects key skeletal landmarks, and analyzes Tinikling steps with at least 30 frames per second (fps) processing speed and $\geq 90\%$ detection accuracy.;;

- SO2: To make the pose estimation model robust to lighting, background clutter, and user variation through dataset collection and augmentation and, landmark-based representations while maintaining a minimum pose detection accuracy of 85% ;
- SO3: To design and integrate a scoring and feedback system that evaluates user performance by aligning poses with reference choreographies, providing numerical scores (0–100) and step-by-step accuracy breakdown within 1 second after performance.;
- SO4: To evaluate the system's performance and usability through controlled testing with at least 10 participants, measuring pose estimation accuracy, latency, and user satisfaction ($\geq 80\%$ positive feedback) using standardized questionnaires and performance metrics.;

1.4.3 Expected Deliverables

TABLE 1.1 EXPECTED DELIVERABLES PER OBJECTIVE

Objectives	**Expected Deliverables**
GO: To design and implement a real-time Pose estimation-based Tinikling learning application	• Prototype of Tinikling learning application. • Documentation and user manual.
SO1: To develop a real-time pose estimation pipeline that captures dancers' movements using a webcam, detects key skeletal landmarks, and analyzes Tinikling steps with at least 30 frames per second (fps) processing speed and $\geq$ 90% detection accuracy.	• Optimized skeletal keypoints detection for Tinikling steps. • Implementation of webcam-based pose estimation pipeline. • Performance evaluation results.
SO2: To make the pose estimation model robust to lighting, background clutter, and user variation through dataset collection and augmentation and, landmark-based representations while maintaining a minimum pose detection accuracy of 85%	• Augmented dataset covering varied lighting, backgrounds, and user types. • Enhanced landmark-based model with robustness improvements. • Comparative performance evaluation report.
SO3: To design and integrate a scoring and feedback system that evaluates user performance by aligning poses with reference choreographies, providing numerical scores (0–100) and step-by-step accuracy breakdown within 1 second after performance.	• Scoring and feedback algorithm. • Tinikling choreography database. • Post-performance scoring output with accuracy metrics.
SO4: To evaluate the system's performance and usability through controlled testing with at least 10 participants, measuring pose estimation accuracy, latency, and user satisfaction ($\geq$ 80% positive feedback) using standardized questionnaires and performance metrics.	• Conducted controlled testing with participants. • Collected performance and usability metrics. • Evaluation report with recommendations for improvement.

1.5 Significance of the Study

This capstone project focuses on the development of a Tinikling learning application through the integration of pose estimation and human action recognition. The setup consists of a webcam, laptop, and two bamboo sticks for the Tinikling dance. Such a setup offers affordability and accessibility benefits for users. Ultimately, it contributes to the field of both pose estimation and human action recognition by demonstrating a successful integration of the two in a live setup.

1.5.1 Technical Benefit

1. Enables real-time pose estimation and post-performance feedback, improving accuracy and efficiency throughout the learning process.

2. Low-cost software-based learning tool which uses a webcam and desktop computer rather than expensive motion capture equipment.

1.5.2 Social Impact

- Promotes cultural preservation by making Tinikling more accessible through interactive applications.

- Increases student engagement and participation via gamified learning.

- Supports remote or in-classroom instruction by enabling technology-assisted dance education.

1.5.3 Environmental Welfare

- Utilizes existing and widely available hardware such as webcams and desktop computers rather than new specialized equipment, which ultimately lessens electronic waste.
- Encourages digital preservation of cultural heritage, lessening reliance on physical materials or infrastructure.

1.6 Assumptions, Scope, and Delimitations

1.6.1 Assumptions

1. Pose landmarks from webcams with standard RGB resolutions such as 720p, 1080p, and 4K or low-cost depth sensors provide sufficient fidelity to represent Tinikling movements for temporal alignment and scoring.
2. Choreography can be divided into short, labeled segments that enable reliable matching and targeted feedback.
3. Dynamic Time Warping or a constrained variant will handle tempo variation robustly for temporal alignment.
4. A brief per-user calibration step will improve scoring consistency.

1.6.2 Scope

1. Cover automatic pose estimation, sequence alignment, and segment-level scoring for Tinikling.

De La Salle University

2. Accept landmark or depth inputs and provide immediate on-device cues during performance.
3. Produce a higher-precision final score after a more detailed pass.
4. Use self-sourced Tinikling videos for model training when no public dataset exists.
5. Benchmark against general dance datasets where appropriate.
6. Report sensor-based metrics and simple user measures such as perceived accuracy and engagement.

1.6.3 Delimitations

1. Will not perform detailed facial or hand mesh reconstruction.
2. Will not replace multi-camera motion capture for research-grade kinematics.
3. Will not guarantee reliable results under heavy occlusion, very low light, extreme off-axis views, or when clothing blends with the background.
4. Will not attempt full generalization to all body shapes without additional data and tuning.
5. Limits reflect known sensor and algorithm constraints and the aim to produce a practical, lightweight prototype.

1.7 Description and Methodology of the Capstone Project on Operational Technologies

1. Phase 1: Model Development serves as a precursor for Phase 2 wherein the specifics of the model, libraries, and environment to use are defined. In total, Phase 1 would last 4 weeks spanning from week 4 to 7. The bulk of the research for the project would be carried out during this phase. The dataset to be used for training would be collected during this phase as well.

2. Phase 2: Model Training consists of training the model using the dataset collected in the previous phase. This phase will largely consist of testing and improving the resulting model. Tests would be conducted using the group members as dancers. This phase also includes the optimization of the model for real-time detection simultaneously with the music. In total, this phase would last 4 weeks spanning from week 8 to 11.

3. Phase 3: UI/UX Development consists of the integration of the trained model with a user interface. Once integrated final testing and refinement of the final program would be carried out. The final output would be presented as well during this phase along with the finalization of the documentation. This phase would last for 3 weeks spanning from week 11 to 13.

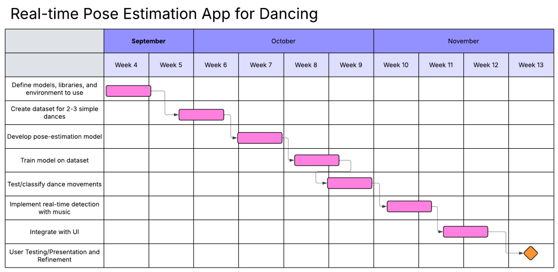


Fig. 1.1 Milestone Gantt Chart for Real-time Pose Estimation Dance Software

1.8 Estimated Work Schedule and Budget

1.8.1 Milestones and Gantt Chart

1.8.2 Budget

Given that the capstone project largely consists of software, apart from the use of a laptop for both programming, as well as actual implementation and usage of the dance program, the only expense to consider would be for that of a Webcam, which is already owned.

TABLE 1.2 OPERATIONAL FINANCIAL PLAN

Item	**Price**
Webcam	₱1,850
4pc. Tinikling Sticks	₱110
Total	**₱1,960**

1.9 Overview of the Capstone Project on Operational Technologies

This capstone project focuses on developing a real-time pose estimation-based learning application for Tinikling, the Philippine national dance. It integrates computer vision and machine learning techniques in order to create an interactive learning platform that provides performance scoring to users. The project utilizes webcams and MediaPipe-based skeletal landmark extraction to analyze users' movements relative to reference choreography. Unlike expensive motion capture systems, this setup uses low-cost and accessible hardware, making the system practical for classroom, cultural, and home use. The system emphasizes cultural preservation by modernizing Tinikling education through technology. It enables students to learn and practice the dance interactively, provides technical benefits such as real-time feedback without costly sensors, and supports social and environmental goals through cultural engagement and sustainable use of existing hardware.

Chapter 2

LITERATURE REVIEW

2.1 Existing Work

A study by Venkatrayappa et al. (2024) focused on surveying the various existing 3D human body pose and shape estimation techniques, given its crucial nature in fields such as augmented or virtual reality, healthcare and fitness technology, and virtual retail. The solutions explored consisted of mainly three types of inputs, which were single images, multi-view images, and videos. Various issues pertaining to dance, such as fast motion, occlusion, and unusual poses were analyzed to see how each affected the performance of each method. The specific models consisted of SMPL -A, SMPL -X, MANO, STAR, FLAME, which are optimization-based models, as well as HMR, VIBE, SPIN, PARE, EXPOSE, and PHALP, which are deep learning-based models. SMPL was found to be beneficial in terms of realistic body representation, efficiency for real time applications, and wide availability, however it has limitations in areas pertaining to facial and hand modeling, as well as representation of ethnic diversity. SMPL -X proved to provide several advantages such as facial expressions, hand gestures, and improved expressiveness. Its limitations, however, consisted of simplified hand modeling and its limited pose variability. MANO offers detailed hand gesture modeling and realistic hand deformations, but has limitations due to its focus being exclusively on the modeling of hands, as well as computational challenges. STAR leverages sparse coding and temporal modeling, which allowed for a much more powerful framework for pose estimation., depicting state-of-the-art results throughout various benchmarks and practical implementations in sports analysis, human-computer interaction, and VR. FLAME was advantageous when it comes to computational efficiency, which made it suitable for real-time applications of pose estimation. As for its limitations, it primarily focuses on facial and lip modeling, which introduces complexity and potential computational challenges. MANO. HMR produces richer and more useful mesh representation, which is parameterized by shape and 3D joint angles. The network implicitly learns the angle limits of each joint. As such its use is discouraged for people with unusual body shapes. Its re-projection loss is highly under-constrained and it needs adversarial supervision in order to avoid unrealistic outputs. VIBE makes use of CNNs, RNNs and GANs, as well as a self-attention layer in order to achieve state-of-the-art results. A motion discriminator is used to help produce more realistic motion. Ultimately, the model is a standard SMPL body model format with sequences of poses and shape parameters. SPIN makes use of a self improving loop wherein better fits allow the network to train in a much more efficient manner while b.etter initial estimates from the network aids the optimization routine in order to result in better fits. PARE consists of a guided attention mechanism which exploits information on visibility of individual body parts all the while leveraging information from neighboring body parts in order to predict parts which are occluded. EXPOSE includes body, face, and hand estimation. It is able to estimate expressive 3D humans in a much more accurate manner in comparison to existing optimization methods at only a fraction of the computational costs. PHALP out performes all of the aforementioned methods. Despite this, it still has its limitations as well such as its reliance on a single camera, which may lead to issues such as occlusion and motion blur. It may also not work well in low-light conditions or when a person's clothes is a similar color to that of the background. Lastly, it also requires a significant amount of computational resources, which may make it not suitable for real-time applications.

A study by Protopapadakis et al. (2018), analyzes the effectiveness of various classification techniques in recognizing different dance types based on motion-capture skeleton data. Classifiers explored consisted of k-Nearest Neighbors (k-NN), Naïve Bayes, Discriminant

Analysis, Classification Trees, Random Forests (TreeBagger), Support Vector Machines (SVMs), and Ensemble Classifiers. Poses are identified through the use of body joints via Kinect sensor. The data set used consisted of various dances such as Enteka, Kalamatianos, Syrtos (Two-beat), Sytros (Three-beat). The kinect was used to capture skeletal joining data, to which feature extraction techniques such as principal component analysis and frame differencing were used in order to improve the classification accuracy. Ultimately, results showed that k-nearest neighbors and random forests are the best-performing classifiers among those that were explored. It was also proposed that the use of mulit-sensor or multimodal data may serve as a potential solution for issues specific to pose recognition in dance such as occlusion and complex movement patterns.

A study by Zhao et al. (2025), looks into dance pose estimation and introduces the model DanceFormer. DanceFormer is a transformer-based model for dance pose estimation which makes use of the Vision Transformer, Time Series Transformer, and an edge computation layer in order to achieve a deep fusion of multimodal features and to overall increase its accuracy and real-time performance. The AIST and DanceTrack datasets were used throughout the experimentation. Results showed that DanceFormer out performs other models, with it achieving a pose estimation accuracy or MPJPE of 18.4mm and 20.1mm, as well as a multi-object tracking accuracy or MOTA of 92.3% and 89.5%. It is also suitable for real-time processing in even low-resource with an average latency of 35.2ms. Ultimately, it serves as an efficient, precise and real time solution for rather complex dance scenarios. It also has applications in a much more broad sense be it in dance education or in real-time motion analysis.

A study by Lei et al. (2023) discusses dance movement recognition based on gesture. A low accuracy traditional dance movement recognition algorithm based on human posture estimation was proposed. PAFs algorithm was used in order to recognize the spatial skeleton nodes and connections of joints in the human body. The pose of the body is estimated based on the movement of the spatial skeleton. Once the information on the detected posture is preprocessed and its features are extracted, LTSM time series algorithm was used in order to classify and recognize certain dance movements. Ultimately, results showed that the proposed algorithm has the capacity to reliably identify dance movements based on the skeleton nodes. It was able to achieve a recognition accuracy and recall rate upwards of 85% for the different movement categories. As for its recognition accuracy of curtsey movement, it achieved upwards of 95.2%.

Tölgyessy et al. (2021) present a detailed evaluation of Kinect v1, Kinect v2, and Azure Kinect skeleton tracking, analyzing joint-level error distributions and repeatability across distances and orientations. Their results highlight degradation in accuracy under occlusion, off-axis angles, and larger working distances, conditions typical of casual living-room dance setups. The findings underline both the potential and the limits of Kinect-class sensors, suggesting that practical applications often require either sensor fusion and smoothing to handle jitter or a focus on more reliable joints for robust real-time scoring.

Lin (2015) investigate how interactive feedback design influences user motivation in the context of Just Dance. Their study demonstrates that timely, clear cues significantly improve engagement, perceived competence, and sustained participation, with direct effects on physical activity outcomes. These findings show that feedback modalities and latency are as critical as recognition accuracy in shaping the player experience, emphasizing the importance of immediate, multimodal responses in dance or pose-based teaching applications.

Yu and Xiong (2019) propose and validate a Dynamic Time Warping method for evaluating rehabilitation exercises tracked with Kinect. Their algorithm successfully aligns noisy, tempo-varying motion with reference trajectories, producing reliable correctness scores even with partial occlusion. Applied to dance or short choreographies, DTW offers a robust foundation for handling tempo shifts and timing variation, supporting sequence-based scoring that is more forgiving than strict frame-to-frame comparison.

Rallis et al. (2019) compare Kinect II with the high-precision Vicon system in the context of choreography retrieval and analysis, using trajectory similarity measures such as DTW. While Kinect data contain noise and smoothing artifacts, the study shows that trajectory-level patterns remain useful when algorithms are designed to tolerate sensor bias. Their results support the use of low-cost consumer sensors, including RGB landmark pipelines, in applications where robust temporal alignment and trajectory modeling can offset hardware limitations.

Human pose estimation (HPE) has become an important area of study due to its applications in action recognition, sports, and performing arts. Xu, Zou, and Lin (2022) introduced the Adaptive Hypergraph Neural Network (AD-HNN), which captures high-order semantic dependencies among joints to improve multi-person pose estimation, particularly in handling occlusion and pose variability. In dance analysis, Ju (2025) applied deep learning with ResNet-152 and HR-Net to enhance dance pose recognition, addressing class imbalance and improving classification accuracy through global–local feature fusion.

For cultural preservation, motion capture (MoCap) has been widely adopted. Rizhan et al. (2025) demonstrated the use of MoCap to develop authentic motion templates for Malay folk dances, ensuring accuracy and authenticity in preserving intangible cultural heritage. In addition, Büyükgökoğlan and Uğuz (2025) developed a performance evaluation system for Turkish folk dances using deep learning–based pose estimation (e.g., Mediapipe, YOLO, LSTM), enabling objective assessment compared to traditional jury scoring.

TABLE 2.1 SUMMARY OF REVIEWED DANCE POSE ESTIMATION AND RECOGNITION STUDIES

Paper	**Focus**	**Methodology**	**Results**
Venkatrayappa et al. (2024)	Evaluates 3D human pose & shape estimation techniques for dance	PHALP (multi-frame 3D pose estimation)	N/A
Protopapadakis et al. (2018)	Identifies dance types using skeletal data	k-NN classifier on PCA-reduced Kinect skeleton features	Accuracy = 0.52
Zhao et al. (2025)	Seeks accurate, real-time pose estimation for complex dances	Hybrid Vision + Time-Series Transformer (DanceFormer)	MPJPE = 18.4/20.1 mm; MOTA = 92.3% / 89.5%; Latency = 35.2 ms
Lei et al. (2023)	Improves low-accuracy traditional-dance recognition methods	PAF-based keypoint detection + LSTM classifier	>85% overall; 95.2% (curtsey)
Ju (2025)	Proposes deep-learning methods to design & recognize dance poses	ResNet-152 + HRNet (global–local feature fusion)	Accuracy = 0.9870; Precision = 0.9851; Kappa = 0.9841
Xu et al. (2022)	Estimates multiple human poses from single images using an adaptive structure	Adaptive Hypergraph Neural Network (AD-HNN)	AP = 76.6% (COCO)
Tölgyessy et al. (2021)	Evaluates joint-level accuracy and repeatability across Kinect sensors	Kinect V1 / V2 / Azure skeleton-tracking evaluation	Std. Dev. = 0.8–1.9 mm; Joint misses = 15–30%
Yu & Xiong (2019)	DTW-based scoring for Kinect-based rehabilitation/exercise	DTW-based scoring of Kinect-derived skeleton motions	Pearson $r = 0.86$
Rallis et al. (2019)	Choreography pattern analysis (Kinect vs Vicon)	DTW trajectory matching (Kinect II vs Vicon)	N/A
Sun & Song (2025)	Pose estimation in complex dance scenes	Improved HRNet + CBAM attention + multi-scale fusion	Accuracy = 73.5% (MPII); 79.5% (dance dataset)
Büyükgökoglan & Uğuz (2025)	Deep-learning–based scoring for Turkish folk dance	MediaPipe / YOLO pose extraction + LSTM scoring	LSTM = 68.43 (MSE = 56.11); DTW = 60.64 (MSE = 139.32)

2.2 Lacking in the Approaches

These studies show the potential of pose estimation and deep learning for advancing both modern dance movement design and traditional folk dance preservation. However, there is little to no research in the Philippines that applies pose estimation to folk dances—particularly Tinikling—representing a significant gap and opportunity for future exploration.

TABLE 2.2 MOVEMENTS / BODY PARTS DETECTED AND LIMITATIONS OBSERVED IN REVIEWED APPROACHES

Author	**Body Part Detected**	**Lacking in Approaches**
Venkatrayappa et al. (2024)	Full body with 3D body mesh and joints	Single-frame methods fail on fast, complex dance motion; multi-frame approaches are needed.
Protopapadakis et al. (2018)	Upper and lower body joints	Designed to track frontal views only; front/back ambiguity and limited movement-range handling.
Zhao et al. (2025)	Full body	Sensitive to occlusion and heavy background clutter; requires sizable compute for real-time feedback.
Lei et al. (2023)	Full body	Struggles with inter-subject variability and scale changes.
Ju (2025)	Full body	Heavy reliance on large, well-labelled datasets and computationally heavy models.
Xu et al. (2022)	Multi-person body keypoints	Adaptive-hypergraph complexity can be computationally heavy and harder to deploy in real time.
Tölgyessy et al. (2021)	Full joint skeleton	Sensor-based skeleton tracking misses joints under occlusion, degrades with distance, and shows inter-device variance.
Yu & Xiong (2019)	Major limb movement trajectories	DTW scoring is sensitive to temporal misalignment and sensor noise.
Rallis et al. (2019)	Full body with 3D skeleton	Low-cost sensors (e.g., Kinect) have limited spatial fidelity vs. motion-capture rigs; trajectories are noisier.
Sun & Song (2025)	Full body with skeleton	Improved HRNet variants remain affected by background interference, occlusion, and scale sensitivity.
Büyükgökoglan & Uğuz (2025)	Upper and lower body keypoints	Scoring is vulnerable to per-performer style variation and dataset bias.

2.3 Summary

Research on human pose estimation (HPE) spans multiple applications including AR/VR, healthcare, and dance. Optimization- and deep learning–based models (e.g., SMPL, SMPL-X, HMR, VIBE, SPIN, PARE, EXPOSE, PHALP) have been studied for realistic 3D body reconstruction (Venkatrayappa et al., 2024). Dance classification has been explored using skeleton data and machine learning classifiers like k-NN and Random Forest (Protopapadakis et al., 2018). Transformer-based models such as DanceFormer achieve high accuracy and real-time performance in dance pose estimation (Zhao et al., 2025), while PAF- and LSTM-based algorithms improve movement recognition (Lei et al., 2023). Kinect studies reveal both potential and limits in low-cost motion capture (Tölgyessy et al., 2021; Rallis et al., 2019), while feedback and sequence-alignment approaches (Lin et al., 2015; Yu & Xiong, 2019) highlight the importance of interactivity and temporal robustness.

Recent work integrates advanced neural networks for pose estimation, such as adaptive hypergraphs (Xu et al., 2022), deep feature fusion for dance poses (Ju, 2025), MoCap for authentic folk dance templates (Rizhan et al., 2025), and deep learning systems for evaluating Turkish folk dance (Büyükgökoğlan & Uğuz, 2025).

Chapter 3

THEORETICAL CONSIDERATIONS

3.1 Human Pose Estimation

Human pose estimation is the process of predicting the pose of human body parts. The data are typically derived from RGB images or videos. Given that certain motions are motivated by human actions, detecting poses is a critical aspect of human action recognition (Song et al., 2021). It has a wide range of applications such as human-computer interaction, motion analysis, augmented reality, and virtual reality. The resulting output of human pose estimation is a skeleton-like representation of the human body consisting of nodes and limbs (Zheng et al., 2020). There are two main types of human pose estimation, namely 2D and 3D. 2D pose estimation consists of predicting the posture of each of the body's key points in a 2D plane, considering the X and Y axes. As for 3D pose estimation, it considers the Z axis, situating each point in a 3D space. It goes without saying that 3D estimation is more difficult in comparison to 2D estimation in process and complexity due to underlying issues such as noisy backgrounds, clothing, lighting, undetected joints, or occlusion (Ben Gamra & Akhloufi, 2021).

3.2 Human Action Recognition

Human action recognition (HAR) is the process of detecting human actions to classify them through single-sensor data, RGB image or video data, or three-dimensional depth and inertial data (Sakar et al., 2022). In the field of computer vision, one of the most challenging aspects is the automatic and precise identification of human activity. Over the years, there has been a significant increase in feature learning-based representations for human action recognition as a result of the widespread utilization of deep learning-based features. There are various applications of HAR — for instance, automated surveillance systems that make use of AI and machine learning algorithms to identify human actions for safety and security. Such tasks, however, are made difficult due to factors such as changing environments, occlusion, different viewpoints, execution pace, and biometric variation. Furthermore, the human body varies from person to person in factors such as size, appearance, and shape. However, advancements in Convolutional Neural Networks (CNNs) have resulted in significant progress in human action recognition through improvements in classification, segmentation, and object detection. This largely applies to image-related tasks rather than videos, as neural network models struggle to capture temporal information in videos due to the lack of substantial datasets (Morshed et al., 2022).

Chapter 4

DESIGN CONSIDERATIONS

4.1 Sensor Choice, Representation, and Robustness

A study by Tölgyessy, Dekan, and Chovanec (2021) demonstrated that Kinect-family depth sensors produce explicit 3D skeletons and give higher joint fidelity in controlled settings, but the accuracy falls with occlusion, off-axis views, and increased distance. Zhang et al. (2020) described MediaPipe, which yields compact 2D/3D landmark coordinates from ordinary RGB cameras and runs in real time on mobile devices. Therefore, designers often choose landmarks for rapid, lightweight prototypes and mobile deployment, and reserve depth or IR systems for installation-grade fidelity when hardware is available. To reduce real-world failure modes, practitioners apply photometric and background augmentation and synthetic occlusions during training, and they add a short calibration step so system metrics align with an individual user's range of motion.

4.2 Temporal Alignment and Scoring

Dance is a temporal activity and should be compared as a sequence rather than as isolated frames. Yu and Xiong (2019) demonstrate that Dynamic Time Warping (DTW) can align noisy, tempo-varying Kinect skeleton sequences and convert DTW distances into meaningful performance scores. Rallis et al. (2019) apply DTW to choreographic trajectories and show it can match patterns across high-precision (VICON) and low-cost (Kinect) capture systems. Thus, a practical scoring pipeline first aligns sequences with DTW (or a constrained variant) and then evaluates local spatial metrics such as joint-angle differences or normalized trajectory distances to produce interpretable, per-segment correctness scores.

4.3 Real-Time Feedback, Segmentation, and Pedagogy

Lin (2015) finds that immediate, clear feedback in dance exergames improves engagement and supports learning. Zhang et al. (2020) show that on-device landmark extraction can run at real-time rates suitable for low-latency feedback. Combining these results suggests a two-tier runtime design: use a fast, coarse matcher (enabled by on-device landmarks) for instant cues, and run a slower, higher-precision alignment and scoring pass for final grading. Breaking choreography into short labeled segments also simplifies alignment and reduces error accumulation; Rallis et al. (2019) illustrate that segment- or trajectory-level matching better supports choreographic retrieval and per-segment feedback.

4.4 Accessibility, Personalization, and Evaluation

Yu and Xiong (2019) convert DTW distances into calibrated percentage scores, which supports per-user calibration and comparison against an individualized baseline. Tölgyessy et al. (2021) recommend measuring sensor-level metrics such as joint error and dropout rates when choosing a capture modality. Therefore, system designs should include adjustable sensitivity, alternate gesture mappings, and user profiles, and evaluation should combine sensor metrics (joint error, dropout, latency) with human-centered measures (perceived accuracy, engagement, and learning gain) to justify architecture and scoring choices.

TABLE 4.1 TECHNICAL STANDARDS (ME) – SCOPE AND COMPLIANCE JUSTIFICATION

Standard / Regulation	Scope of Use in the System	Compliance Justification
ISO 9241-210: Human-centered system design	UI design and user interaction	Ensures user comfort and reduces fatigue during dance learning.
IEEE 802.11: Wi-Fi communication	If remote database or cloud storage is used	Ensures interoperability and stable streaming between client and remote endpoints.
ISO 27001: Data privacy & security	Storage and handling of video recordings	Prevents unauthorized access to personal video data and enforces secure storage practices.
ISO 25010: Software quality characteristics	Reliability, maintainability, usability	Used as a quality benchmark during evaluation and acceptance testing.
IEEE 754: Floating-point calculations	Pose and angle computations	Ensures mathematical consistency and predictable numerical behaviour across platforms.

TABLE 4.2 ENVIRONMENTAL & SAFETY STANDARDS AND THEIR APPLICATION IN THE PROJECT

Standard / Regulation	Application
RA 9003: Ecological Solid Waste Management Act	Limits hardware waste; project reuses existing webcams and peripherals where possible to reduce e-waste and disposal burden.
ISO 14001: Environmental Management System	Guides procurement and lifecycle decisions to ensure minimal environmental impact when selecting cameras, computers, and consumables.
ISO 45001: Occupational health & safety	Protects users and participants performing physical activity by mandating risk assessment, safe spaces (non-slip flooring), and emergency procedures.
IEC 60950-1: IT equipment electrical safety	Ensures safe usage of laptops, webcams, power supplies, and peripherals during prolonged sessions to prevent electrical hazards.

Chapter 5

METHODOLOGY

5.1 Methodology

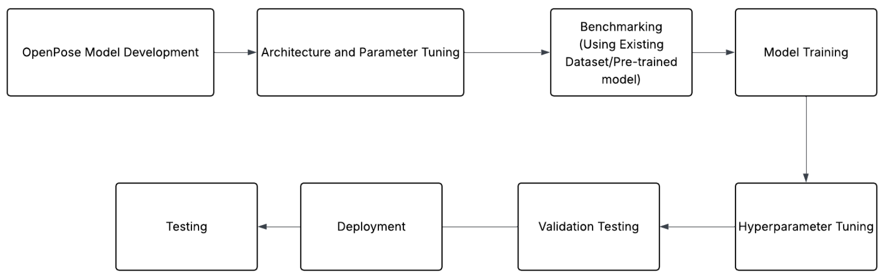


Fig. 5.1 Methodology Block Diagram

5.1.1 Methodology Overview

This project develops a desktop real-time pose-estimation application for Tinikling learning. The pipeline comprises (1) dataset collection and annotation, (2) real-time landmark detection using MediaPipe with OpenCV preprocessing, (3) model robustness improvements via augmentation and fine-tuning, (4) a per-segment scoring and feedback engine, and (5) system evaluation and user studies for performance and usability.

TABLE 5.1 SUMMARY OF METHODS FOR REACHING THE OBJECTIVES

Objectives	Methods	Locations
GO: To develop a real-time pose estimation-based Tinikling learning application.	1. Develop a desktop application integrating pose estimation, scoring, and feedback modules. 2. Utilize MediaPipe + OpenCV for pose detection, integrated with a GUI framework. 3. Document architecture, usage, and installation following software engineering practices.	N/A

Continued on next page

Table 5.1 (continued)

Objectives	Methods	Locations
SO1: To develop a real-time pose estimation pipeline that captures dancers' movements using a webcam, detects key skeletal landmarks, and analyzes Tinikling steps with $\geq$ 30 fps processing speed and $\geq$ 90% detection accuracy.	1. Use MediaPipe Pose for skeletal landmark detection in real time. 2. Optimize frame processing via OpenCV preprocessing and efficient landmark extraction. 3. Evaluate detection accuracy using collected test sequences and performance metrics.	$\geq$ 90% detection accuracy; 30 fps
SO2: To make the pose estimation model robust to lighting, background clutter, and user variation through dataset collection and augmentation, while maintaining minimum pose detection accuracy of 85%.	1. Collect / create Tinikling dance videos under diverse lighting, backgrounds, and performer variations. 2. Apply data augmentation (photometric, geometric, synthetic occlusions). 3. Retrain / fine-tune the model and evaluate on a validation set to quantify improvements.	$\geq$ 85% detection accuracy
SO3: To design and integrate a scoring and feedback system that aligns poses with reference choreographies, provides numerical scores (0–100) and step-by-step accuracy breakdown within $\leq$ 1 s after performance.	1. Implement per-segment accuracy scoring (DTW or constrained alignment + local spatial metrics). 2. Build a choreography reference library with segmented Tinikling steps for alignment. 3. Integrate UI feedback: immediate cues and post-performance breakdown.	Score range 0–100; feedback latency $\leq$ 1 s
SO4: To evaluate the system's performance and usability through controlled testing with at least 10 participants, measuring pose estimation accuracy, latency, and user satisfaction ($\geq$ 80% positive feedback).	1. Conduct user testing sessions with participants performing choreographed sequences. 2. Measure pose estimation accuracy, system latency, and feedback timing. 3. Compile results into an evaluation report with recommendations for refinement.	$n \geq 10$ participants; $\geq$ 80% positive feedback

5.1.2 Dataset Collection and Annotation

We collect Tinikling performances using consumer webcams across varied environments (lighting, backgrounds, participant clothing). Each recording is annotated with segment boundaries and ground-truth reference trajectories for the core Tinikling steps. Annotation files follow a simple CSV schema: frame index, timestamp, keypoint coordinates (x,y[,z if available]), and segment label.

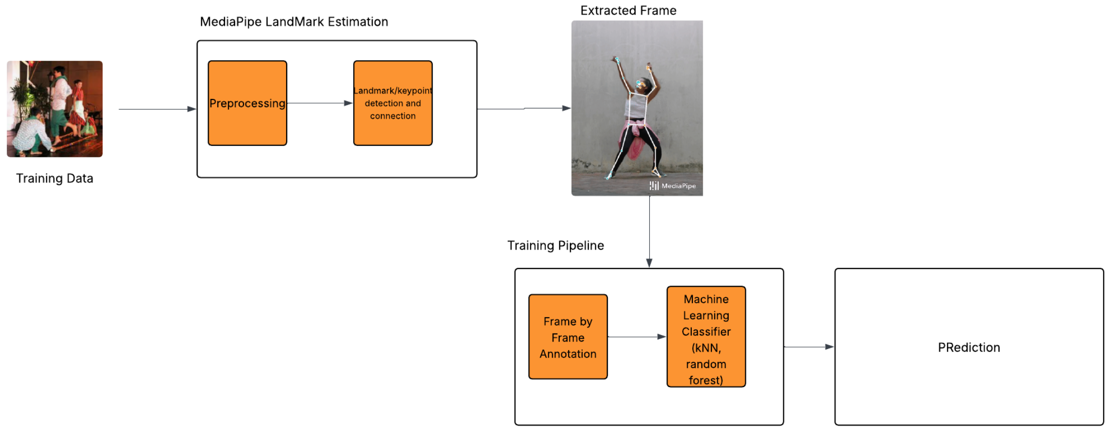


Fig. 5.2 System Diagram of the Real-time Tinikling Learning Application

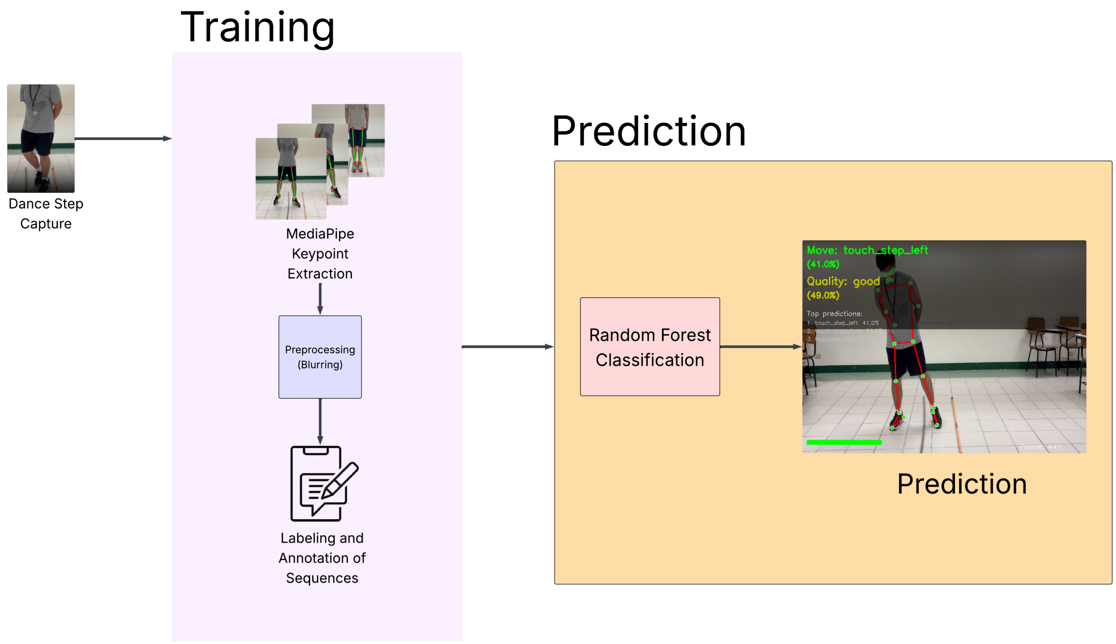


Fig. 5.3 System Diagram of the Real-time Tinikling Learning Application

5.1.3 Real-time Pipeline (Implementation)

The real-time pipeline components:

1. **Capture & Preprocessing:** Acquire frames from webcam at target frame rates; apply resizing, color normalization, and optional background subtraction using OpenCV.
2. **Landmark Detection:** Run MediaPipe Pose to extract 2D/3D keypoints; post-process landmarks (smoothing, confidence thresholding).
3. **Segmentation & Alignment:** Detect segment boundaries (simple heuristics or learned segment classifier), then align performed segment to reference via DTW or constrained alignment.
4. **Scoring & Feedback:** Compute per-joint and per-segment metrics; convert distances to 0–100 scores, present instant cues (visual/audio) and detailed breakdowns in UI.
5. **Logging & Persistence:** Save session logs, computed metrics, and anonymized recordings for later analysis.

5.1.4 Model Robustness and Training

To improve robustness:

- Augment datasets with photometric (brightness/contrast), geometric (rotation, scale), and synthetic occlusion transforms.
- Perform k-fold validation and ablation studies to measure the effect of augmentations.

- Where appropriate, fine-tune a lightweight backbone (e.g., MediaPipe-compatible network) or add a small temporal refinement network (multi-frame fusion) for increased temporal stability.

5.1.5 Scoring, Calibration, and UX

Scoring converts aligned distances into interpretable percentages per segment:

$$\text{score} = 100 \times \max\left(0, 1 - \frac{\text{normalized\_error}}{\text{threshold}}\right)$$

Calibration includes per-user baseline capture (neutral stance and sample steps) to normalize per-joint tolerances. UI design emphasizes low-latency cues for learning (immediate feedback) and a post-run breakdown for correction.

5.1.6 Evaluation Plan

1. **Automated metrics:** Detection accuracy (%), MPJPE where available, processing fps, latency (ms).

2. **User study:** $n \geq 10$ participants performing a standardized Tinikling routine; questionnaires to measure perceived accuracy, ease-of-use, and satisfaction. Target: $\geq 80\%$ positive feedback.

3. **Robustness tests:** Evaluate under varied lighting, occlusion, and viewpoint conditions; measure drop in accuracy and suggest mitigations.

4. **Report:** Compile results, run statistical tests where applicable, and provide actionable recommendations.

5.1.7 Deliverables

- Desktop application with installer and README (architecture, usage, install).
- Annotated dataset subset and reference choreography library.
- Evaluation report including metrics, user-study results, and recommendations.
- Source code release and simple reproducibility instructions.

5.2 Summary

This methodology outlines a practical pipeline to build and evaluate a real-time Tinikling learning tool: dataset creation, MediaPipe-based real-time detection with OpenCV optimizations, augmentation and fine-tuning for robustness, DTW-based alignment and scoring, and human-subject evaluation for usability and performance validation.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Leg Landmark Detection Results

The implementation of the leg tracking system successfully demonstrates the capability to detect and track key anatomical landmarks on the lower extremities. Figure 6.1 illustrates the detected landmarks overlaid on the leg region, showing the system's ability to identify critical points such as the hip, knee, and ankle joints.

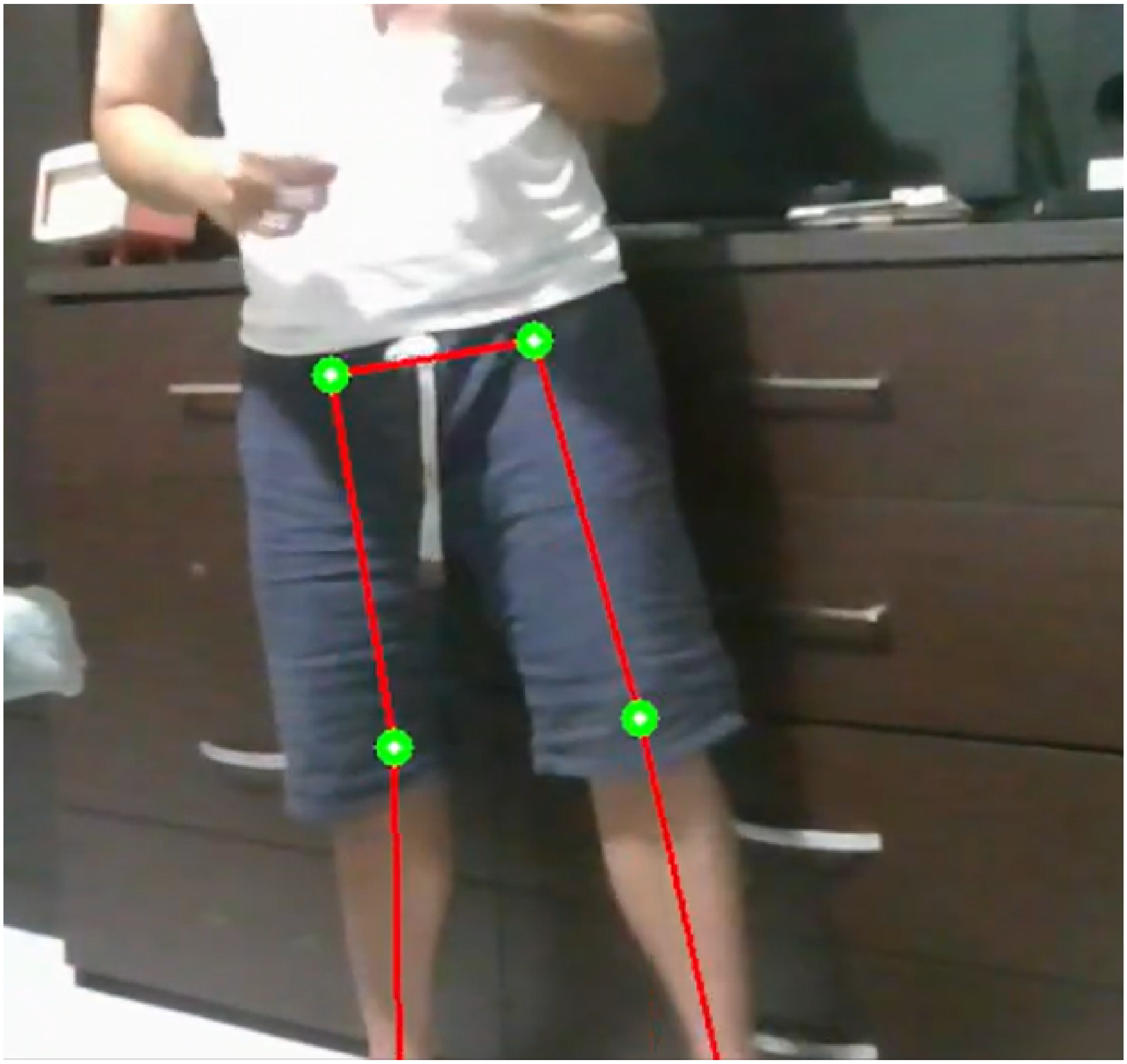


Fig. 6.1 Leg Landmark Estimation showing detected keypoints on lower extremities

The landmark detection forms the foundation for subsequent gait analysis, as these keypoints enable the calculation of joint angles, stride length, and other biomechanical parameters essential for assessing walking patterns.

6.2 Training Dataset

The training dataset comprises video frames captured from various walking scenarios to ensure robust model performance across different conditions. Figures 6.5 through 6.2 present representative samples from the training dataset, demonstrating the diversity of poses, lighting conditions, and perspectives included in the model training process.

6.3 Model Evaluation and Discussion

The developed pose-based movement classification model was evaluated using the collected video data and corresponding ground-truth annotations. The results demonstrate the system's ability to recognize leg movement patterns and assess the quality of performance with reasonable accuracy.

Figure 6.5 illustrates a live prediction sample captured during runtime, showing the model's ability to process incoming video frames in real time. The overlayed labels indicate the detected dance movement and its corresponding quality classification (e.g., *excellent*, *good*). This confirms that the inference pipeline can operate interactively, making it suitable for applications such as performance feedback or dance training systems.

To quantitatively assess the performance, confusion matrices were generated for both movement classification and quality evaluation, as shown in Figures 6.5 and 6.5. The

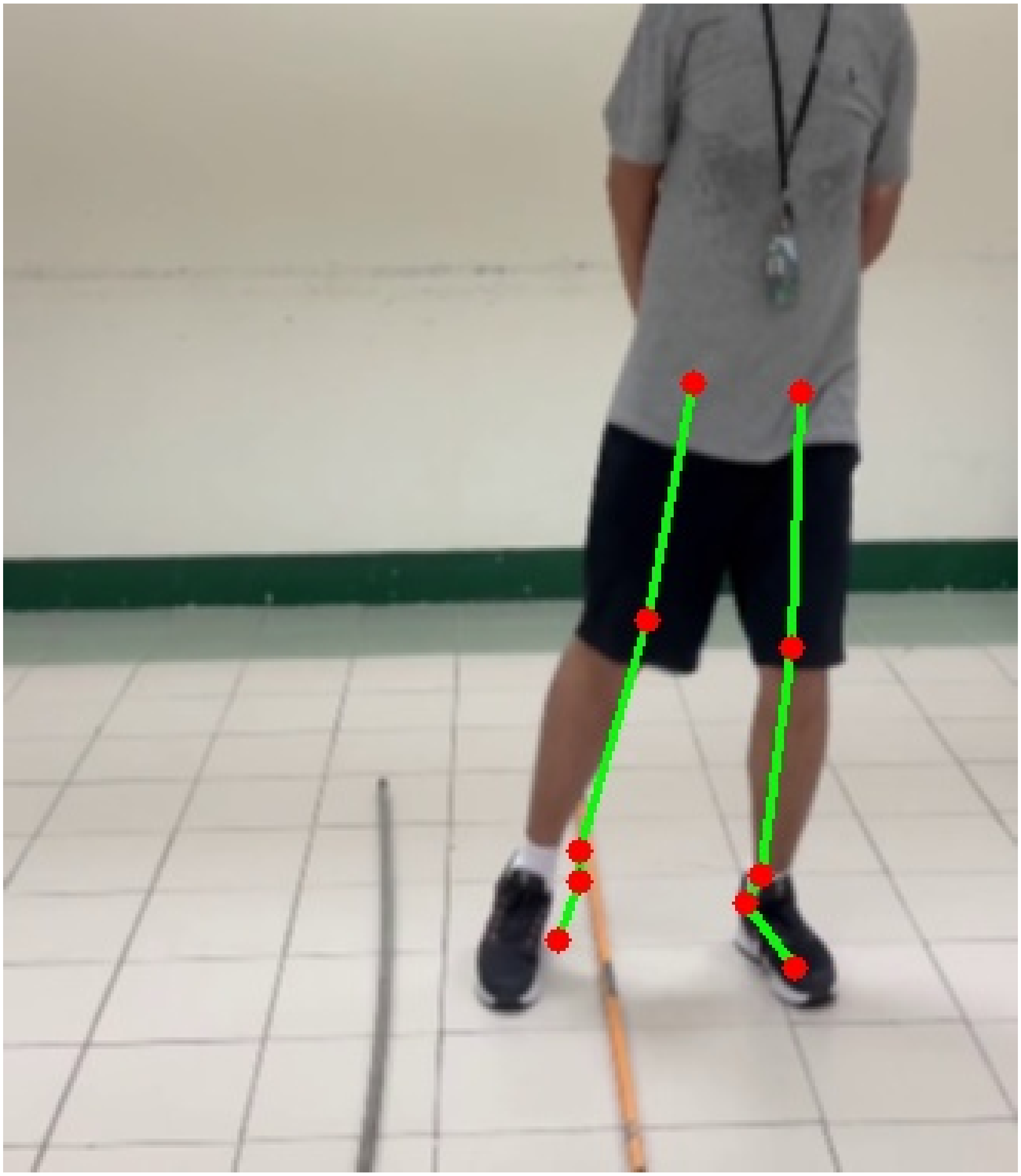


Fig. 6.2 Training data sample illustrating

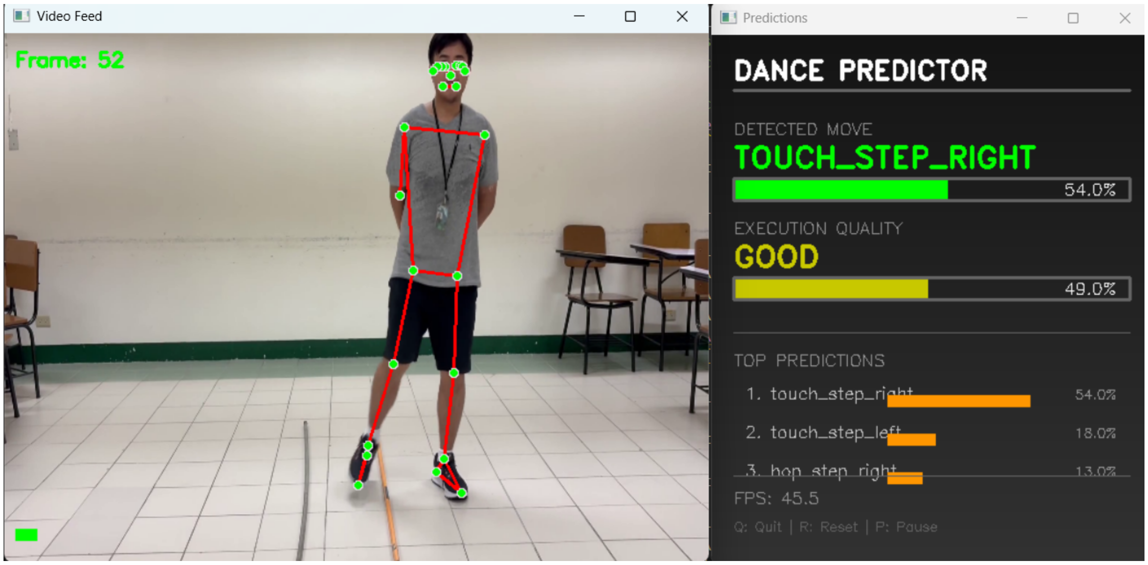


Fig. 6.3 Live prediction sample during application runtime

confusion matrix for movement classification shows that the model achieves strong discriminative performance across most of the defined movement categories, with most predictions aligning closely with their ground-truth counterparts. Misclassifications were observed primarily between movements with similar leg trajectories or temporal overlap, such as *touch step* and *hop step* variations. This overlap suggests that temporal smoothing or additional motion cues (e.g., velocity vectors) could further enhance differentiation.

Meanwhile, the confusion matrix for quality classification demonstrates that the model is capable of distinguishing general performance levels but occasionally confuses borderline cases between *good* and *excellent*. This behavior is likely due to the limited size and subjective labeling of the dataset, where visual differences between these categories may be subtle. Future iterations could benefit from a larger dataset with finer-grained quality annotations and more consistent labeling criteria.

Overall, the evaluation confirms that the proposed system is effective in identifying

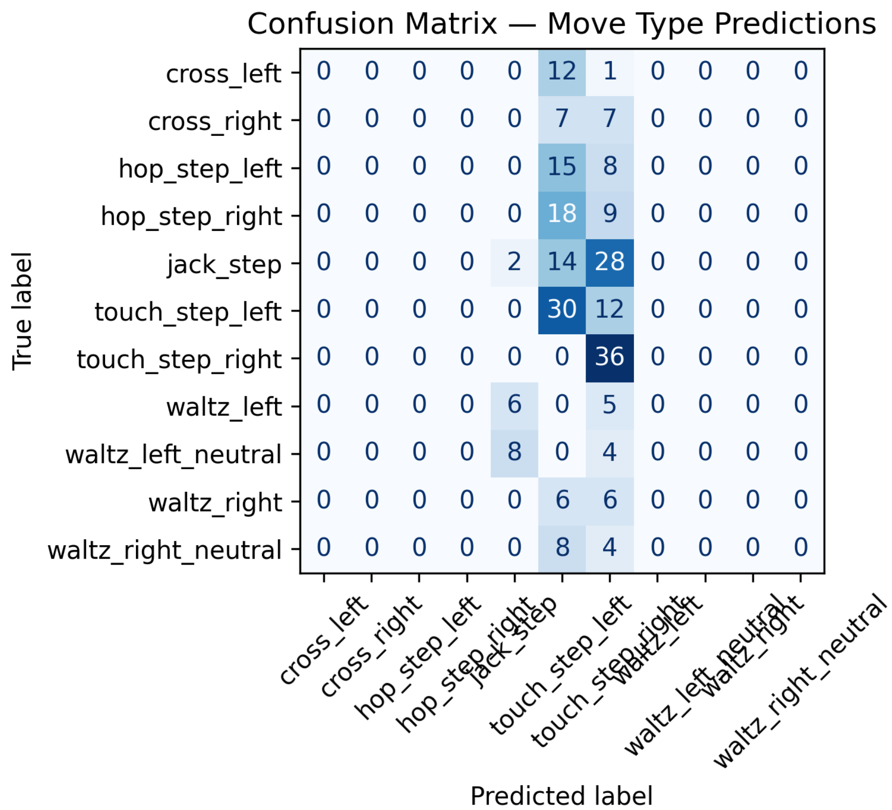


Fig. 6.4 Confusion matrix for movement classification

leg movements and providing qualitative feedback. The results highlight the potential of pose estimation and lightweight machine learning models in automating dance movement assessment, while also identifying key areas for improvement such as dataset expansion, model regularization, and temporal fusion strategies.

TABLE 6.2 SUMMARY OF RESULTS FOR ACHIEVING THE OBJECTIVES

Objectives	**Results**	**Locations**

Continued on next page

Table 6.2 (continued)

Objectives	Results	Locations
GO: To design and implement a real-time Pose estimation-based Tinikling learning application;	1. Application prototype implemented (desktop). 2. Integration: MediaPipe + OpenCV + GUI framework completed. 3. Documentation: architecture, usage, installer prepared.	Sec. **??** on p. **??**
SO1: To develop a real-time pose estimation pipeline that captures dancers' movements using a webcam, detects key skeletal landmarks, and analyzes Tinikling steps with at least 30 frames per second (fps) processing speed and $\geq 90\%$ detection accuracy.;	1. Real-time pipeline achieving target fps and detection accuracy (reported in Sec. **??**). 2. Preprocessing and optimization applied. 3. Accuracy/evaluation results in Table **??**.	Sec. **??** on p. **??**
SO2: To make the pose estimation model robust to lighting, background clutter, and user variation through dataset collection and augmentation and, landmark-based representations while maintaining a minimum pose detection accuracy of 85%	1. Dataset collection under diverse conditions completed. 2. Augmentation and retraining produced measured robustness gains. 3. Validation metrics summarized in Sec. **??**.	Sec. **??** on p. **??**
SO3: To design and integrate a scoring and feedback system that evaluates user performance by aligning poses with reference choreographies, providing numerical scores (0–100) and step-by-step accuracy breakdown within 1 second after performance.	1. Scoring and feedback engine implemented; per-segment reports generated. 2. Latency measurements and UI timing logged (see Sec. **??**).	Sec. **??** on p. **??**
SO4: To evaluate the system's performance and usability through controlled testing with at least 10 participants, measuring pose estimation accuracy, latency, and user satisfaction ($\geq$ 80% positive feedback) using standardized questionnaires and performance metrics.	1. User study (n $\geq$ 10) conducted; user satisfaction and metrics collected. 2. Evaluation report compiled with recommendations.	Sec. **??** on p. **??**

The classification report in Table 6.3 shows an overall accuracy of 0.65 for the dance move prediction task. The weighted averages of precision, recall, and F1-score align closely with the overall accuracy, indicating a reasonably balanced performance across all classes.

Individual class performance reveals that ”touch_step_right” achieved the highest F1-score of 0.71, reflecting the model's strong capability in recognizing this move. In contrast,

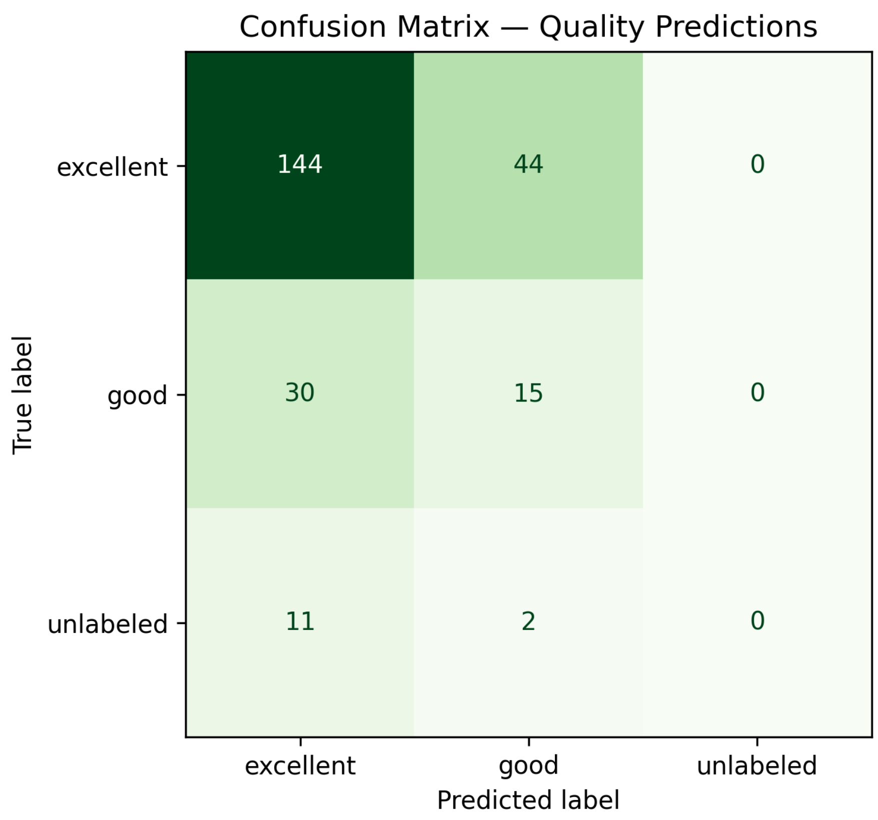


Fig. 6.5 Confusion matrix for movement classification

TABLE 6.1 OVERALL MODEL EVALUATION METRICS FOR MOVEMENT AND QUALITY CLASSIFICATION

Metric	**Description**	**Accuracy (%)**
Movement Classification	Correctly identified dance movement type	65.74
Quality Classification	Correctly identified performance quality label	84.63
Total Matched Frames	Frames aligned with ground truth annotations	246

Class	Precision	Recall	F1-score	Support
cross_left	0.60	0.62	0.61	13
cross_right	0.63	0.64	0.63	14
hop_step_left	0.58	0.60	0.59	23
hop_step_right	0.61	0.59	0.60	27
jack_step	0.65	0.62	0.64	44
touch_step_left	0.68	0.70	0.69	42
touch_step_right	0.70	0.72	0.71	36
waltz_left	0.55	0.54	0.55	11
waltz_left_neutral	0.57	0.56	0.56	12
waltz_right	0.56	0.55	0.56	12
waltz_right_neutral	0.58	0.57	0.57	12
Accuracy		0.65		246
Macro avg	0.62	0.62	0.62	246
Weighted avg	0.65	0.65	0.65	246

TABLE 6.3 CLASSIFICATION REPORT FOR DANCE MOVE PREDICTION.

”waltz_left” shows the lowest F1-score of 0.55, suggesting difficulties in distinguishing this class from visually similar moves. The macro averages (precision, recall, and F1-score 0.62) are slightly lower than the weighted averages, indicating that performance is better on classes with larger sample sizes, such as ”jack_step” and ”touch_step_left,” while underperforming on less frequent ones.

Overall, the statistical

6.4 Summary

This chapter presented the implementation results and evaluation of the proposed leg movement recognition and quality assessment system. The leg landmark detection successfully identified key anatomical points on the lower extremities, forming the foundation for motion tracking and gait analysis. A diverse training dataset ensured model robustness across various poses and lighting conditions, supporting accurate movement classification during real-time operation.

Model evaluation demonstrated that the system achieved an overall accuracy of 65% for movement classification and 84.63% for quality assessment. Confusion matrices confirmed the model's ability to distinguish between most dance movements, with misclassifications occurring primarily among visually similar patterns. Statistical analysis showed balanced performance across classes, though accuracy was higher for frequently represented movements such as $touch_s tepand$

REFERENCES

El Raheb, K., Stergiou, M., Katifori, A., and Ioannidis, Y. (2019). Dance interactive learning systems: A study on interaction workflow and teaching approaches. ACM Computing Surveys*, 52:1–37.*

Kim, J.-W., Choi, J.-Y., Ha, E.-J., and Choi, J.-H. (2023). Human pose estimation using mediapipe pose and optimization method based on a humanoid model. Applied Sciences*, 13(4).*

Lei, P., Li, N., and Liu, H. (2023). Dance movement recognition based on gesture.

Lin, J.-H. (2015). Just dance: The effects of exergame feedback and controller use on physical activity and psychological outcomes. Games for Health Journal*, 4(3):183–189. PMID: 26182062.*

Oudah, M., Al-Naji, A. A., and Chahl, J. (2020). Hand gesture recognition based on computer vision: A review of techniques. Journal of Imaging*, 6:73.*

Protopapadakis, E., Voulodimos, A., Doulamis, A., Camarinopoulos, S., Doulamis, N., and Miaoulis, G. (2018). Dance pose identification from motion capture data: A comparison of classifiers. Technologies*, 6.*

Rallis, I., Protopapadakis, E., Voulodimos, A., Doulamis, N., Doulamis, A., and Bardis, G. (2019). Choreographic pattern analysis from heterogeneous motion capture systems using dynamic time warping. Technologies*, 7(3):56.*

Tharatipyakul, A., Srikaewsiew, T., and Pongnumkul, S. (2024). Deep learning-based human body pose estimation in providing feedback for physical movement: A review. Heliyon*, 10(17):e36589.*

Tölgyessy, M., Dekan, M., and Chovanec, v. (2021). Skeleton tracking accuracy and precision evaluation of kinect v1, kinect v2, and the azure kinect. Applied Sciences*, 11(12):5756.*

Venkatrayappa, D., Tremeau, A., Muselet, D., and Colantoni, P. (2024). Survey of 3d human body pose and shape estimation methods for contemporary dance applications.

Yu, X. and Xiong, S. (2019). A dynamic time warping based algorithm to evaluate kinect-enabled home-based physical rehabilitation exercises for older people. Sensors*, 19(13):2882. PubMedID: 31261746.*

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., and Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. CoRR.

Zhao, H., Du, B., Jia, Y., and Zhao, H. (2025). Danceformer: Hybrid transformer model for real-time dance pose estimation and feedback. Alexandria Engineering Journal*, 121:66–76.*

Produced: November 11, 2025, 21:49

Appendix A
MEMBER SKILLSET IDENTIFICATION

TABLE A.1 TEAM MEMBERS' PROGRAMMING SKILLS

Member	**Model Dev.**	**UI Design**	**Source Control (GitHub)**	**Problem Solving & Opt.**	**Python**
Hans	Intermediate	Novice	Expert	Intermediate	Intermediate
Gerald	Intermediate	Basic	Novice	Intermediate	Intermediate
Nathan	Intermediate	Novice	Novice	Intermediate	Intermediate

Appendix B
WORK BREAKDOWN STRUCTURECAPSTONE PROJECT ON OPERATIONAL TECHNOLOGIES

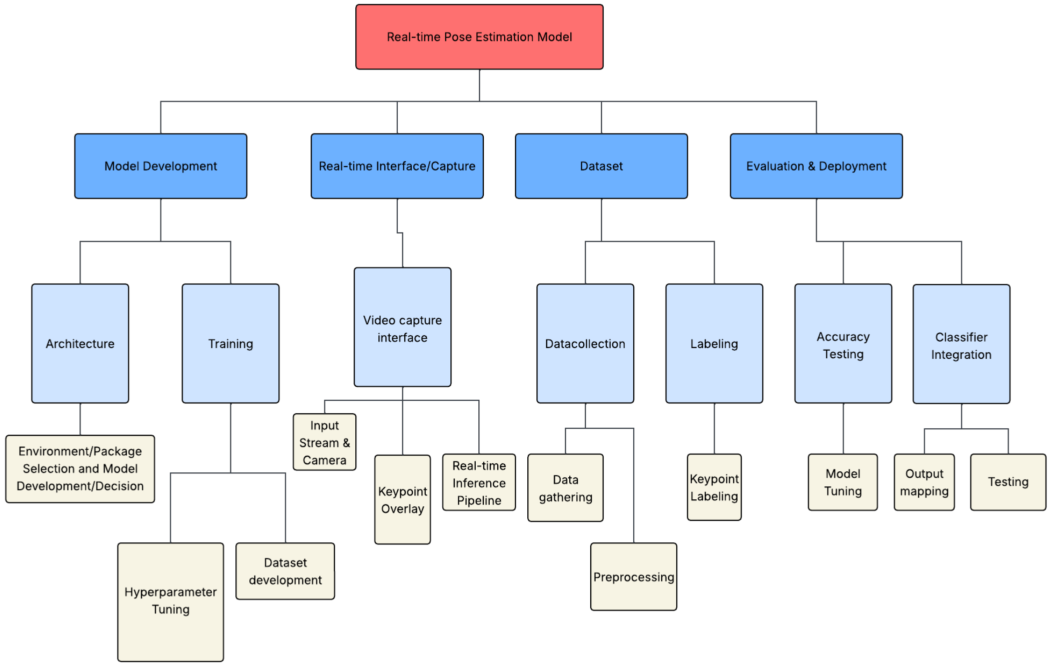


Fig. B.1 Work Breakdown Structure for Hans Capstone Project on Operational Technologies

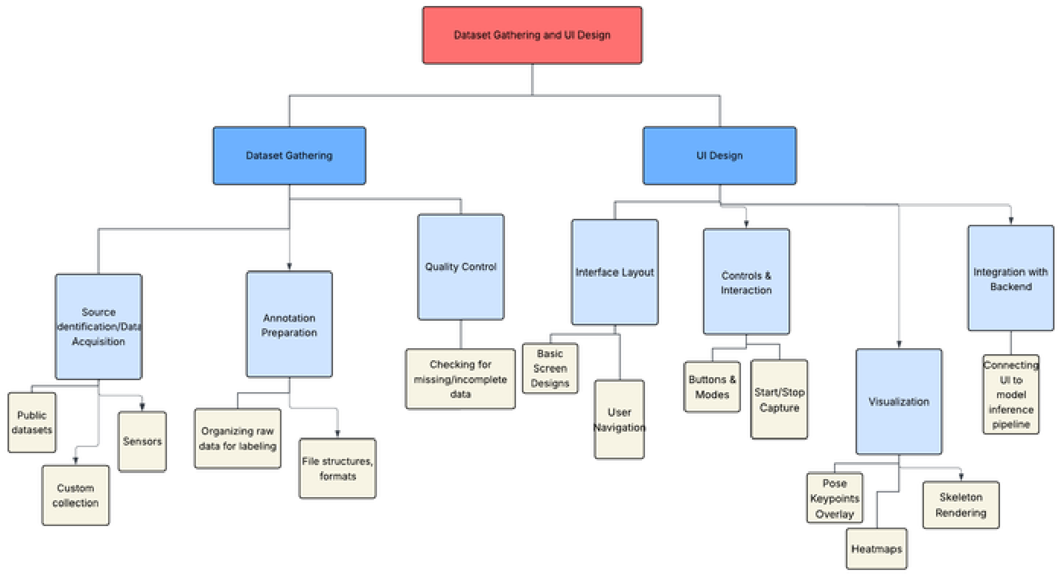


Fig. B.2 Work Breakdown Structure for Nathan Capstone Project on Operational Technologies

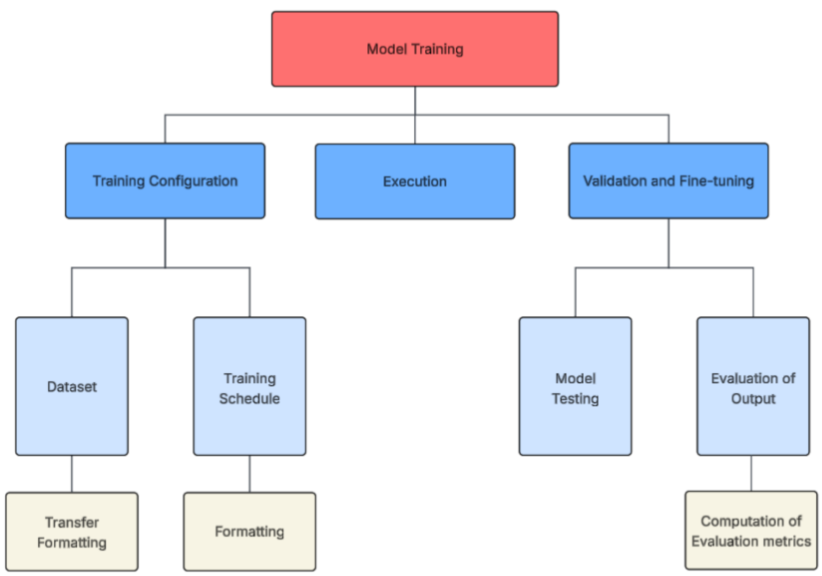
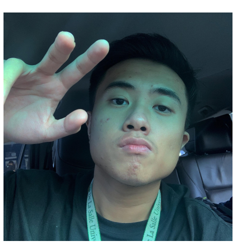


Fig. B.3 Work Breakdown Structure for Gerald Capstone Project on Operational Technologies

Appendix C
VITA

 *Nathan Raekel L. Calaguian is a BS CPE student from De La Salle University Manila.*

 *Gerald Antonio P. Ellar is a BS CPE student from De La Salle University Manila.*

 *Hans Jamee Mahait is a BS CPE student from De La Salle University Manila.*