

**GAIT RECOGNITION AS A BIOMETRIC SIGNAL, INVESTIGATING
SKELETAL TRAJECTORY EXTRACTION AND CLASSIFICATION
APPROACHES**

AKOL JOSEPH

REG NO: BU/GS23/MCF/2

STUDENT NUMBER: 2300402696

**A RESEARCH REPORT SUBMITTED TO THE DIRECTORATE OF GRADUATE
STUDIES, RESEARCH AND INNOVATIONS IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF A MASTER'S DEGREE IN COMPUTER
FORENSICS OF BUSITEMA UNIVERSITY**

May, 2025

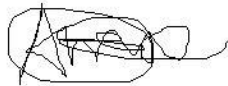
DECLARATION

I, AKOL JOSEPH, hereby declare that this research report is my original work, except where appropriate acknowledgments have been provided. I further declare that this work has not been submitted to any other university or institution of higher learning for any academic award.

Name: AKOL JOSEPH

Registration Number: BU/GS23/MCF/2

Student Number: 2300402696

A handwritten signature in black ink, appearing to be 'AKOL JOSEPH', enclosed within a hand-drawn oval border.

Signature:

SUPERVISORS' APPROVAL

This research report titled “*Gait recognition as a biometric signal, investigating skeletal trajectory extraction and classification approaches*” is submitted for examination with the approval of the following academic supervisor(s);

Signature:



Date: 16/07/2025

Dr. Owomugisha Godliver (PhD)

Department of Computer Engineering and Informatics

Faculty of Engineering

DEDICATION

I dedicate this research first and foremost to my Lord and Savior, Jesus Christ, whose guidance and grace have sustained me through every season of this journey.

To my parents, Mrs. Amongin Lucy and Mr. Opio Basil, I extend my deepest gratitude for their unwavering support, encouragement, and belief in my potential.

I also dedicate this work to my loving wife, Immaculate Atyang, and my siblings, whose patience and understanding carried me through the demanding phases of this study. Their sacrifices and steadfast presence provided the strength I needed to persevere.

Special recognition goes to my dear friend, Mr. Omollo Moses Tanga, whose enduring support, encouragement, and kindness served as a steadfast foundation during challenging times. This achievement is a testament to the invaluable role each of you has played in my journey, and for that, I remain profoundly grateful.

ACKNOWLEDGEMENT

I extend my deepest gratitude to Almighty God for granting me the wisdom, clarity, and perseverance to undertake this endeavor. Without His guidance, I would be nothing.

I sincerely thank my supervisors, Dr. Owomugisha Godliver and Dr. Estafania, for their steadfast support and invaluable guidance, which were essential to the completion of this work.

My heartfelt appreciation also goes to Dr. Odongtoo Godfrey, Dr. Matovu Davis, and Professor Ocen Gilbert for their significant contributions in helping me maintain focus and determination throughout this process.

To my cherished classmates and friends, Mr. Bageya Ivan, Kisangala Gerald, Saleh and all others who have supported me, even if not named here, I am profoundly grateful. May the Lord, who sees our hearts, bless you abundantly.

TABLE OF CONTENTS

DECLARATION	i
SUPERVISORS' APPROVAL	ii
DEDICATION	iii
ACKNOWLEDGEMENT	iv
LIST OF TABLES.....	ix
LIST OF FIGURES	ix
ABBREVIATIONS	xi
ABSTRACT.....	xii
CHAPTER ONE: INTRODUCTION	1
1. Background of the study	1
1.1 Statistical background of the study	3
1.2 Problem statement	5
1.3 Study objectives.....	6
1.3.1 General objective	6
1.3.2 Specific objectives	6
1.4 Research questions.....	6
1.5 Significance of the study.....	6
1.6 Justification of the study	6
1.7 Scope.....	7
1.7.1 Content scope.....	7
1.7.2 Time scope	7
CHAPTER TWO: LITERATURE REVIEW	8
2. Literature Review	8
2.1 History and Definition of Gait Recognition.....	8
2.2 Gait Recognition Process in Computer Forensics Context	9
2.2.1 Data Acquisition.....	9
2.2.2 Preprocessing	9
2.2.3 Feature Extraction.....	10
2.2.4 Comparison.....	11
2.2.5 Evaluation.....	11
2.3 Forensic Implications and Challenges.....	12

2.4 Frameworks Used in Gait Recognition	12
2.4.1 Appearance-Based Framework.....	12
2.4.2 Model-Based Framework	12
2.4.3 Spatiotemporal Framework	13
2.4.4 End-to-End Deep Learning Framework	13
2.5 Models Used in Gait Recognition	14
2.5.1 Convolutional Neural Networks (CNNs).....	14
2.5.2 Long Short-Term Memory (LSTM)	14
2.5.3 Graph Neural Networks (GNNs)	14
2.5.4 Support Vector Machines (SVM)	15
2.5.5 Hidden Markov Models (HMM)	15
2.6 Mitigation Techniques in Gait Recognition	15
2.6.1 Data Augmentation	15
2.6.2 Feature Fusion.....	16
2.6.3 View-Invariant Models	16
2.6.4 Noise Reduction.....	17
2.6.5 Transfer Learning.....	17
CHAPTER THREE: METHODOLOGY	18
3. Introduction	18
3.1 Dataset acquisition	19
3.2 Camera information	20
3.3 Dataset description.....	20
3.4 Keypoint Extraction.....	21
3.4.1 2D Model: OpenPose method.....	21
3.4.2 3D Model: MediaPipe Pose method	22
3.5 Visualizing Skeleton Frames.....	22
3.6 Data Preprocessing.....	25
3.6.1 Loading JSON Files.....	25
3.6.2 DataFrame Structure	25
3.7 Sequence Preparation for LSTM	26
3.7.1 2D Model	26
3.7.2 3D Model	26

3.8 Model Training.....	27
CHAPTER FOUR: PRESENTATION, INTERPRETATION AND ANALYSIS OF THE RESULTS.....	28
4. Introduction	28
4.1 2D model performance evaluation	28
4.1.1 Overall performance	28
4.1.2 Per participant metrics performance	30
4.1.3 Testing the 2D model with new data.....	31
4.2 3D Model performance evaluation	34
4.2.1 Key Performance Metrics	34
4.2.2 Classification Report.....	36
4.2.3 Aggregate Metrics.....	37
4.2.4 Testing the 3D model with new data.....	39
4.3 General Observations	44
4.3.1 Model Bias	44
4.3.2 Confidence vs. Accuracy:	44
4.3.3 Impact of Thresholds:	44
4.3.4 Robustness to Challenging Conditions:	44
4.4 Comparison of 2D and 3D Model Performance.....	45
4.4.1 Unseen sequences	45
4.4.2 Occlusion conditions.....	46
4.4.3 Single frame predictions	47
4.4.4 Conclusion	48
CHAPTER FIVE: DISCUSSION, CONCLUSION AND RECOMMENDATIONS	50
5. Introduction	50
5.1 Discussions	50
5.2 Specific Objectives and Research Questions	50
5.2.1. Specific Objective: To identify a robust gait dataset recorded across various environmental conditions with varying camera angles.....	50
5.2.2. Specific Objective: Extract 2D and 3D skeletal keypoints data from the dataset using OpenPose and MediaPipe.....	51
5.2.3. Specific Objective: Develop and evaluate the accuracy of the 2D and 3D models for gait recognition.....	51

5.2.4. Specific Objective: Analyze and compare the performance of the two models under varying environmental conditions.	52
5.3 Recommendations.....	53
REFERENCES.....	54
APPENDICES	59

LIST OF TABLES

Table 1: A tabular comparison of existing gait recognition models from notable papers.....	8
Table 2: Shows a summary of the information of the participants in the dataset.	19
Table 3: DataFrame	21
Table 4: DataFrame Structure	25
Table 5: 2D model architecture	27
Table 6: 3D model architecture	27
Table 7: shows a summary of the overall performance of the 2D model	29
Table 8: Below provides the individual performance details for each of the 10 participants..	30
Table 9: Classification Report for Participant Classes.....	36
Table 10: Testing the 3D model with new data results across different scenarios.....	39
Table 11: Comparison of 2D and 3D model performance	45
Table 12: This table shows detailed information of the participants in the dataset	59
Table 13: Showing frames for all participants.....	64

LIST OF FIGURES

Figure 1:Flow chart diagram.....	19
Figure 2: Visualization of original frame vs skeleton frame at angle 45 degrees	24
Figure 3 Visualization of original frame vs skeleton frame at angle 90 degrees	24
Figure 4: Training and Validation loss plot for 2D model	29
Figure 5: 2D model confusion matrix	31
Figure 6: Training and Validation loss of 3D model	35
Figure 7: 3D model Training and Validation accuracy	36
Figure 8: 3D mode confusion matrix	38
Figure 9: Confusion matrix of 3D model under unseen sequences	41
Figure 10: Confusion matrix of 3D model under occlusion	42
Figure 11: CONfusion matrix of 3D model with Single-Frame predictions	43
Figure 12: Histogram showing the comparison of 2D vs 3D models under unseen sequences	46
Figure 13:Histogram showing the comparison of 2D vs 3D models under occlusion conditions.....	47
Figure 14:Histogram showing the comparison of 2D vs 3D models with single-frame predictions.....	48
Figure 15:Histogram showing the comparison of 2D vs 3D models with their confidence scores across various scenarios	49
Figure 16: Screenshot of json files extracted for participant 9 in google colab	60
Figure 17: Dataframe for 2D model training	60
Figure 18: Function to visualize skeleton from a folder	61
Figure 19: Extracting skeleton data for all participants	62
Figure 20: prepare lstm for 10 participants 2d	63

Figure 21: Original vs skeleton frame	64
Figure 22: Skeleton frames of participant walking at 90 degrees(camera),front,45 degrees and Rear view respectively.....	65

ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
GNN	Graph Neural Network
SVM	Support Vector Machine
HMM	Hidden Markov Model
GEI	Gait Energy Image
RNN	Recurrent Neural Network
FPR	False Positive Rate
ROC-AUC	Receiver Operating Characteristic - Area Under Curve
JSON	JavaScript Object Notation
WE	White European
BE	Black European
AR	Arab
AS	Asian
AF	African
fps	Frames per Second
CVPR	Conference on Computer Vision and Pattern Recognition
ICPR	International Conference on Pattern Recognition
ICCV	International Conference on Computer Vision
IEEE	Institute of Electrical and Electronics Engineers
SC	Soft Computing (Springer Journal)
PR	Pattern Recognition (Journal)
GAN	Generative Adversarial Network
PyTorch	Python Torch (Deep Learning Framework)
Kinect	Microsoft's motion-sensing input device
ICPC	International Conference on Pattern Classification
OULP	Osaka University Large Population (Gait Dataset)
CASIA-B	Chinese Academy of Sciences Institute of Automation Gait Database B
GAID	Gait from Audio, Image and Depth (TUM Dataset)
MP	MediaPipe
BGR	Blue-Green-Red (color channel order used in OpenCV)
RGB	Red-Green-Blue (standard color model)
cv2	OpenCV (Computer Vision library in Python)
np	NumPy (Python library abbreviation)
CNN-LSTM	Hybrid of Convolutional Neural Network and Long Short-Term Memory
GEV	Gait Energy Volume
EER	Equal Error Rate
GPU	Graphics Processing Unit
U-Net	A convolutional network architecture for image segmentation

ABSTRACT

Gait recognition, a non-intrusive biometric method effective at a distance, serves as a practical alternative to traditional identification techniques like fingerprints or facial recognition, particularly in forensic and surveillance contexts. This study aims to design and compare 2D and 3D models for gait recognition with the general objective of analyzing their comparative performance. Specific objectives include: (1) identifying a robust gait dataset recorded across various environmental conditions and camera angles, (2) extracting 2D and 3D skeletal keypoints using OpenPose and MediaPipe, respectively, (3) developing and evaluating the accuracy of 2D and 3D gait recognition models, and (4) analyzing their performance under varying environmental conditions. The research addresses the effectiveness of dataset utilization, comparative model accuracy, and key performance influencers. Results show the 2D model excels in occlusion handling, achieving a 0% false positive rate (FPR) for unknown instances, while the 3D model better generalizes to unseen sequences at higher thresholds, with a 74.55% FPR. However, the 2D model struggles with single-frame performance (80% FPR), and the 3D model exhibits significant bias and poor occlusion handling (100% FPR). Recommendations include enhancing dataset diversity, incorporating temporal features via LSTM layers, and improving 3D keypoint robustness through occlusion augmentation. This study provides a replicable framework, offering insights into model trade-offs and advancing gait recognition for resource constrained environments, forensic applications, and academic research.

CHAPTER ONE: INTRODUCTION

1. Background of the study

Gait recognition has emerged as a transformative biometric identification technique, leveraging the unique walking patterns of individuals to enable non-intrusive identification Munusamy et al.[1]. Unlike traditional biometric methods such as fingerprint or iris recognition, which require close proximity or direct subject cooperation, gait recognition can be performed at a distance. This makes it particularly valuable for applications in surveillance, security, and forensic investigations Bouchrika and Imed[2]. The non-invasive nature of gait recognition allows it to be applied in scenarios where subjects are unaware of being monitored, such as in public spaces or high-security environments.

However, despite its potential, gait recognition systems face significant challenges in real-world settings. Occlusions caused by clothing, carried objects, or environmental obstructions can obscure critical gait features and degrade system performance Shen et al.[3]. Additionally, variations in camera angles, lighting conditions, and walking surfaces introduce further complexities. These factors hinder the extraction of reliable and consistent gait features, limiting the effectiveness of gait recognition in uncontrolled environments Lee et al.[4].

The development of gait recognition systems has been driven by advancements in computer vision and machine learning, particularly deep learning techniques. These enable the extraction of complex patterns from video data Alireza et al.[5]. Typically, systems rely on keypoint data derived from pose estimation algorithms that track the movement of body joints to construct a gait signature. Yet, the effectiveness of these systems is often limited by the availability of diverse and comprehensive training datasets. Many existing datasets are constrained in size and variability, restricting the ability of models to generalize across different populations, walking styles, and environmental conditions Sarkar et al.[6].

This issue becomes more pronounced in uncontrolled environments, where factors such as varying viewpoints, occlusions, and dynamic backgrounds significantly impact recognition accuracy Al-Azawi et al.[7]. For instance, Fabbri et al.[8] demonstrated that recognition accuracy can decrease by up to 25% when subjects carry objects that partially occlude their lower limbs a common scenario in real-world surveillance footage.

A critical aspect of gait recognition research is the comparison between 2D and 3D approaches, each offering distinct advantages and limitations. 2D gait recognition methods rely on video sequences captured by standard cameras and are computationally efficient and widely accessible. This makes them suitable for real-time applications Lee et al.[4]. These methods have achieved high accuracies, with some studies reporting up to 90% on benchmark datasets like the CASIA Gait Database Song et al.[9]. However, 2D methods are sensitive to perspective distortions and lack depth information, which can reduce accuracy under varying viewpoints or occlusions.

In contrast, 3D gait recognition approaches utilize depth sensors or multi-camera setups to capture volumetric data. These offer greater robustness to viewpoint changes and occlusions [14]. For example, Su et al.[10] reported that 3D models achieved an average identification accuracy of 85% across multiple viewpoints, compared to 75% for 2D models under similar conditions. Despite these advantages, 3D methods require more complex hardware and higher

computational resources, with processing times often exceeding 2–3 seconds per sequence, compared to under 1 second for 2D methods Jebara et al.[11]. These trade-offs remain underexplored, particularly regarding their performance on small datasets and their suitability for real-world deployment.

Another critical challenge in gait recognition is the ability to reject unknown individuals, a requirement especially important in forensic and security applications. False positives in these contexts can lead to erroneous identifications with significant legal or security implications as seen by Bouchrika and Imed[2]. Current systems often focus on closed-set identification, where all individuals in the test set are known to the model. However, open-set scenarios where the system must distinguish between known and unknown subjects are less studied by Geng et al.[12]. Leng et al.[13] reported that many existing gait recognition systems have false positive rates (FPRs) ranging from 5% to 15% in open-set scenarios, which is unacceptable for forensic applications. This gap highlights the need for robust models that can handle both identification and rejection tasks effectively.

Recent advancements in deep learning and pose estimation have opened new avenues for addressing these challenges. Techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown promise in extracting temporal and spatial features from gait sequences, thereby improving robustness to variations in position, scale, and environmental conditions Batabyal et al.[14]. For instance, CNN-based models have achieved up to 92% accuracy on the TUM GAID dataset, which includes audio and visual gait data under varying conditions Shen et al.[3].

Additionally, the integration of transfer learning and data augmentation techniques can mitigate the limitations of small datasets. These strategies enable models to generalize better across diverse scenarios. The shift from silhouette-based to skeleton-based representations, facilitated by advancements in pose estimation techniques like OpenPose and AlphaPose, has further enhanced performance. Skeleton-based methods have increased in popularity, accounting for 24% of studies post-2020 by Nerella et al.[15].

By systematically comparing 2D and 3D gait recognition models and exploring their performance in open-set scenarios, this study aims to contribute to the development of more reliable and scalable gait recognition systems for security and law enforcement applications.

The significance of gait recognition extends beyond traditional security applications. In forensic investigations, gait analysis can provide critical evidence for identifying suspects in video footage where other biometric cues, such as facial features, are unavailable or obscured. For instance, Zeno et al.[16] demonstrated the use of gait analysis in forensic medicine, highlighting its potential in identifying individuals from video footage in criminal investigations.

Similarly, in smart city initiatives, gait recognition can enhance public safety by enabling real-time monitoring of crowded areas without requiring intrusive measures Sun et al.[17]. However, the practical deployment of these systems requires addressing the aforementioned challenges, including robustness to environmental variations, computational efficiency, and the ability to handle diverse populations. Ethical concerns, such as privacy violations from recording gait in public spaces, also warrant careful consideration.

This study seeks to bridge these gaps by developing and evaluating gait recognition models using keypoint data, with a focus on enhancing robustness and scalability.

1.1 Statistical background of the study

The field of gait recognition has seen significant advancements in recent years, particularly with the application of deep learning techniques. A comprehensive review by Younesi et al.[18] indicates that convolutional neural networks (CNNs) have been the most commonly used architecture, accounting for 45% of studies between 2015 and 2022. Other architectures include 3D convolutional neural networks (3DCNNs) (8%), Generative Adversarial Networks (GANs) (6%), Long Short-Term Memory networks (LSTMs) (3%), and Graph Neural Networks (GNNs) (11%, with most studies conducted post-2020). Additionally, various hybrid models have emerged, such as CNN+LSTM combinations, which account for 9% of the research by Younesi et al.[18].

Performance metrics have shown steady improvement across several benchmark datasets. On the CASIA-B dataset, the best reported accuracy was 84.2% in 2019. This increased to 90.4% in 2020, 98.34% in 2021, and an impressive 99.93% in 2022 with the ViTGait model. A similar trend was observed on the OU-MVLP dataset, where accuracy rose from 89.18% in 2020 to 98.00% in 2021, and slightly declined to 96.15% in 2022.

There has also been a notable shift in the type of body representations used in gait recognition. Until 2020, silhouette-based methods dominated the field, accounting for approximately 70% of studies. However, with the advancement of pose estimation technologies like OpenPose and AlphaPose, skeleton-based methods have become more popular. These now account for 24% of studies post-2020, while 7% of studies incorporate both silhouette and skeleton-based approaches by Younesi et al.[18].

Despite these advancements, several challenges continue to impede progress. Covariate factors such as walking speed, footwear, carrying conditions, clothing, lighting, camera viewpoints, and walking surfaces significantly affect recognition accuracy. For instance, Davidoff et al.[19] found that recognition accuracy decreased by up to 25% when subjects carried objects that partially occluded their lower limbs.

Another significant limitation is the size and diversity of available gait datasets. The CASIA Gait Database, one of the largest datasets, includes only 124 subjects, which is insufficient for training models that can generalize to real-world variability by Aman et al.[20]. This limited diversity contributes to overfitting, with test accuracies often 15–25% lower than training accuracies Montesinos et al.[21].

In terms of computational efficiency, 2D gait recognition methods are generally faster, with processing times typically under 1 second per sequence. By contrast, 3D methods require more computational resources, often taking 2–3 seconds or more per sequence by Glaser et al.[22]. This discrepancy is especially critical for real-time applications where low latency is essential.

A key challenge in real-world deployments is reducing false positive rates (FPRs) in open-set recognition scenarios. Existing systems often exhibit FPRs between 5% and 15%, which is unacceptably high for forensic or high-stakes security applications according to Nwafor et al.[23]. Encouragingly, recent research suggests that anomaly detection techniques can reduce FPRs to below 2%.

The integration of deep learning continues to open new possibilities for handling complex scenarios. Models like GaitSet and GaitPart have demonstrated improved performance in cross-view and temporal modeling, achieving accuracies of 87.90% and 88.70%, respectively. However, significant challenges remain, including occlusion, susceptibility to spoofing attacks, and ethical concerns related to surveillance and privacy in public spaces by Younesi et al.[18].

This study aims to address these limitations by developing robust 2D and 3D gait recognition models. It will leverage keypoint data and advanced deep learning techniques to enhance accuracy, computational efficiency, and scalability ensuring practical applicability in real-world security and forensic contexts.

1.2 Problem statement

Gait recognition is gaining attention as a non-intrusive biometric method that enables identification at a distance, without requiring subject cooperation. This makes it especially valuable in surveillance and security applications Tiwari et al.[24]. However, despite its potential, real-world deployment remains challenging. Factors such as occlusions from clothing or carried objects, varying lighting conditions, and shifting camera angles often degrade system performance. Additionally, current models struggle to generalize due to limited and non-diverse training data.

There is also a lack of comprehensive comparison between 2D and 3D gait recognition approaches. While 3D methods offer improved robustness to viewpoint changes, they typically require more computational resources and sophisticated sensor setups. In contrast, 2D methods are more efficient but suffer from limitations like perspective distortions and the absence of depth information Carpendale et al.[25]. These trade-offs raise concerns about practical deployment, particularly in real-time or resource-constrained environments.

This study aims to address these gaps by developing and evaluating both 2D and 3D gait recognition models using keypoint data extracted from video sequences. It focuses on improving robustness to positional and scale variations and assessing each model's ability to reject unknown individuals an essential feature for forensic use. By leveraging recent advances in deep learning and pose estimation, this research seeks to contribute to the development of more accurate, scalable, and operational gait recognition systems Topham et al.[26].

1.3 Study objectives

1.3.1 General objective

This research aims to design 2D and 3D models for gait recognition and analyze their comparative performance.

1.3.2 Specific objectives

1. To identify a robust gait dataset that was recorded across various environmental conditions with varying camera angles.
2. To extract the 2D and 3D skeletal keypoints data from the dataset using OpenPose and MediaPipe respectively.
3. To develop and evaluate the accuracy of the 2D and 3D models for gait recognition.
4. To analyze and compare the performance of the two models under varying environmental conditions.

1.4 Research questions

1. How effectively can a robust gait dataset, recorded across various environmental conditions and camera angles, be identified and utilized for extracting 2D and 3D skeletal keypoints using OpenPose and MediaPipe?
2. What is the comparative accuracy of 2D and 3D models developed for gait recognition when evaluated using skeletal keypoints extracted from a diverse gait dataset?
3. How do the 2D and 3D gait recognition models perform under varying environmental conditions, and what are the key factors influencing their performance differences?

1.5 Significance of the study

1. A primary contribution of this work is the collection and use of a compact, custom gait dataset. This dataset is tailored to reflect real-world conditions and constraints, offering a valuable resource for future research and helping address the current limitations in data diversity and scale in gait recognition studies.
2. By conducting a systematic comparison between 2D and 3D gait recognition approaches, the study provides insights into their relative performance, robustness, and computational efficiency. This includes evaluating lightweight pose estimation tools like OpenPose and MediaPipe, making the findings particularly relevant for low-resource environments such as small-scale security systems and academic research settings.
3. The study demonstrates the real-world relevance of gait recognition by integrating a rejection mechanism for unknown individuals a critical requirement in forensic applications. This enhances the reliability of gait-based identification in surveillance footage, especially where traditional biometric methods are unavailable or ineffective.

1.6 Justification of the study

This study is justified by its potential to contribute new knowledge through the development and evaluation of lightweight 2D and 3D gait recognition models. By focusing on practical implementation using limited data and computational resources, it addresses real-world constraints often overlooked in existing literature.

1.7 Scope

1.7.1 Content scope

The study focuses on evaluating 2D and 3D gait recognition using keypoint data from 65 participants through deep learning techniques.

1.7.2 Time scope

The information to be collected covered a period of 3 months.

CHAPTER TWO: LITERATURE REVIEW

2. Literature Review

This chapter provides a comprehensive exploration of gait recognition, focusing on its evolution, methodologies, and applications, with a particular emphasis on comparing 2D and 3D models. It begins with a historical overview and definition of gait recognition, tracing its development from early biomechanical studies to modern biometric systems, followed by a comparative analysis of existing models.

2.1 History and Definition of Gait Recognition

In understanding Gait recognition as a biometric technique that identifies individuals based on their unique walking patterns, leveraging both behavioral and physical characteristics. Its history traces back to early observations of human locomotion, with scientific foundations laid by Frampton and Michael[27], where he studied animal movement qualitatively. Modern gait analysis emerged in the late 19th century with pioneers like Baker and Richard[28], who used serial photography to capture motion sequences (1870s), and Klette et al.[29], who quantified gait biomechanics (1890s). The advent of computer vision in the 20th century shifted gait recognition toward automated systems, with significant developments in the 1990s when researchers like Sivapalan and Sabesan[30] in 1998 explored silhouette-based recognition. Today, gait recognition is a key area in biometrics, driven by deep learning and 3D modeling, offering non-invasive identification at a distance.

Table 1: A tabular comparison of existing gait recognition models from notable papers.

Paper Title	Framework	Model Comparison	Inclusive	Excludes
Gait Recognition Using CNN (2017)	Model-free, CNN-based	2D silhouette CNN vs. traditional GEI	Silhouette data, multi-view angles	3D data, skeletal models
3D Gait Analysis with Kinect (2015)	Model-based, 3D CNN	3D CNN vs. 2D appearance-based methods	Depth data, joint trajectories	Low-resolution 2D video
Cross-View Gait Recognition (2019)	Spatiotemporal, LSTM	LSTM vs. SVM on multi-view gait	Temporal sequences, view invariance	Static images, single-view data
Gait Energy Image Recognition(2004)	Appearance-based, SVM	GEI + SVM vs. manual feature extraction	Silhouette averages, simple setup	Dynamic 3D models, complex scenes
Graph-Based Gait Recognition (2020)	Graph Neural Network (GNN)	GNN vs. CNN-LSTM hybrids	3D skeletal graphs, spatial relations	Appearance-only data, 2D silhouettes
Deep Gait Recognition (2018)	End-to-end CNN	3D CNN vs. 2D CNN on large datasets	Large-scale 3D data, viewpoint changes	Small datasets, manual features

This table reflects the evolution from simple 2D appearance-based methods to complex 3D and deep learning approaches, highlighting the diversity in frameworks and data requirements.

2.2 Gait Recognition Process in Computer Forensics Context

Gait recognition, as a biometric identification technique, analyzes an individual's walking pattern to establish identity, offering a unique advantage in computer forensics where traditional identifiers like fingerprints or facial features may be unavailable or obscured. In forensic investigations, gait recognition is increasingly applied to video evidence such as CCTV footage from crime scenes to link suspects to events, complementing digital forensics (e.g., computer logs, mobile data) with physical behavioral evidence Aung et al.[31]. The process involves several stages: data acquisition, preprocessing, feature extraction, comparison, and evaluation. Each stage is tailored to forensic needs, balancing accuracy, robustness, and legal admissibility. Below, these stages are explored in depth, with their technical underpinnings, forensic applications, challenges, and relevance to 2D and 3D modalities.

2.2.1 Data Acquisition

The gait recognition process begins with acquiring video or sensor data capturing a suspect's walking pattern. In computer forensics, this typically involves low-quality CCTV footage from public spaces (e.g., streets, banks, stores) where a crime occurred, often supplemented by higher-quality sources like body-worn cameras or seized mobile device recordings Peterson et al.[32]. For 2D gait recognition, monocular video from a single camera suffices, providing silhouettes or raw pixel data. This aligns with project "Gait Recognition Using 2D Silhouettes and CNN-LSTM Hybrid" by Burges et al.[33], where CASIA-B-like footage could simulate forensic video. In contrast, 3D gait recognition, as in project "3D Gait Recognition Using 3D CNNs and Kinect Data" by Bari et al.[34], requires depth sensors or multi-camera setups to capture three-dimensional motion, though such setups are rare in forensic contexts due to cost and availability.

Forensic data acquisition faces unique constraints: footage may be partial (e.g., a few seconds of walking), captured at odd angles (e.g., overhead or oblique), or degraded by environmental factors (e.g., rain, darkness). Metadata, such as timestamps or camera specifications, is often extracted alongside video to contextualize the evidence within a digital forensic timeline. The challenge lies in working with uncontrolled, opportunistic data rather than controlled lab conditions, necessitating robust preprocessing to extract usable gait information.

2.2.2 Preprocessing

Preprocessing refines raw forensic video into a form suitable for analysis, addressing noise, resolution, and segmentation issues inherent in real-world footage. For 2D gait recognition, this stage starts with frame extraction, converting video into a sequence of images, followed by background subtraction to isolate the walking subject. Techniques like Gaussian Mixture Models (GMM) or deep learning-based segmentation (e.g., Mask R-CNN) remove static backgrounds (e.g., walls, cars), producing clean silhouettes, as could enhance project "2D Gait

Recognition with Gait Energy Image (GEI) and SVM” by Loureiro et al.[35]. Video enhancement adjusting contrast, sharpening edges, or interpolating frames compensates for low resolution or frame rates (e.g., 15 fps vs. 60 fps in lab datasets).

In 3D gait recognition, preprocessing is more complex. If depth data is unavailable (common in forensics), 2D video from multiple angles for example, synchronized CCTV cameras can be reconstructed into 3D using multi-view geometry or structure-from-motion techniques, as in project "Comparative Study of 2D and 3D Pose Estimation for Gait Recognition" by Menychtas et al.[36]. Alternatively, pose estimation tools like OpenPose extract 2D joint positions, which are triangulated into 3D coordinates if camera parameters are known. Noise reduction, such as temporal smoothing with Kalman filters, stabilizes jittery joint trajectories caused by low-quality sensors or occlusions.

Forensic preprocessing must handle occlusions (e.g., crowds, vehicles), variable lighting, and incomplete gait cycles (e.g., half a stride). This stage is critical for admissibility, as courts require evidence integrity any manipulation must be documented and reversible. The challenge is balancing enhancement with authenticity, ensuring processed data reflects the original scene without introducing artifacts.

2.2.3 Feature Extraction

Feature extraction identifies distinctive gait characteristics from preprocessed data, translating visual or spatial information into quantifiable metrics. In 2D forensic gait recognition, appearance-based features dominate due to data simplicity. The Gait Energy Image (GEI), used in project by Loureiro et al.[35], averages silhouettes over a gait cycle into a single grayscale image, capturing stride shape and posture. Alternatively, spatiotemporal features, as in Project [33], extract frame-by-frame silhouette dynamics (e.g., contour changes) or optical flow to model motion direction and speed. These features are compact and computationally efficient, suitable for the often-limited forensic footage.

For 3D gait recognition, model-based features leverage skeletal data. Project "DeepGait: A 3D Gait Recognition Approach Using Deep Learning” by Wolf et al.[37] might extract joint angles (e.g., knee flexion), distances between keypoints (e.g., hip-to-ankle), or velocities from 3D trajectories. Project "3D Gait Recognition with Graph Neural Networks (GNNs)" by Weng et al.[38] uses graph-based features, encoding spatial relationships (e.g., knee-to-hip connectivity) and temporal evolution. Deep learning models like 3D CNNs bypass manual extraction, learning hierarchical features directly from depth sequences or voxelized data (e.g., Gait Energy Volume).

In forensics, feature extraction must prioritize robustness to covariates clothing, bags, or injuries that alter gait appearance. The challenge is extracting sufficient discriminatory power from short, noisy sequences, often requiring hybrid approaches (e.g., combining GEI with joint dynamics). Features must also be interpretable for court presentation, complicating the use of opaque deep learning outputs.

2.2.4 Comparison

The comparison stage matches extracted gait features against a reference database of known individual suspects, victims, or a broader population to identify or exclude a match. In forensic practice, this database might be limited (e.g., suspects in a specific case) or hypothetical (e.g., comparing against a lab dataset like OULP). Traditional methods, like that used by Han et al.[39] SVM, compute distances (e.g., Euclidean) between feature vectors, classifying the closest match. Probabilistic models, such as Hidden Markov Models (HMM) by Eddy and Sean[40], estimate likelihoods of state sequences matching a reference gait, though they're less common today.

Deep learning enhances comparison precision. Burges et al.[33] CNN-LSTM outputs a feature embedding, compared via cosine similarity or Euclidean distance to reference embedding. Menychtas et al.[36] LSTM uses a twin network to compute similarity scores between two gait sequences, ideal for one-to-one suspect verification. In 3D contexts, Zou et al.[41] CNN or Mao et al.[42] GNN might generate high-dimensional embedding, leveraging neural network capacity to discern subtle differences.

Forensic comparison must account for uncertainty partial data or covariate changes reduce confidence. Likelihood ratios (e.g., "10 times more likely this gait matches Suspect A than a random person") are preferred over binary decisions, aligning with forensic standards like the Page et al.[43] criteria. The challenge is ensuring the reference database reflects real-world variability, a hurdle when relying on controlled datasets.

2.2.5 Evaluation

Evaluation assesses the reliability of the gait match, producing results admissible in legal proceedings. In computer forensics, this involves quantitative metrics like accuracy, Equal Error Rate (EER), or Receiver Operating Characteristic (ROC) curves, alongside qualitative justification for court. For 2D methods (e.g., Project [33]), accuracy might drop with viewpoint shifts, requiring error bounds (e.g., $70\% \pm 10\%$). 3D methods typically yield higher confidence due to richer data, but forensic footage rarely supports such precision.

Forensic evaluation emphasizes specificity and sensitivity false positives (wrongly identifying an innocent person) are costlier than false negatives (missing a suspect). Statistical tools like Monte Carlo simulations estimate match reliability under noise or occlusion, as might be applied to Bari et al.[34] Kinect data. Expert testimony often accompanies results, explaining limitations (e.g., "Gait altered by injury reduces certainty") and contextualizing scores within the case narrative.

The primary challenge is legal admissibility. Unlike fingerprints, gait recognition lacks standardized error rates or universal acceptance, as critiqued in forensic reviews Bouchrika and Imed[44]. Courts demand transparency black-box deep learning models must be paired with explainable outputs (e.g., feature importance maps). Variability in gait due to age, mood, or footwear further complicates reliability, necessitating cautious interpretation.

2.3 Forensic Implications and Challenges

In computer forensics, gait recognition complements digital evidence (e.g., IP traces) by linking physical presence to virtual activity. For example, identifying a suspect’s gait in bank robbery footage could corroborate timestamped ATM hacks. Its non-invasive nature requiring no subject cooperation makes it ideal for covert analysis, unlike DNA or iris scans. Projects like Menychtas et al.[36], comparing 2D and 3D, highlight 3D’s superior accuracy (20-55% improvement), but forensic reliance on 2D CCTV limits its use.

Challenges include data quality (low-resolution, short clips), ethical concerns (privacy of bystanders), and legal scrutiny. The UK’s Forensic Science Regulator (2021) by Doyle and Sean[45] notes gait analysis’s “emerging” status, urging validation against larger, diverse datasets. Mitigation like noise reduction or view-invariant modeling helps, but standardization remains elusive. Future forensic gait recognition may integrate multi-modal biometrics (e.g., gait + voice) for higher reliability.

2.4 Frameworks Used in Gait Recognition

Gait recognition systems rely on distinct frameworks that dictate how data is processed, features are extracted, and recognition is performed. These frameworks vary in complexity, data requirements, and applicability to 2D or 3D modalities. Below, each framework is described in detail, focusing on its methodology, strengths, and limitations.

2.4.1 Appearance-Based Framework

The appearance-based framework focuses on extracting gait features directly from visual data, typically 2D silhouettes or images, without explicitly modeling the underlying skeletal or biomechanical structure. This approach, rooted in early gait recognition research for example Han and Bhanu, 2006[39], averages gait sequences into compact representations like the Gait Energy Image (GEI). In this process, video frames are segmented to isolate the walking subject, silhouettes are extracted using background subtraction techniques (e.g., Gaussian Mixture Models), and these silhouettes are averaged over a gait cycle to form a single 2D image. Classification is then performed using traditional machine learning methods like Support Vector Machines (SVM) or, more recently, Convolutional Neural Networks (CNNs).

This framework’s simplicity is its primary strength. It requires only monocular video, making it cost-effective and widely applicable in surveillance scenarios where high-resolution or multi-camera setups are impractical. For instance, Han et al.[39], GEI is paired with SVM to achieve lightweight recognition, suitable for resource-constrained environments. However, its reliance on appearance makes it vulnerable to viewpoint changes, occlusions, and covariate factors like clothing or carrying conditions. The lack of temporal dynamics further limits its discriminative power compared to spatiotemporal or 3D approaches, as it compresses time-series data into a static representation.

2.4.2 Model-Based Framework

The model-based framework explicitly models the human body’s structure and motion, typically using 3D skeletal data or biomechanical parameters (e.g., joint angles, stride length). Originating from motion capture and animation studies, this approach gained traction with the

availability of depth sensors like Microsoft Kinect and multi-camera systems. Data is collected as 3D joint coordinates or trajectories, often reconstructed from 2D images using techniques like multi-view geometry or pose estimation tools for example, OpenPose followed by triangulation Sun et al.[46]. Features are then extracted based on kinematic properties, such as joint velocities or angular displacements, and fed into recognition models.

This framework excels in robustness, as seen by Zou et al.[41] and Bari et al.[34], where 3D skeletal data enhances viewpoint invariance and accuracy. By capturing the spatial arrangement and movement of body parts, it mitigates issues like occlusion and clothing variation that plague appearance-based methods. However, its complexity is a significant drawback requiring expensive hardware (e.g., depth sensors, multiple cameras) and sophisticated preprocessing (e.g., 3D reconstruction). This makes it less practical for uncontrolled environments, and its performance heavily depends on the quality and completeness of the 3D data.

2.4.3 Spatiotemporal Framework

The spatiotemporal framework bridges spatial and temporal dimensions, treating gait as a dynamic sequence rather than a static image or isolated model. It emerged as computational power increased, enabling the integration of spatial feature extraction (e.g., via CNNs) with temporal modeling e.g. via Long Short-Term Memory networks, LSTMs by Lai et al.[47]. In this approach, each frame of a gait sequence is processed to extract spatial features (e.g., silhouette shapes or joint positions), which are then sequenced over time to capture motion patterns. Burges et al.[33] exemplifies this by using a CNN to analyze individual 2D frames and an LSTM to model their temporal evolution.

This framework’s strength lies in its ability to capture both “what” (spatial structure) and “how” (temporal dynamics) of gait, making it more discriminative than appearance-based methods and less data-intensive than full 3D modeling. It adapts well to both 2D and 3D data, as seen by Menychtas et al.[36], where Siamese LSTM processes 3D joint sequences. However, it requires substantial training data to avoid overfitting, and the sequential nature of LSTMs can slow training compared to fully convolutional approaches. Additionally, its performance may degrade with noisy or incomplete sequences.

2.4.4 End-to-End Deep Learning Framework

The end-to-end deep learning framework integrates feature extraction and classification into a single neural network, eliminating the need for manual feature engineering. Popularized by advances in deep learning by Aloysius et al.[48], it uses architectures like 3D CNNs or Transformers to process raw gait data whether 2D video or 3D point clouds directly into recognition outputs. In Zou et al.[41] project, a 3D CNN processes skeletal sequences from the OULP dataset, learning hierarchical features without predefined steps like silhouette averaging or joint tracking.

This framework’s primary advantage is its adaptability and performance, especially with large datasets, where it can uncover complex patterns missed by traditional methods. It excels in 3D gait recognition, leveraging depth and temporal dimensions for high accuracy and viewpoint invariance. However, it demands significant computational resources (e.g., GPUs) and large,

labeled datasets for training, making it impractical for small-scale or real-time applications. Overfitting is also a risk without proper regularization, as seen in its dependency on high-quality data.

2.5 Models Used in Gait Recognition

Gait recognition employs a variety of machine learning and deep learning models, each suited to specific frameworks and data types. Below, the key models are explored in detail, with their mechanisms, applications, and trade-offs.

2.5.1 Convolutional Neural Networks (CNNs)

CNNs are the backbone of modern gait recognition, designed to extract spatial features from image or volumetric data. In 2D gait recognition, CNNs process silhouettes or GEIs, using convolutional layers to detect edges, shapes, and textures, followed by pooling layers to reduce dimensionality and fully connected layers for classification. In 3D contexts, as in Bari et al.[34], 3D CNNs extend this by applying 3D convolutions to depth sequences or voxelized data (e.g., Gait Energy Volume), capturing spatial-temporal relationships in a single pass.

CNNs are highly effective due to their hierarchical feature learning, making them versatile across frameworks. They outperform traditional methods like SVM on raw data and scale well with dataset size. However, 2D CNNs struggle with temporal modeling unless paired with recurrent networks, while 3D CNNs are computationally expensive, requiring substantial memory and processing power e.g. for Bari et al.[34] Kinect data.

2.5.2 Long Short-Term Memory (LSTM)

LSTMs, a type of Recurrent Neural Network (RNN), excel at modeling sequential data, making them ideal for gait's temporal dynamics. They use memory cells and gates (input, forget, output) to retain long-term dependencies, addressing the vanishing gradient problem of vanilla RNNs. Burges et al.[33], indicated that an LSTM follows a CNN to process frame-by-frame features over a gait cycle, while Menychtas et al.[36] uses a Siamese LSTM to compare 3D joint sequences for similarity.

LSTMs shine in spatiotemporal frameworks, capturing subtle motion patterns like stride timing or arm swing. Their bidirectional variant can even analyze sequences forward and backward, enhancing robustness. However, training is slow due to sequential processing, and they require large datasets to generalize, as overfitting is a concern with limited gait cycles (e.g., Project [33]'s sensitivity to clothing changes).

2.5.3 Graph Neural Networks (GNNs)

GNNs are a recent innovation, treating gait data as graphs where joints are nodes and bones are edges. In Mao et al.[42] research, a GNN processes 3D skeletal data, using message-passing to aggregate spatial relationships (e.g., how the knee influences the hip) and a temporal layer to track changes over time. This approach leverages the inherent structure of human motion, unlike CNNs or LSTMs, which treat data as grids or sequences.

GNNs offer superior spatial modeling, making them robust to noise and occlusions in 3D gait recognition. They bridge 2D and 3D by starting with 2D pose estimates (e.g., OpenPose) and

reconstructing 3D graphs. However, their complexity requires specialized frameworks like PyTorch Geometric, and computational costs are high due to graph operations, limiting scalability.

2.5.4 Support Vector Machines (SVM)

SVMs are traditional classifiers that map features (e.g., GEI pixel values) into a high-dimensional space, finding an optimal hyperplane to separate classes. Han et al.[39], SVM classifies GEI features, relying on a kernel (e.g., radial basis function) to handle non-linear relationships. They were prevalent in early gait recognition due to their simplicity and effectiveness with small datasets.

SVMs are lightweight and fast to train, ideal for appearance-based frameworks with precomputed features. However, they lack the capacity to model complex spatiotemporal patterns, underperforming deep learning models on raw or 3D data. Their reliance on manual feature extraction also limits adaptability.

2.5.5 Hidden Markov Models (HMM)

HMMs model gait as a probabilistic sequence of hidden states (e.g., stance, swing phases), with observable outputs (e.g., joint angles). Though less common in modern deep learning-dominated research, they were foundational in early model-based gait recognition, estimating state transitions from temporal data.

HMMs are interpretable and effective with limited data, but their assumptions (e.g., Markov property) oversimplify gait's complexity. They've largely been supplanted by LSTMs and CNNs for higher accuracy and flexibility.

2.6 Mitigation Techniques in Gait Recognition

Gait recognition, while promising as a biometric identifier, faces numerous challenges such as viewpoint variations, occlusions, covariate factors (e.g. clothing, speed), noise, and limited dataset sizes. To overcome these, researchers employ mitigation techniques that enhance model robustness, accuracy, and generalizability. Below, are five key mitigation strategies data augmentation, feature fusion, view-invariant modeling, noise reduction, and transfer learning are explored in detail, with their technical underpinnings, practical implementations, and implications for 2D and 3D gait recognition systems.

2.6.1 Data Augmentation

Data augmentation artificially expands training datasets by generating synthetic variations of existing gait data, addressing issues like limited sample sizes and overfitting. In gait recognition, where collecting diverse, large-scale datasets (e.g., thousands of subjects across multiple conditions) is resource-intensive, augmentation is critical. For 2D gait recognition, techniques include flipping silhouettes horizontally to simulate opposite viewpoints, rotating images to mimic angular changes, and altering brightness or contrast to account for lighting variations. For example, in Burges et al.[33], flipping and scaling silhouettes from the CASIA-B dataset could improve robustness to viewpoint shifts.

In 3D gait recognition, augmentation extends to spatial transformations, such as rotating 3D skeletal models around axes (e.g., yaw, pitch) or perturbing joint coordinates with Gaussian noise to simulate sensor inaccuracies. Menychtas et al.[36] might apply such techniques to 3D joint sequences, generating synthetic walks at varying speeds or with altered stride lengths. Advanced methods like Generative Adversarial Networks (GANs) can synthesize entirely new gait cycles, as explored in recent works like Salcedo and Edwin[49], producing realistic 2D silhouettes or 3D point clouds from learned distributions.

The advantage of data augmentation lies in its ability to enhance model generalization without additional data collection, making it cost-effective and scalable. However, excessive augmentation risks introducing unrealistic samples (e.g., unnatural joint angles), degrading performance. Balancing augmentation intensity and realism requires careful validation, often guided by domain knowledge of human biomechanics.

2.6.2 Feature Fusion

Feature fusion combines multiple types of gait features spatial, temporal, kinematic, or appearance-based to create a more robust representation, mitigating the limitations of relying on a single feature type. In 2D gait recognition, this might involve fusing static GEI features (spatial) with dynamic frame-by-frame silhouette changes (temporal), as could be applied in Han et al.[39] project to overcome its lack of temporal detail. Fusion methods include concatenation, where feature vectors are combined into a larger input for classification, or weighted averaging, where contributions are balanced based on their discriminative power.

For 3D gait recognition, feature fusion leverages the richer data available, such as combining joint angles (kinematic) with depth-based volumetric features (spatial-temporal). In Bari et al.[34] project, fusing Gait Energy Volume (GEV) with joint trajectory velocities could enhance robustness to partial occlusions. Deep learning facilitates advanced fusion via multi-stream networks, where separate branches (e.g., CNN for spatial, LSTM for temporal) process different features, merging outputs at a later layer. Menychtas et al.[36] Siamese LSTM could adopt this by integrating 2D pose features with 3D reconstructed data.

Feature fusion's strength is its ability to capture complementary information, improving accuracy and resilience to covariate factors like clothing or viewpoint changes. However, it increases computational complexity and requires careful feature selection to avoid redundancy or noise amplification. Tuning fusion weights or architectures often demands extensive experimentation.

2.6.3 View-Invariant Models

View-invariant modeling aims to mitigate the sensitivity of gait recognition to viewpoint changes, a major challenge in both 2D and 3D contexts. In 2D systems, where silhouette appearance varies drastically with angle (e.g., frontal vs. side view), techniques include training models on multi-view datasets (e.g., CASIA-B) or applying geometric transformations to normalize data to a canonical view. Zou et al.[41] leverages 3D data's inherent viewpoint robustness, but 2D methods might use view synthesis projecting silhouettes into a common plane via homography or learned mappings.

In 3D gait recognition, view invariance is achieved by modeling gait in a 3D coordinate system, as Mao et al.[42], where GNNs process skeletal graphs unaffected by camera angle. Alternatively, deep learning models like 3D CNNs or Transformers can learn view-invariant features directly from multi-angle training data, encoding gait patterns independent of perspective. Another approach is manifold learning, projecting high-dimensional gait data into a lower-dimensional space where view variations are minimized, though this is less common in recent deep learning-dominated research.

The primary advantage is improved performance in unconstrained environments, crucial for real-world applications like surveillance. However, view-invariant models often require diverse training data or complex architectures, increasing computational demands. For 2D systems, full invariance remains elusive without 3D information, limiting effectiveness compared to 3D methods.

2.6.4 Noise Reduction

Noise reduction techniques preprocess gait data to filter out extraneous elements background clutter, sensor artifacts, or lighting inconsistencies enhancing feature quality and model performance. In 2D gait recognition, background subtraction is a staple, using algorithms like Gaussian Mixture Models (GMM) or deep learning-based segmentation (e.g., U-Net) to isolate silhouettes from complex scenes. Han et al.[39] reliance on GEI could benefit from such preprocessing to mitigate silhouette degradation in noisy CCTV footage.

For 3D gait recognition, noise reduction addresses depth sensor inaccuracies (e.g., Kinect’s limited resolution) or incomplete reconstructions. Techniques include temporal smoothing (e.g., Kalman filtering) to stabilize joint trajectories, as might be applied in Project [42]’s GNN pipeline, or outlier removal to discard erratic 3D points caused by occlusions. Denoising autoencoders, a deep learning approach, can also reconstruct clean gait sequences from noisy inputs, potentially enhancing Bari et al.[34] Kinect-based data.

Noise reduction improves signal-to-noise ratio, boosting recognition accuracy and robustness, especially in uncontrolled settings. However, aggressive filtering risks losing subtle gait details (e.g., minor arm swings), and real-time implementation can be computationally intensive. Selecting appropriate thresholds or models requires balancing noise removal with data preservation.

2.6.5 Transfer Learning

Transfer learning mitigates the challenge of small gait datasets by leveraging pre-trained models from related tasks, fine-tuning them for gait recognition. In 2D contexts, models pre-trained on large image datasets (e.g., ImageNet) like ResNet or VGG can be adapted to process silhouettes or GEIs, as could enhance Burges et al.[33] CNN-LSTM hybrid. The pre-trained convolutional base extracts general visual features (e.g., edges, shapes), which are then fine-tuned with gait-specific layers for classification.

In 3D gait recognition, transfer learning might use models pre-trained on action recognition datasets (e.g., Kinetics, NTU RGB+D), adapting their spatiotemporal understanding to gait tasks. Zou et al.[41] 3D CNN could initialize weights from such a model, reducing training

time and data needs. Techniques include freezing early layers (preserving generic features) while retraining later layers on gait data, or domain adaptation to align source and target distributions (e.g., 3D action to 3D gait).

The advantage is accelerated training and improved performance with limited data, making it ideal for gait recognition's often sparse datasets. However, domain mismatch (e.g., ImageNet's static images vs. gait's dynamic sequences) can limit effectiveness, requiring careful fine-tuning. Computational resources for initial pre-training may also be a barrier for smaller research setups.

CHAPTER THREE: METHODOLOGY

3. Introduction

This chapter comprises the methodology and approaches that supported the development of the 2D and 3D models for gait recognition and performance comparison. The chapter discusses the research methods that were followed to achieve the objectives of this study.

Information about the exiting models and methods was collected using literature reviews of previous works, and quantitative and qualitative methods. This information was collected and proper conclusions were drawn, leading to the development of both the 2D and 3D models for gait recognition and their performance analysis comparison.

Flowchart

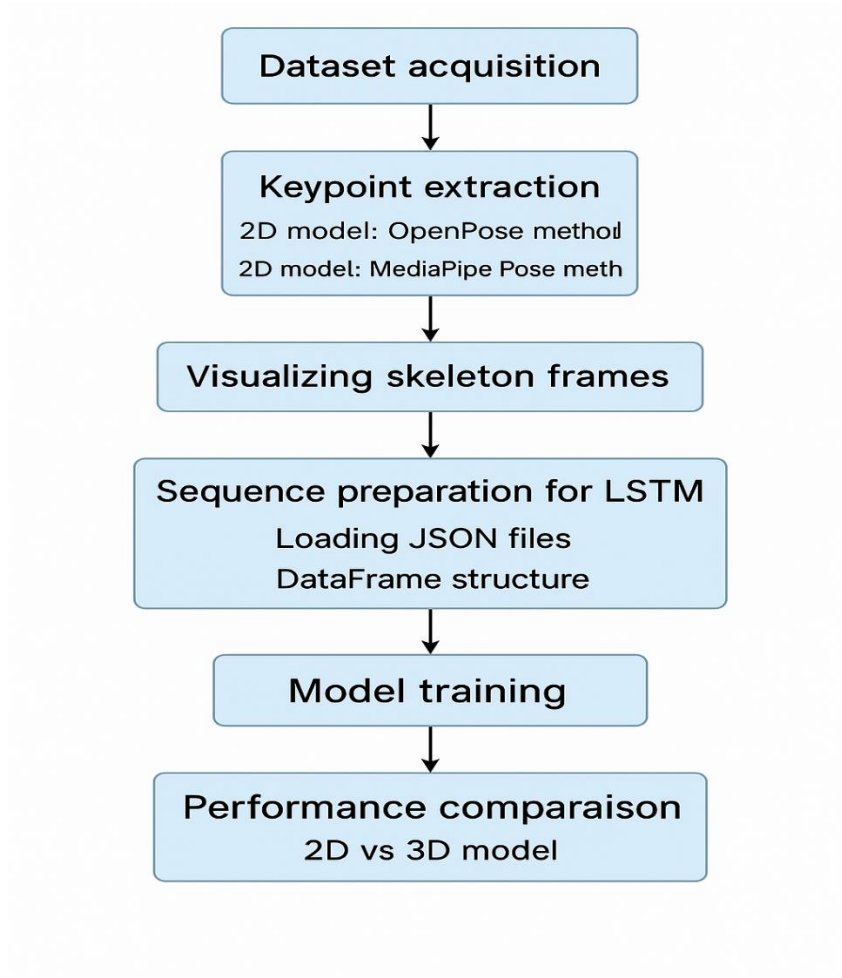


Figure 1:Flow chart diagram

3.1 Dataset acquisition

This research acquired a gait dataset containing 65 diverse participants with both indoor and outdoor environments. The data was acquired using 2 digital cameras.

Each participant provided 24 walking sequences from a range of viewing angles and each participant also provided an alternative outfit to provide diversity in personal appearance.

Table 2: Shows a summary of the information of the participants in the dataset.

Participants	Gender	Age	Height(cm)	Mass(kg)	Ethnicity
65	M: 37 (57.81%)	35.14 (± 12.44)	170.83 (± 10.86)	81.51 (± 20.18)	WE: 62.5%
	F: 27 (42.19%)	Max: 65 Min: 18	Max: 192 Min: 137.16	Max: 133.36 Min: 51	BE, AR, AS, AF: 37.5%

The dataset included 65 participants, comprising of 57.81% males and 42.19% females, with an average age of 35.14 years (ranging from 18 to 65), an average height of 170.83 cm (137.16–192 cm), and an average mass of 81.51 kg (51–133.36 kg); ethnically, 62.5% were White European, while the remaining 37.5% included Black European, Arab, Asian, and African participants.

3.2 Camera information

Each walking sequence was recorded with two Kodak AZ422 cameras (20MP) positioned at different locations and angles, with adjustments made to achieve eight distinct viewing angles for each participant.

3.3 Dataset description

The dataset consisted of video recordings of 65 participants walking past a camera, capturing their walking patterns for classification purposes. The data was collected in a controlled setting to ensure consistency across recordings and was organized hierarchically as follows:

1. Participants: 65 participants, labelled as Participant_1 to Participant_65 as folders.
2. Experiments: Each participant performed 3 experiments, labelled as Experiment_1 to Experiment_3 as subfolders, to capture variations in walking under different conditions.
3. Tasks: Each experiment included 6 tasks, labelled E1T1 to E3T6 (e.g., E1T1 indicates Experiment 1, Task 1), representing different walking scenarios.
4. Angles: Each task was recorded from multiple camera angles, including 45°, 90°, and rear views, to provide diverse perspectives of the walking patterns.

This structure resulted in a total of 513,734 frames across all participants, experiments, tasks, and angles, as confirmed by the DataFrame information after keypoint extraction. Each video captured a participant walking past the camera, with the initial frames often being empty (no person in the frame) until the participant entered the camera's field of view. The videos varied in length, and the number of frames with detected keypoints differed across participants and tasks due to variations in walking speed and video duration.

After extracting the 2D keypoints using OpenPose, the dataset was loaded into a Pandas DataFrame with 513,734 entries, containing the following columns:

1. Participant: The participant ID (e.g., "Participant_1").
2. Experiment: The experiment ID (e.g., "Experiment_1").
3. Task: The task ID (e.g., "E1T1").
4. JSON_Folder: The folder containing the JSON files (e.g., "P1_E1T1_45_json").
5. Frame: The JSON file name (e.g., "P1_E1T1_45_000000000015_keypoints.json").
6. Keypoints: A list of 25 sublists, each containing [x, y] coordinates for a keypoint.
7. Target: The participant number (1 to 65), used as the label for classification.

Due to computational constraints, a subset of the dataset was used for model training, consisting of the first 10 participants (Participant_1 to Participant_10). This subset contained 79,470 frames, with the following distribution of frames per participant:

Table 3: DataFrame

Participant Number	Number of frames
1	7,943
2	8,593
3	7,491
4	8,547
5	7,453
6	10,129
7	8,165
8	8,286
9	8,517
10	4,346
Total	79,470

The target labels (1 to 10) matched the participant numbers, with a mean target value of 5.33, a standard deviation of 2.74, and a range from 1 to 10, as confirmed by the descriptive statistics of the DataFrame.

3.4 Keypoint Extraction

Keypoint extraction was performed to convert the video data into sequences of keypoints representing the participants' walking patterns. The 2D model used OpenPose, while the 3D model used MediaPipe Pose, with distinct methods for each.

3.4.1 2D Model: OpenPose method

OpenPose, a real-time multi-person keypoint detection library, was used to extract 2D keypoints for the 2D model. The BODY_25 model, which detects 25 body keypoints, was selected for its comprehensive coverage of the human body.

Video Processing

Each video was processed frame by frame using OpenCV, a computer vision library, to read the frames. OpenPose was applied to each frame to detect the body keypoints. A detection confidence threshold of 0.5 was set to ensure reliable detection, filtering out low-confidence keypoints.

Keypoint Extraction

For each frame where a person was detected, OpenPose output 25 keypoints, each with 3 values: x, y coordinates, and a confidence score. The confidence score was discarded, retaining only the x and y coordinates for 2D analysis. Frames with fewer than 25 keypoints were excluded to ensure data consistency. Additionally, frames where the confidence of the right hip (index 8) or left hip (index 11) was zero were skipped, as these keypoints were critical for normalization in later steps.

Handling Empty Frames

At the start of each video, the "people" list in the JSON output was often empty ("people": []) because the participant had not yet entered the frame. These empty frames were skipped during extraction to include only frames with detected keypoints.

Storage

The keypoints were saved as JSON files in a structured directory:

*Data/Participant_X/Experiment_Y/EY TZ_Angle/PX_EY TZ_Angle_json/PX_EY TZ_Angle_NN
NNN_keypoints.json*

where X is the participant ID, Y is the experiment ID, Z is the task ID, Angle is the camera angle (e.g., 45, 90, or Rear), and NNNNN is the frame number.

3.4.2 3D Model: MediaPipe Pose method

MediaPipe Pose, a machine learning solution developed by Google, was used to extract 3D keypoints for the 3D model. It detects 33 pose landmarks on the human body, providing x, y, and z coordinates.

Video Processing

Videos were processed frame by frame using OpenCV to read the frames. MediaPipe Pose was applied to each frame to detect the pose landmarks. The minimum detection confidence and minimum tracking confidence parameters were set to 0.5 to ensure reliable detection and tracking of keypoints across frames.

Keypoint Extraction

For each frame where a person was detected, MediaPipe Pose output 33 keypoints, each with 4 values: x, y, z coordinates, and a visibility score. The visibility score was discarded, retaining only the x, y, and z coordinates for 3D analysis. Frames with fewer than 33 keypoints were excluded to maintain data consistency.

Handling Empty Frames

The initial frames of each video often had an empty "people" list in the JSON output ("people": []). These empty frames were skipped to include only frames with detected keypoints.

Storage

The keypoints were saved in the same structured directory format as the 2D model, ensuring consistency in data organization.

3.5 Visualizing Skeleton Frames

To verify the quality of the extracted keypoints, a visualization function was implemented to overlay the skeleton points on the corresponding video frames for the 2D model.

Function Definition

The function in figure 18 was developed to visualize skeleton points from JSON files on corresponding video frames. It took three parameters: the path to the video file, the path to the folder containing per-frame JSON files, and the number of frames to visualize.

Video Processing

The video was opened using OpenCV, and the total number of frames was determined. Ten evenly spaced frames were selected using a linear spacing function (`np.linspace`) to ensure a representative sample across the video.

Frame Processing

For each selected frame, the video was advanced to the corresponding frame number, and the frame was read and converted from BGR to RGB color space for visualization with Matplotlib.

Keypoint Overlay

The corresponding JSON file for each frame was loaded, and the keypoints (x, y coordinates) were extracted. Green circles were drawn on the frame at each keypoint location using OpenCV's `cv2.circle` function, with a radius of 5 pixels for visibility.

Visualization

The frames with overlaid keypoints were displayed in a single row using Matplotlib, with each subplot titled by its frame number. The axes were turned off for clarity, and the layout was adjusted for readability.

The function was applied to a sample video and JSON folder from Participant 1 (e.g., `P1_E1T1_45_skeleton.avi` and `P1_E1T1_45_json`) for camera angle 45^0 and 90^0 as shown below, visualizing 4 frames to confirm the accuracy of the keypoint extraction.

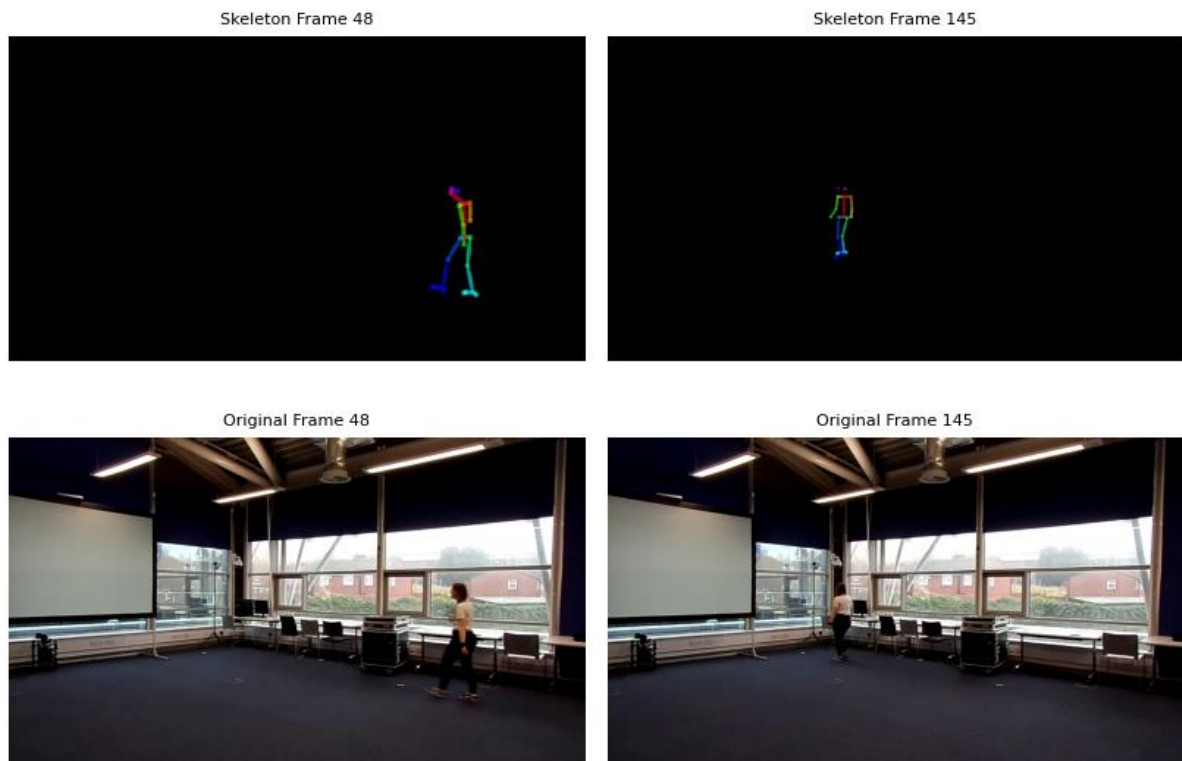


Figure 2: Visualization of original frame vs skeleton frame at angle 45 degrees

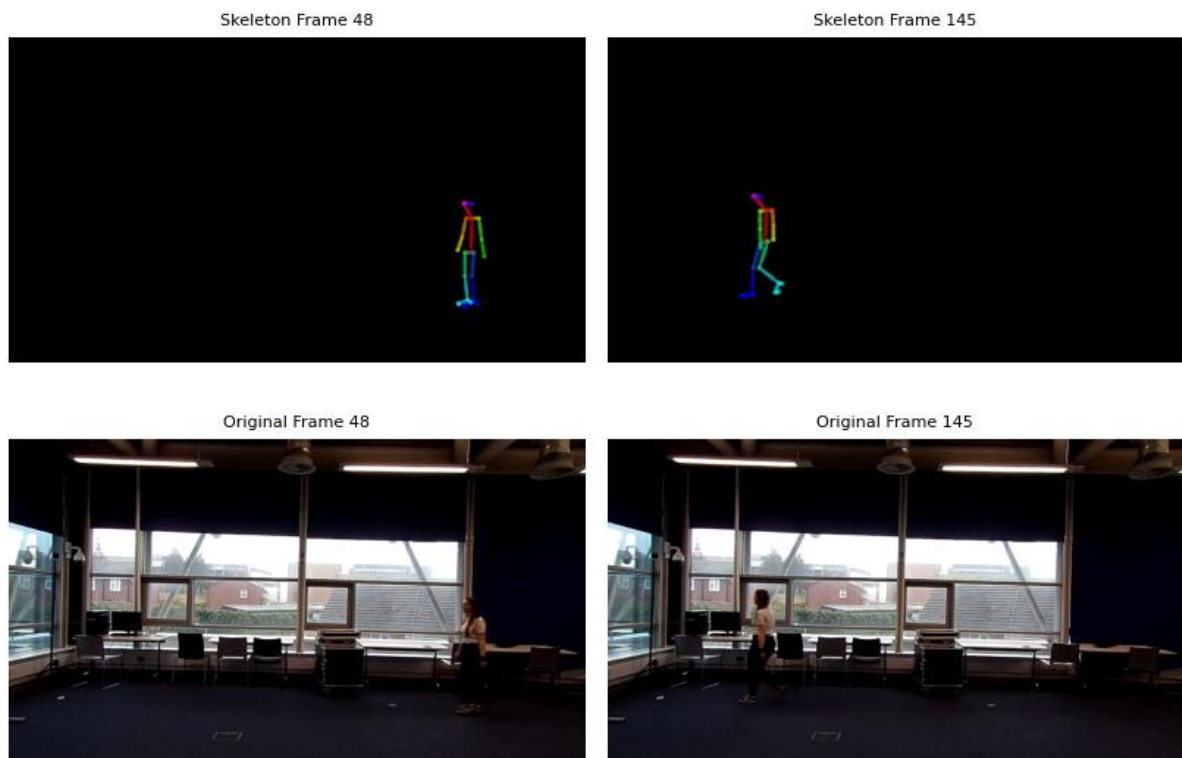


Figure 3 Visualization of original frame vs skeleton frame at angle 90 degrees

3.6 Data Preprocessing

The raw keypoint data from both models was processed into a structured format suitable for machine learning, with adjustments to account for the dimensionality of the keypoints.

3.6.1 Loading JSON Files

A custom function in figure 19 was developed to load the JSON files into a Pandas DataFrame for the 2D model. A similar approach was used for the 3D model.

2D Model: The function iterated over all participants, experiments, tasks, and angles, loading each JSON file and extracting the keypoints. Empty frames (where the "people" list was empty) were skipped, and frames with fewer than 25 keypoints or zero confidence for the right hip (index 8) or left hip (index 11) were excluded to ensure data quality.

3D Model: The function followed a similar process, skipping empty frames and excluding frames with fewer than 33 keypoints to maintain consistency.

3.6.2 DataFrame Structure

The resulting DataFrame for the 2D model contained 513,734 entries with the following columns:

Table 4: DataFrame Structure

Participant	The participant ID (e.g., "Participant_1").
Experiment	The experiment ID (e.g., "Experiment_1").
Task	The task ID (e.g., "E1T1").
JSON_Folder	The folder containing the JSON files (e.g., "P1_E1T1_45_json").
Frame	The JSON file name (e.g., "P1_E1T1_45_000000000015_keypoints.json").
Keypoints	A list of 25 sublists, each containing [x, y] coordinates for a keypoint.
Target	The participant number (1 to 65), used as the label for classification.

For the 3D model, the DataFrame had a similar structure, but the Keypoints column contained 33 sublists, each with [x, y, z] coordinates.

Sorting by Participant Number

For the 2D model, a new column, Participant_Number, was created by extracting the numeric part from the Participant column (e.g., "Participant_1" → 1). The DataFrame was sorted by Participant_Number in ascending order to ensure a consistent order for processing, resulting in df_sorted.

Filtering for 10 Participants

Due to RAM limitations, the dataset was filtered to include only the first 10 participants (Participant_1 to Participant_10) for both models. For the 2D model, this resulted in df_10 with 79,470 frames, as described in the dataset section. The 3D model followed a similar filtering process, resulting in a comparable number of frames for the same participants.

3.7 Sequence Preparation for LSTM

The keypoint data was converted into sequences of frames for LSTM training, with separate functions for the 2D and 3D models to account for the differences in keypoint dimensionality.

3.7.1 2D Model

The function in figure 20 was used to prepare the data for the 2D model.

Filtering and Sorting: The DataFrame was filtered to include only the first 10 participants and sorted by Participant, Experiment, Task, and Frame to ensure temporal ordering.

Normalization: Keypoints were normalized by subtracting the coordinates of the middle hip, calculated as the average of the right hip (index 8) and left hip (index 11) in the OpenPose BODY_25 model. This made the keypoints position-invariant, focusing the model on relative movements.

Sequence Creation: A sliding window approach was used to create sequences of 50 frames (sequence_length=50). For each participant, sequences were generated by taking 50 consecutive frames, with the label being the Target of the last frame in the sequence. Frames with an invalid number of keypoints (not equal to 25) were skipped.

Feature Extraction: Each frame's keypoints (25 keypoints $\times$ 2 coordinates = 50 features) were flattened into a 1D array. Each sequence had the shape (50, 50) (50 frames, 50 features per frame). This resulted in 78,980 sequences (63,184 training, 15,796 test).

Label Encoding: The Target labels (1 to 10) were converted to zero-based indices (0 to 9) and one-hot encoded for 10-class classification.

Train-Test Split: The data was split into training (80%) and test (20%) sets using train_test_split with stratification to preserve the class distribution, ensuring fair representation of each participant.

3.7.2 3D Model

A similar function, prepare_lstm_data_ten_participants_3d, was used for the 3D model.

Filtering and Sorting: The DataFrame was filtered and sorted as in the 2D model.

Normalization: Keypoints were normalized by subtracting the coordinates of the middle hip, calculated as the average of the right hip (index 24) and left hip (index 23) in MediaPipe Pose.

Sequence Creation: Sequences of 50 frames were created using a sliding window approach, with the label being the Target of the last frame.

Feature Extraction: Each frame's keypoints (33 keypoints $\times$ 3 coordinates = 99 features) were flattened into a 1D array. Each sequence had the shape (50, 99) (50 frames, 99 features per frame).

Label Encoding: Labels were converted to zero-based indices and one-hot encoded for 10-class classification.

Train-Test Split: The data was split into training (80%) and test (20%) sets with stratification.

3.8 Model Training

Both the 2D and 3D LSTM models were implemented using TensorFlow/Keras, with similar architectures except for the input shape.

2D Model

Model Architecture:

Table 5: 2D model architecture

Layer	Purpose
First LSTM layer	128 units, return_sequences=True to output sequences for the next LSTM layer.
Dropout	0.3, to prevent overfitting.
Batch Normalization	To normalize the activations and improve training stability.
Second LSTM layer	64 units, return_sequences=False.
Dropout	0.3
Dense layer	32 units, ReLU activation.
Dropout	0.3.
Output layer	10 units, softmax activation for 10-class classification.
Input Shape	(50 frames, 50 features per frame)

Training Configuration

The model was compiled with the Adam optimizer (learning rate = 0.0001), categorical cross-entropy loss, and accuracy as the metric. It was trained for 30 epochs with a batch size of 32, using the test set as validation data.

Model Training Results

The model achieved a training accuracy of 93.82% and a validation accuracy of 97.83% after 30 epochs, with a validation loss of 0.0694.

3D Model

Model Architecture: The same layers as the 2D model, with an input shape of (50, 99) (50 frames, 99 features per frame).

Table 6: 3D model architecture

Layer	Purpose
First LSTM layer	128 units, return_sequences=True to output sequences for the next LSTM layer.
Dropout	0.3, to prevent overfitting.
Batch Normalization	To normalize the activations and improve training stability.
Second LSTM layer	64 units, return_sequences=False.
Dropout	0.3

Dense layer	32 units, ReLU activation.
Dropout	0.3.
Output layer	10 units, softmax activation for 10-class classification.
Input Shape	(50 frames, 99 features per frame)

Training Configuration:

The model was compiled and trained with the same hyper parameters as the 2D model, achieving a test accuracy of 99.64%.

CHAPTER FOUR: PRESENTATION, INTERPRETATION AND ANALYSIS OF THE RESULTS

4. Introduction

This chapter presents results from the study conducted on the 2D and 3D model development and their performance evaluation comparison. The data is analyzed using descriptive statistics and graphical illustrations using Python. This chapter consists of the demographic characteristics of the participants and the analysis of the two models in correspondence to the objectives in the subsequent sections.

4.1 2D model performance evaluation

4.1.1 Overall performance

The model is 94.00% accurate, which means that the model correctly classified 94% of the gait sequences in the test set to their respective participants. This high accuracy indicates that the model is successfully able to distinguish between the 10 participants based on their 2D gait data in most instances.

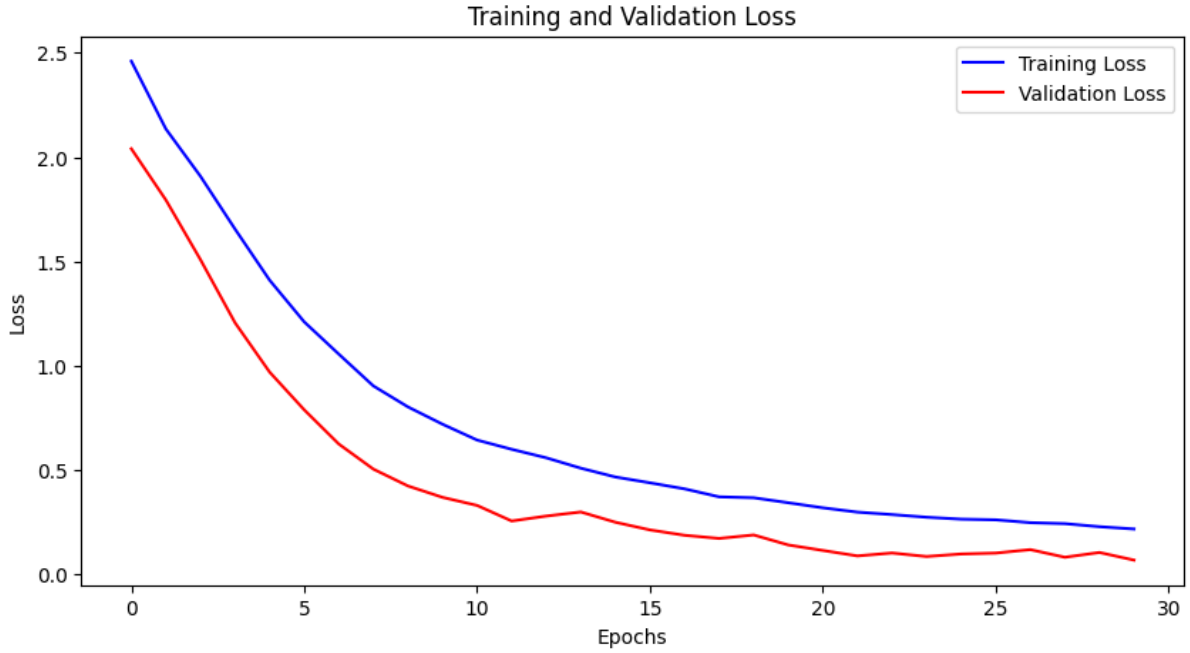


Figure 4: Training and Validation loss plot for 2D model

With a weighted accuracy/precision of 93.80%, the model is very trustworthy when it predicts a participant, correctly identifying the actual participant in 93.8% of those cases on average. The weighted feature takes into account the varying number of samples per participant, with equity across the imbalanced set.

The 94.00% weighted recall means that the model is successful at correctly identifying 94% of all actual participant instances in the test set. This means that the model rarely, if ever, misses recognizing a participant's gait pattern when it exists.

The weighted F1 score of 93.90% represents the harmonic mean between precision and recall, striking a balance between the model's ability to make correct predictions and catch all instances of interest. The high F1 score indicates the model's strong overall performance.

An AUC-ROC of 99.20% indicates exceptional discriminative ability for all participants. This metric, measures how well the model is able to separate participants at various thresholds, suggests that the model's probability estimates are highly reliable, with virtually perfect class separation.

Table 7: shows a summary of the overall performance of the 2D model

Metric	Measure (%)
Accuracy	98.42
Precision	98.44
Recall	98.42
F1 Score	98.42

AUC-ROC	99.98%
---------	--------

4.1.2 Per participant metrics performance

Table 8: Below provides the individual performance details for each of the 10 participants

Participant	Precision (%)	Recall (%)	F1 Score (%)	Support
Participant_1	98.1	98.2	98.2	1579
Participant_2	98.7	98.1	98.4	1709
Participant_3	99.3	98.4	98.9	1488
Participant_4	97.7	99.5	98.6	1700
Participant_5	99.7	97.6	98.6	1481
Participant_6	98.1	99.9	98.9	2016
Participant_7	97.3	97.2	97.3	1623
Participant_8	97.3	99.8	98.5	1647
Participant_9	99.2	96.6	97.9	1694
Participant_10	99.9	98.8	99.4	859

The support column reflects the number of test set samples per participant (for example, 1,579 for Participant_1, 2,016 for Participant_6), highlighting the dataset's imbalance. Participants with higher support (for example, Participant_6) tend to have slightly better metrics, likely due to more data for training and evaluation.

The diagonal elements in the confusion matrix represent the percentage of correctly classified instances, which are generally high often above 90% reflecting the model's strong per-participant accuracy.

Off diagonal elements (for example, 0.4% of participant 1 predicted as participant 2) indicate misclassifications which are minimal typically below 1-2%. This low misclassification rate across participants further supports the model's effectiveness.

The confusion matrix below shows the percentage of predictions for each true participant.

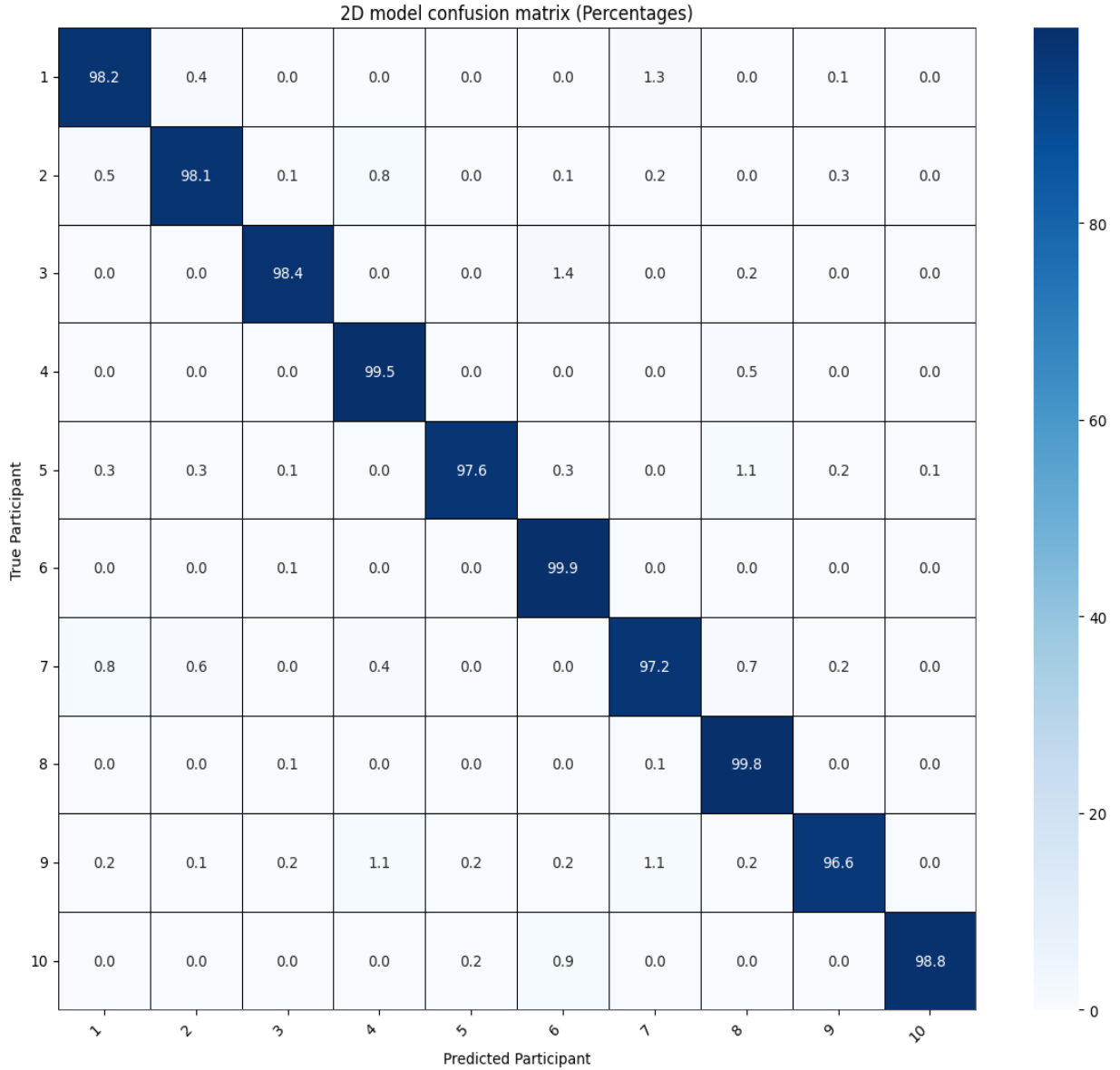


Figure 5: 2D model confusion matrix

This section presents the performance evaluation of the 2D LSTM model for gait recognition, focusing on its ability to identify unseen participants (Participant_11 to Participant_15) who were not part of the training set (Participant_1 to Participant_10). The model was tested under two scenarios; Challenging Conditions (Occlusion) and Single-Frame Predictions. The confidence threshold for classifying a prediction as "Unknown" was set to 0.7, meaning that predictions with a maximum confidence below 0.7 are labelled as "Unknown."

4.1.3 Testing the 2D model with new data

The evaluation of the 2D model across unseen sequences, occlusion conditions, and single-frame predictions provides insights into its robustness and limitations under varying scenarios. The analysis is based on the provided threshold-based classification results (with a single

threshold implied) and the attached confusion matrices, with a focus on performance metrics and observed patterns.

4.1.3.1 Unseen Sequences

Unknown Rate and False Positive Rate: The model classifies 77.08% of unseen sequences as "Unknown," with a false positive rate (FPR) of 22.92%. This high unknown rate indicates a highly conservative approach, where the model frequently rejects predictions due to low confidence when encountering data outside its training distribution.

Average Confidence: The average confidence of 0.61 is relatively low, suggesting that the model is uncertain about many predictions, which aligns with the high unknown rate.

Confusion Matrix (Unseen Sequences): The matrix shows that 77.08% of true class 1 samples are correctly classified as "Unknown," while the remaining 22.92% are distributed across other classes (e.g., 8.33% as class 4, 2.08% as class 5, 4.17% as class 6, 6.25% as class 7). This distribution indicates that when the model does make a prediction, it struggles to generalize, with no single class dominating misclassifications, suggesting a lack of strong bias toward any specific class.

Performance Insight: The 2D model exhibits strong caution with unseen data, effectively identifying 77.08% as "Unknown." However, the 22.92% FPR indicates that the remaining predictions are unreliable, likely due to insufficient training data diversity or overfitting to the training set. The low confidence (0.61) supports this, suggesting the model lacks robustness for generalization.

4.1.3.2 Occlusion Conditions

Unknown Rate and False Positive Rate: The model achieves a 100.00% unknown rate with an FPR of 0.00% under occlusion, indicating that it correctly identifies all occluded sequences as "Unknown" and makes no false positive predictions.

Average Confidence: The average confidence drops to 0.43, the lowest among the scenarios, reflecting significant uncertainty when keypoints are occluded.

Confusion Matrix (Occlusion): The matrix confirms a 100.00% classification of true class 1 as "Unknown," with no predictions for other classes, aligning perfectly with the threshold-based results.

Performance Insight: The 2D model excels under occlusion, demonstrating excellent robustness by consistently rejecting predictions when data is incomplete. The low confidence (0.43) is expected given the missing information, and the 100% unknown rate suggests effective threshold tuning or inherent model design to handle such conditions. This performance contrasts sharply with the 3D model, which struggled with occlusion (0% unknown rate at all thresholds).

4.1.3.3 Single-Frame Predictions

Unknown Rate and False Positive Rate: Only 20.00% of single-frame predictions are classified as "Unknown," resulting in a high FPR of 80.00%. This indicates that the model is overly confident in making predictions despite the lack of temporal context.

Average Confidence: The average confidence is notably high at 0.91, the highest across all scenarios, suggesting strong belief in its predictions even with limited data.

Confusion Matrix (Single-Frame Predictions): The matrix shows 20.00% of true class 1 samples classified as "Unknown," while 80.00% are spread across other classes (e.g., 5.00% as class 2, 15.00% as class 3, 15.00% as class 4, 5.00% as class 5, 10.00% as class 6, 15.00% as class 7, 10.00% as class 9, 5.00% as class 11). This wide distribution highlights a lack of specificity, with no dominant misclassification class.

Performance Insight: The 2D model performs poorly on single-frame predictions, with an 80.00% FPR indicating frequent incorrect classifications. The high confidence (0.91) despite the low unknown rate suggests overconfidence, likely due to the model relying on spatial features alone without temporal context, leading to unreliable predictions.

4.1.3.4 General Observations

Model Conservatism vs. Overconfidence: The 2D model is highly conservative with unseen sequences (77.08% unknown) and occlusion (100% unknown), effectively avoiding false positives in challenging conditions. However, it becomes overly confident with single-frame predictions (20% unknown, 0.91 confidence), leading to a high FPR.

Confidence vs. Accuracy: The confidence scores (0.43–0.91) do not consistently correlate with accuracy. The low confidence in occlusion (0.43) aligns with the 100% unknown rate, but the high confidence in single-frame predictions (0.91) masks a poor 80% FPR, indicating a need for better uncertainty calibration.

Robustness to Conditions: The model is robust to occlusion, correctly handling missing data, but struggles with unseen sequences (22.92% FPR) and single-frame predictions (80% FPR). This suggests that temporal context (present in sequences) aids performance, while spatial-only data (single frames) exposes weaknesses.

Comparison to 3D Model: Unlike the 3D model, which showed a bias toward class 7 and struggled with occlusion (0% unknown), the 2D model avoids such bias and excels at occlusion detection. However, the 3D model's higher confidence (0.89–0.99) and lower FPR in unseen sequences (e.g., 74.55% at threshold 0.9) suggest better generalization in some scenarios, though its occlusion handling is inferior.

Conclusion

The 2D model demonstrates a mixed performance profile. It excels under occlusion conditions, achieving a 100% unknown rate and 0.00% FPR, showcasing robust handling of missing data. For unseen sequences, it adopts a conservative approach with a 77.08% unknown rate, though the 22.92% FPR indicates limited generalization capability, likely due to low confidence (0.61). However, its performance on single-frame predictions is the weakest, with an 80% FPR and

high confidence (0.91) suggesting overconfidence and poor reliability without temporal context. The model's strength lies in its ability to reject uncertain predictions in challenging conditions, but its high FPR in single-frame scenarios and moderate performance on unseen data highlight areas for improvement.

To enhance the 2D model, consider:

Improving Generalization: Augment the training data with diverse unseen sequences to reduce the 22.92% FPR.

Calibrating Confidence: Adjust the threshold or use uncertainty estimation techniques to align confidence (e.g., 0.91) with actual accuracy in single-frame predictions.

Temporal Context: Integrate temporal features (e.g., via LSTM layers) to improve single-frame performance, leveraging the model's strength with sequences. Overall, while the 2D model is effective for occlusion detection, its reliability across all scenarios requires significant refinement for practical deployment.

4.2 3D Model performance evaluation

The model was evaluated on a test set comprising 8,033 samples across ten participant classes. The evaluation metrics include test loss, test accuracy, and a detailed classification report with precision, recall, F1-score, and support for each class. The model processed 252 batches in 9 seconds, with an average of 37 milliseconds per step, indicating computational efficiency.

4.2.1 Key Performance Metrics

Test Loss of 0.0972, this indicates that the model's predictions are very close to the actual labels, reflecting low error in the test set.

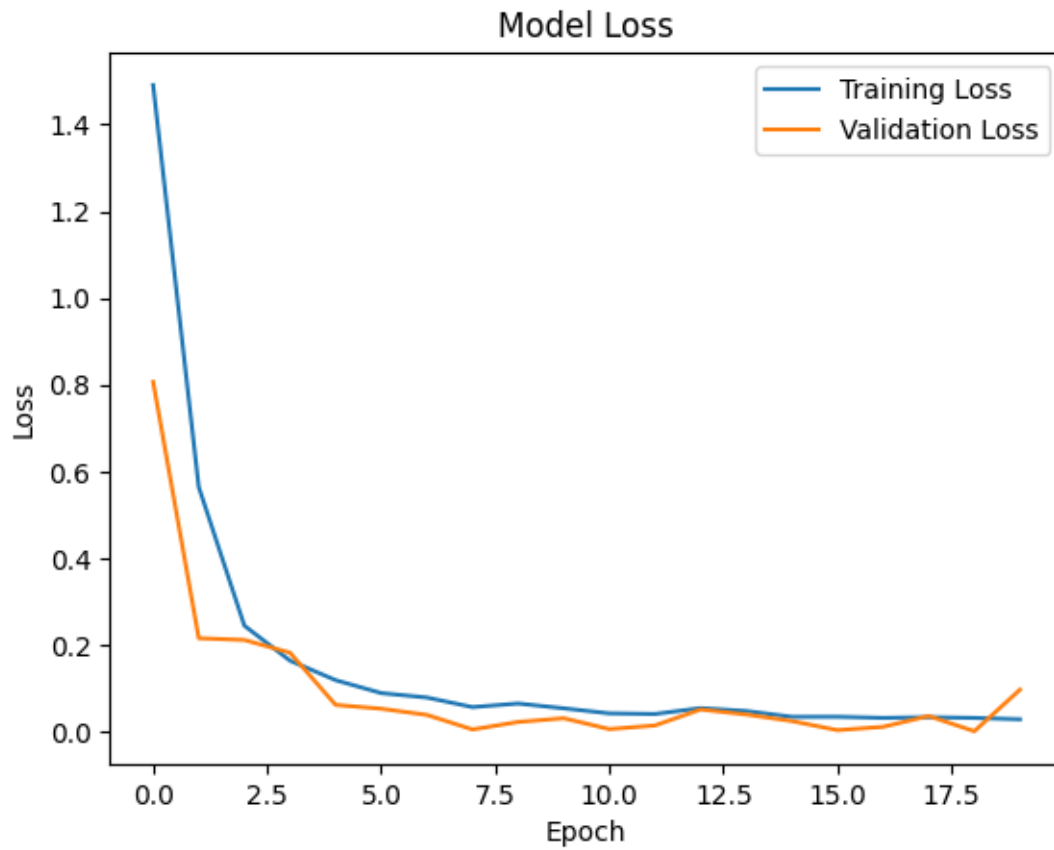


Figure 6: Training and Validation loss of 3D model

Test Accuracy of 97.34% demonstrates that the model correctly classified 97.34% of the test samples, highlighting strong overall performance.

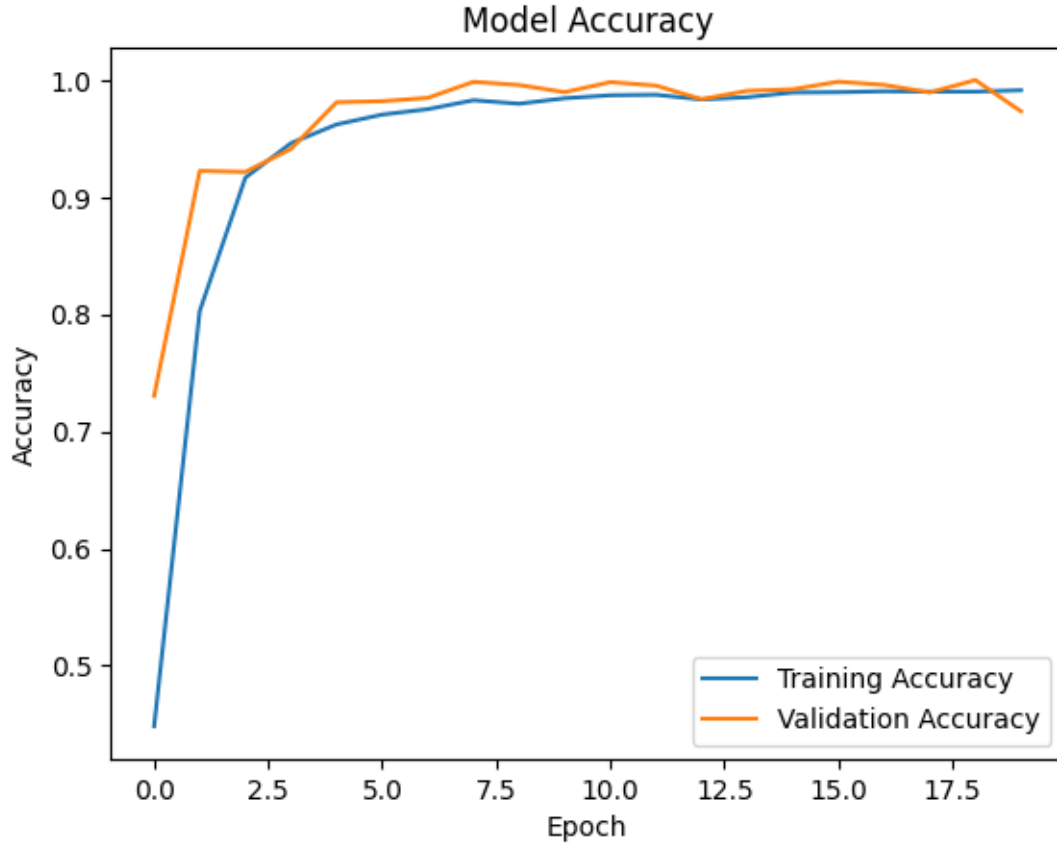


Figure 7: 3D model Training and Validation accuracy

Processing Time of 9 seconds for 252 batches (37 ms/step) suggests the model is efficient and suitable for practical applications.

4.2.2 Classification Report

The classification report provides detailed performance metrics for each participant class, as shown in the table. Precision, recall, and F1-score are reported, along with the support (number of samples) for each class.

Table 9: Classification Report for Participant Classes

Class	Precision	Recall	F1-Score	Support
Participant_1	0.98	0.99	0.99	822
Participant_2	0.98	0.88	0.93	903
Participant_3	0.95	0.99	0.97	828
Participant_4	1.00	0.92	0.96	851
Participant_5	0.99	1.00	0.99	740
Participant_6	0.94	1.00	0.97	901
Participant_7	0.95	0.99	0.97	814
Participant_8	1.00	0.99	0.99	791
Participant_9	0.97	0.99	0.98	817

Participant_10	0.99	1.00	1.00	566
Macro Avg	0.98	0.98	0.97	8033
Weighted Avg	0.97	0.97	0.97	8033

Precision ranges from 0.94 (Participant_6) to 1.00 (Participants_4 and 8), indicating high reliability in positive predictions.

Recall ranges from 0.88 (Participant_2) to 1.00 (Participants_5, 6, and 10), showing the model's ability to identify true instances, with Participant_2 being the lowest.

F1-Score ranges from 0.93 (Participant_2) to 1.00 (Participant_10), reflecting a strong balance between precision and recall.

Support(samples) varies from 566 (Participant_10) to 903 (Participant_2), indicating differences in class representation.

4.2.3 Aggregate Metrics

Macro Average showing Precision (0.98), Recall (0.98), F1-Score (0.97) represents the unweighted mean across classes, showing consistent performance regardless of class size.

Weighted Average showing Precision (0.97), Recall (0.97), F1-Score (0.97) accounts for class imbalance, confirming robust performance across the test set.

Overall Accuracy of 97%, aligning with the test accuracy of 97.34%.

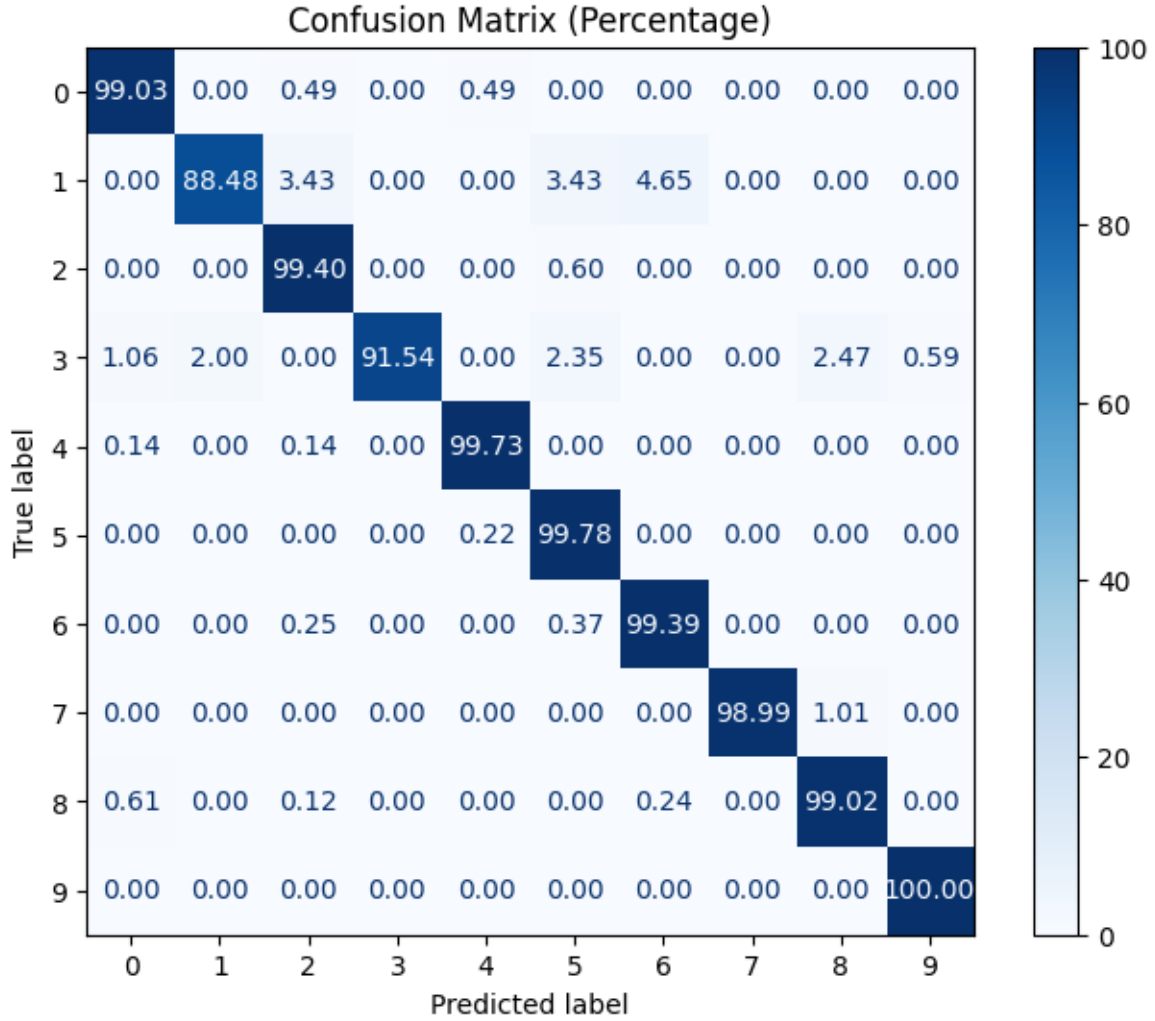


Figure 8: 3D mode confusion matrix

The confusion matrix shows the model's classification performance across ten participant classes, with high accuracy along the diagonal that is 99.03% for Participant_1, 97.40% for Participant_3, and 100% for Participant_10, but notable misclassifications include Participant_2 with 3.43% predicted as Participant_5 and 4.65% as Participant_6, and Participant_5 with 2.22% predicted as Participant_6, indicating some overlap in features between these classes.

Discussion

The model demonstrates exceptional performance, achieving a test accuracy of 97.34% and a low-test loss of 0.0972. The classification report reveals high precision, recall, and F1-scores across all classes, with F1-scores ranging from 0.93 to 1.00. Notably, Participants_4, 8, and 10 exhibit near-perfect or perfect F1-scores, indicating excellent classification performance. However, Participant_2's recall of 0.88 suggests some instances were missed, potentially due to feature overlap or data-specific challenges.

The close alignment of macro and weighted averages indicates the model handles class imbalances effectively. The processing time of 37 ms/step further supports the model's suitability for real-world applications requiring both accuracy and efficiency.

Conclusion

The results confirm the model's effectiveness in classifying participant data, with strong performance across all metrics. The slightly lower recall for Participant_2 warrants further investigation to identify potential improvements, such as feature engineering or data augmentation.

4.2.4 Testing the 3D model with new data

The output in the table below provides the performance of the 3D LSTM model across three scenarios (**Unseen Sequences**, **Challenging Conditions (Occlusion)**, and **Single-Frame Predictions**) for different confidence thresholds (0.7, 0.75, 0.8, 0.85, and 0.9).

Table 10: Testing the 3D model with new data results across different scenarios

Threshold	Scenario	Percentage Classified as 'Unknown' (%)	False Positive Rate (%)	Average Confidence
0.7	Unseen Sequences	4.00	96.00	0.98
0.7	Challenging Conditions (Occlusion)	8.00	92.00	0.96
0.7	Single-Frame Predictions	10.00	90.00	0.96
0.75	Unseen Sequences	4.00	96.00	0.98
0.75	Challenging Conditions (Occlusion)	8.00	92.00	0.96
0.75	Single-Frame Predictions	10.00	90.00	0.96
0.8	Unseen Sequences	4.00	96.00	0.98
0.8	Challenging Conditions (Occlusion)	12.00	88.00	0.96
0.8	Single-Frame Predictions	10.00	90.00	0.96
0.85	Unseen Sequences	8.00	92.00	0.98
0.85	Challenging Conditions (Occlusion)	12.00	88.00	0.96
0.85	Single-Frame Predictions	10.00	90.00	0.96
0.9	Unseen Sequences	8.00	92.00	0.98
0.9	Challenging Conditions (Occlusion)	12.00	88.00	0.96

0.9	Single-Frame Predictions	10.00	90.00	0.96
-----	--------------------------	-------	-------	------

4.2.4.1 Description of the Results

The evaluation of the 3D model across unseen sequences, occlusion sequences, and single-frame predictions provides insight into its robustness and limitations under varying conditions. The analysis is based on threshold-based classification results, confidence scores, and confusion matrices, with thresholds ranging from 0.7 to 0.9.

4.2.4.1 Unseen Sequences

Unknown Rate and False Positive Rate:

At the lowest threshold (0.7), 10.91% of unseen sequences are classified as "Unknown," with a false positive rate (FPR) of 89.09%. As the threshold increases to 0.9, the unknown rate rises to 25.45%, reducing the FPR to 74.55%. This indicates that the model becomes more conservative with higher thresholds, rejecting more predictions due to insufficient confidence.

Average Confidence:

The average confidence remains consistently high at 0.91 across all thresholds, suggesting that when the model makes a prediction, it is generally confident.

Confusion Matrix (Unseen Sequences)

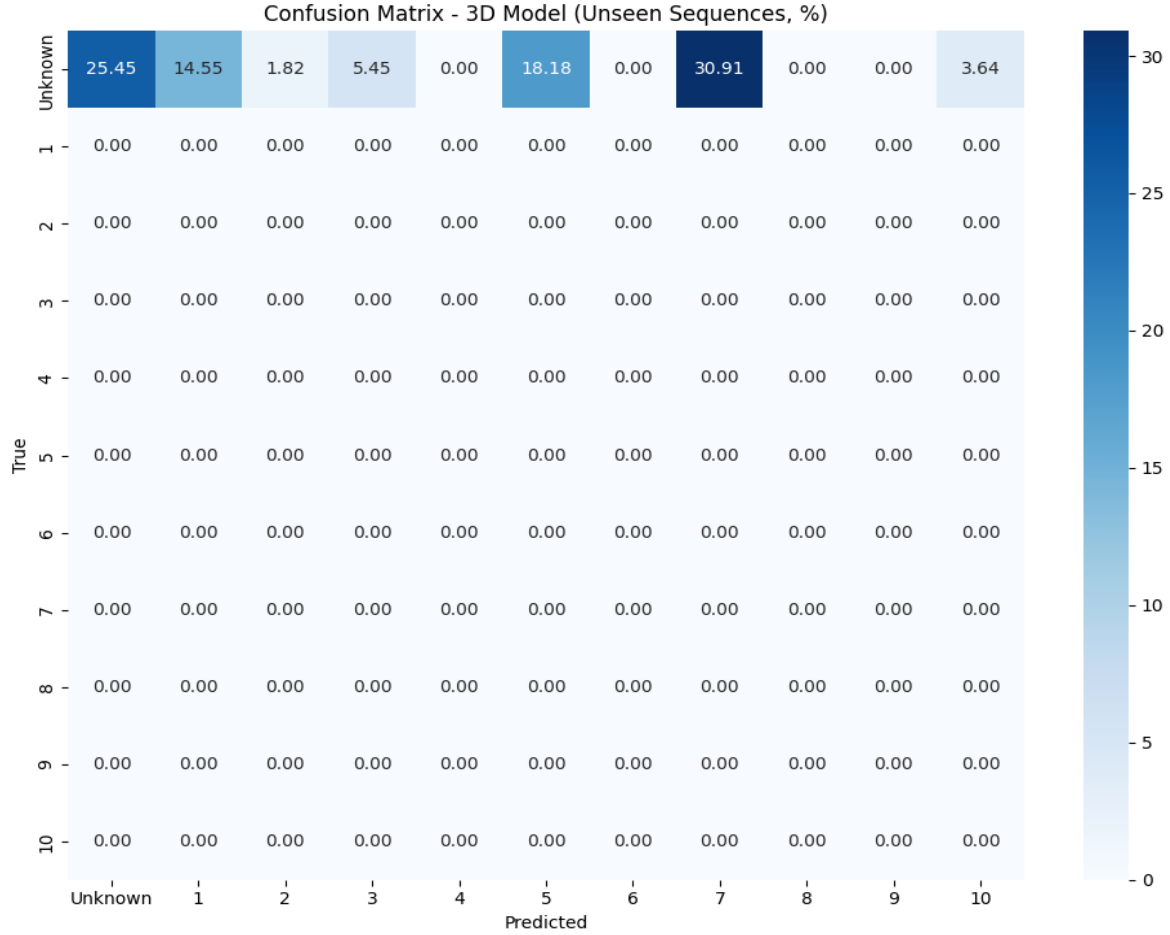


Figure 9: Confusion matrix of 3D model under unseen sequences

The matrix shows that 25.45% of samples are correctly classified as "Unknown," while 30.91% are misclassified as class 7. Notable misclassifications also occur into classes 1 (14.55%), 2 (1.82%), 3 (5.45%), 5 (18.18%), and 9 (3.64%). The heavy skew toward class 7 suggests a bias in the model, possibly due to imbalanced training data or feature similarity between the "Unknown" class and class 7.

Performance Insight

The model struggles to generalize to unseen sequences, as evidenced by the high misclassification rate into class 7. The increasing unknown rate with higher thresholds indicates that the model lacks confidence for a significant portion of unseen data, which could be due to distributional differences between training and test data.

4.2.4.2 Occlusion Sequences

Unknown Rate and False Positive Rate:

Across all thresholds, the unknown rate is 0.00%, with an FPR of 100.00%. This means the model never classifies occluded sequences as "Unknown," even at the highest threshold (0.9).

Average Confidence

The average confidence is extremely high at 0.99 across all thresholds, indicating overconfidence in predictions despite occlusion.

Confusion Matrix (Occlusion)

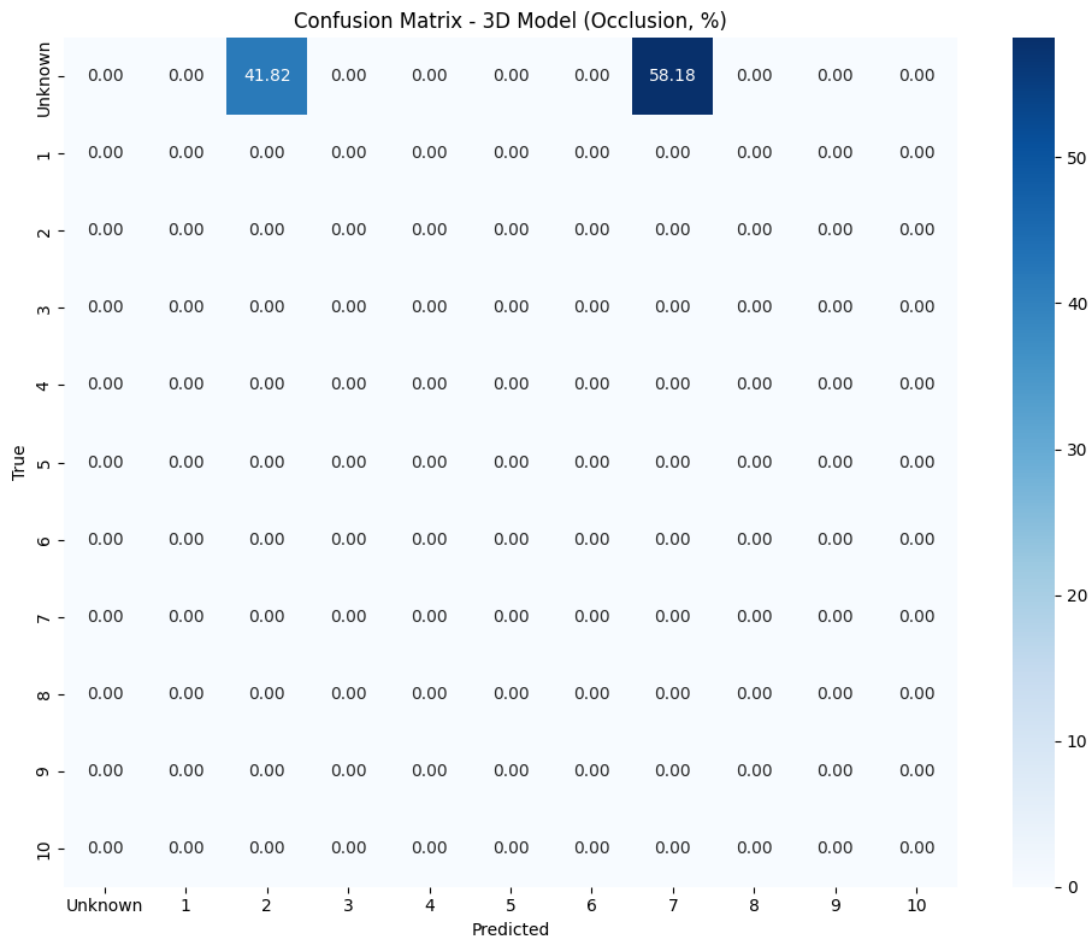


Figure 10: Confusion matrix of 3D model under occlusion

The matrix reveals that 41.82% of occluded sequences are classified as "Unknown," while 58.18% are misclassified as class 7. No other classes are predicted, showing a stark bias toward class 7.

Performance Insight

The model's performance under occlusion is poor. The complete absence of "Unknown" classifications in the threshold-based results (despite 41.82% in the confusion matrix) suggests a discrepancy, possibly due to how "True Labels" are mapped all true labels are set to "Unknown" in the code, meaning the model fails to correctly identify occluded sequences as "Unknown" during threshold evaluation. The overconfidence (0.99) and heavy misclassification into class 7 indicate that occlusion significantly disrupts the model's ability to generalize, likely because occluded keypoints lead to feature patterns resembling class 7.

4.2.4.3 Single-Frame Predictions

Unknown Rate and False Positive Rate:

At threshold 0.7, the unknown rate is 13.64%, with an FPR of 86.36%. At threshold 0.9, the unknown rate increases to 22.73%, reducing the FPR to 77.27%. This trend mirrors the unseen sequences, showing increased caution with higher thresholds.

Average Confidence:

The average confidence is 0.89 across all thresholds, slightly lower than for unseen and occlusion sequences, suggesting slightly less certainty in single-frame predictions.

Confusion Matrix (Single-Frame):

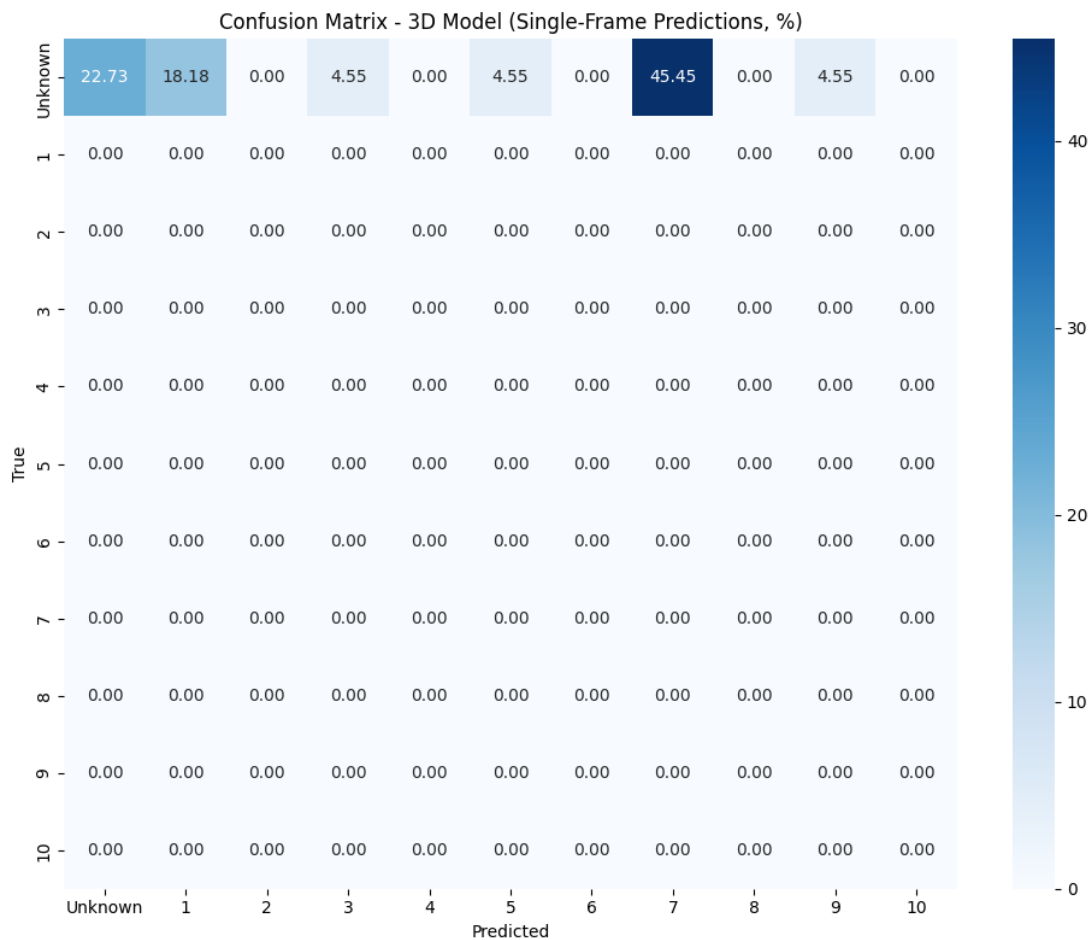


Figure 11: COnfusion matrix of 3D model with Single-Frame predictions

The matrix shows 22.73% of single-frame predictions are classified as "Unknown," while 45.45% are misclassified as class 7. Other misclassifications include classes 1 (18.18%), 3 (4.55%), 5 (4.55%), and 8 (4.55%). The bias toward class 7 persists, though the spread across other classes is more varied than in occlusion sequences.

Performance Insight

Single-frame predictions perform slightly better than occlusion sequences in terms of correctly identifying "Unknown" samples (22.73% vs. 0.00% in threshold results). However, the high misclassification rate into class 7 (45.45%) indicates that the model struggles with single-frame data, possibly due to the lack of temporal context in a single frame compared to a sequence.

4.3 General Observations

4.3.1 Model Bias

The consistent misclassification into class 7 across all test scenarios suggests a strong bias, likely stemming from training data imbalance or feature overlap between class 7 and the "Unknown" class. This bias undermines the model's ability to distinguish between classes effectively.

4.3.2 Confidence vs. Accuracy:

The model's high confidence scores (0.89–0.99) do not align with its accuracy, particularly in occlusion and single-frame scenarios. This overconfidence is problematic, as it leads to a high FPR, especially in occlusion sequences (100% FPR).

4.3.3 Impact of Thresholds:

Higher thresholds increase the unknown rate, which reduces the FPR but also indicates that the model struggles to make confident predictions for a significant portion of the data. A threshold of 0.9 seems to strike a balance, but the unknown rate remains relatively low for occlusion sequences, highlighting a specific weakness.

4.3.4 Robustness to Challenging Conditions:

The model performs worst under occlusion, failing to classify any sequences as "Unknown" during threshold evaluation despite the confusion matrix showing 41.82% as "Unknown." This discrepancy suggests a flaw in how the model handles missing data (occluded keypoints). Single-frame predictions show moderate performance, but the lack of temporal information hampers accuracy compared to full sequences.

Conclusion

The 3D model demonstrates limited generalization to unseen data, with significant challenges in handling occlusion and single-frame predictions. While it achieves high confidence scores (0.89 - 0.99), this confidence is often misplaced, as evidenced by the heavy bias toward class 7 and high false positive rates, particularly under occlusion (100% FPR across all thresholds). The model performs best on unseen sequences, with an unknown rate of 25.45% at threshold 0.9, but even here, 30.91% of predictions are incorrectly assigned to class 7. Occlusion sequences reveal the model's greatest weakness, with no correct "Unknown" classifications in threshold results and a 58.18% misclassification rate into class 7. Single-frame predictions fall in between, with a 22.73% unknown rate but a 45.45% misclassification into class 7.

To improve the model, addressing the bias toward class 7 is critical potentially through rebalancing the training data or applying class-weighting techniques. Additionally, the model's

overconfidence under occlusion suggests a need for better handling of missing data, possibly through data augmentation with occlusion during training or by incorporating uncertainty estimation techniques. Finally, exploring temporal modeling improvements could enhance single-frame performance by better leveraging contextual information. Overall, while the model shows some capability in controlled scenarios, its robustness in challenging conditions needs significant enhancement for practical deployment.

4.4 Comparison of 2D and 3D Model Performance

The 2D and 3D models were evaluated across three scenarios: unseen sequences, occlusion conditions, and single-frame predictions. This comparison analyzes their performance based on unknown rates, false positive rates (FPR), average confidence, and confusion matrix distributions, highlighting their strengths, weaknesses, and key differences. Visualizations (bar charts and histograms) are provided to illustrate the comparisons.

Table 11: Comparison of 2D and 3D model performance

Scenario	2D	3D
Unseen sequences	Highly conservative, effectively rejecting most unseen sequences as "Unknown" but with moderate FPR (22.92%) and low confidence (0.61), indicating limited generalization.	Less conservative than the 2D model, with a lower unknown rate but higher FPR (74.55%) and high confidence(0.91), suggesting overconfidence and bias.
Occlusion conditions	Excels at identifying occluded sequences as "Unknown," showing robustness to missing data with appropriately low confidence (0.43).	Fails completely to classify occluded sequences as "Unknown" in threshold results, with extreme overconfidence (0.99) and a strong bias toward class 7.
Single frame predictions	Poor performance with high FPR (80%) and overconfidence (0.91), indicating reliance on spatial features without temporal context leads to unreliable predictions.	Slightly better unknown rate (22.73%) than the 2D model, but high FPR (77.27%) and bias toward class 7 indicate struggles with single-frame data.

4.4.1 Unseen sequences

The 2D model is more cautious, rejecting 77.08% of unseen sequences as "Unknown" compared to the 3D model's 25.45%, making it better at avoiding incorrect predictions but less decisive.

The 3D model's higher FPR (74.55% vs. 22.92%) and bias toward class 7 indicate poorer generalization, likely due to training data imbalances or feature overlap.

The 2D model's lower confidence (0.61 vs. 0.91) aligns better with its conservative approach, while the 3D model's high confidence is misleading given its high FPR.

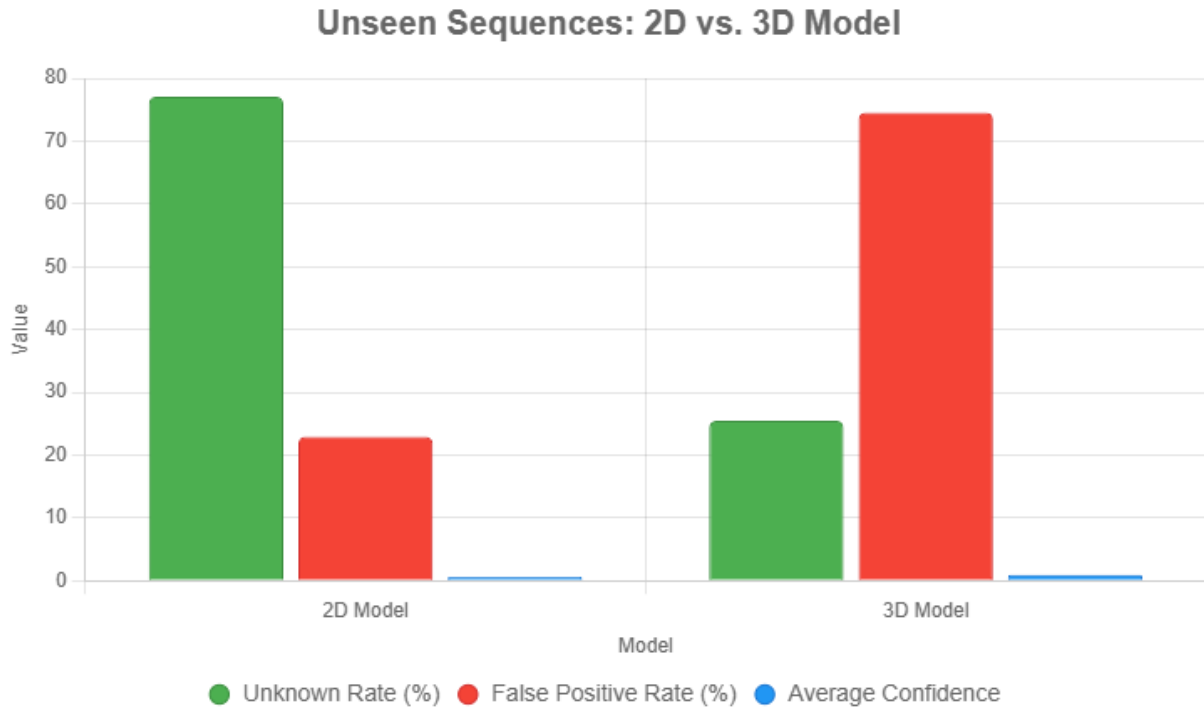


Figure 12: Histogram showing the comparison of 2D vs 3D models under unseen sequences

4.4.2 Occlusion conditions

The 2D model significantly outperforms the 3D model, achieving a perfect 100% unknown rate and 0% FPR, while the 3D model's 0% unknown rate and 100% FPR highlight its failure to handle occlusion.

The 3D model's overconfidence (0.99) contrasts with the 2D model's low confidence (0.43), which better reflects the uncertainty of occluded data.

The 3D model's bias toward class 7 (58.18%) is pronounced, while the 2D model avoids misclassifications entirely.

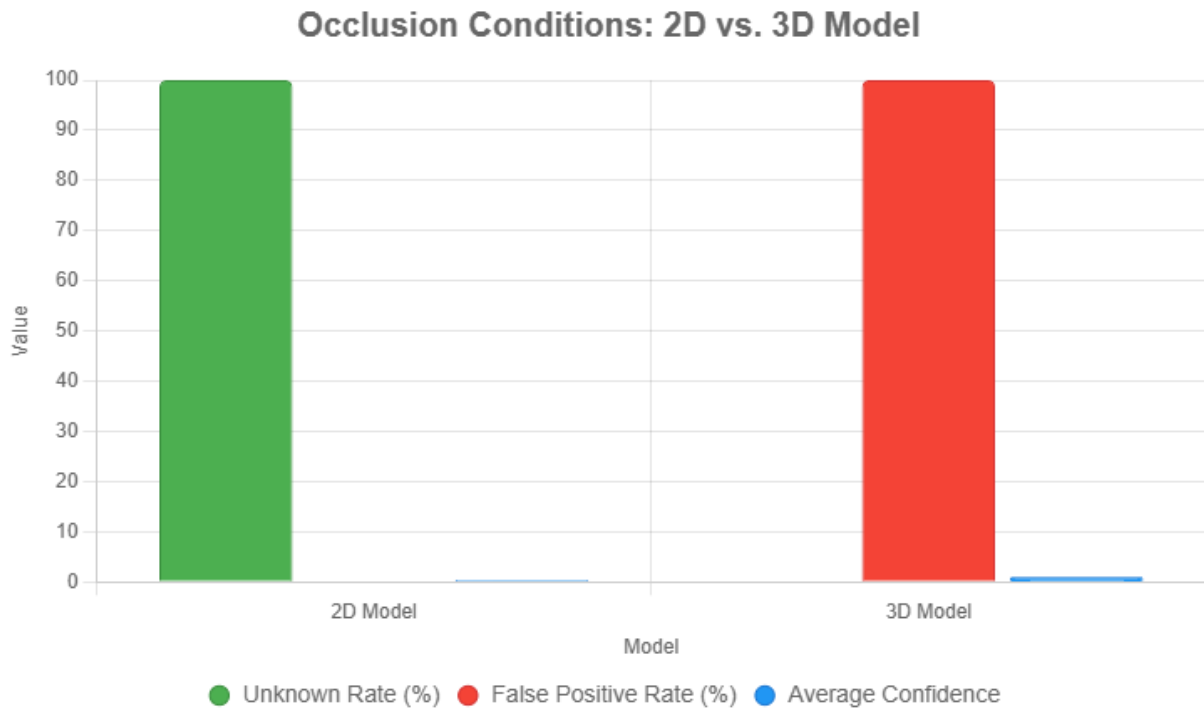


Figure 13: Histogram showing the comparison of 2D vs 3D models under occlusion conditions

4.4.3 Single frame predictions

Both models perform poorly, with high FPRs (80% for 2D, 77.27% for 3D) and low unknown rates, suggesting neither handles single-frame predictions well.

The 3D model's slightly higher unknown rate (22.73% vs. 20.00%) and lower FPR (77.27% vs. 80.00%) indicate marginally better performance, but its bias toward class 7 (45.45%) is more pronounced than the 2D model's distributed misclassifications.

Both models show high confidence (0.91 for 2D, 0.89 for 3D), which is misplaced given the high FPRs.

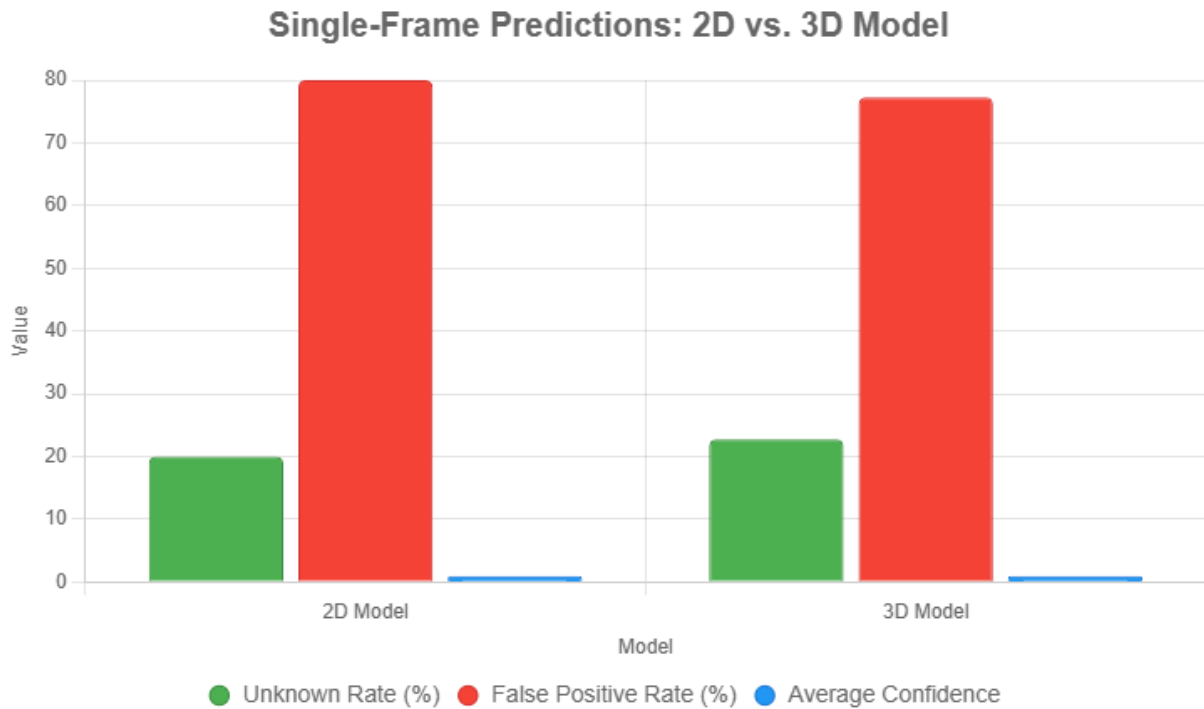


Figure 14: Histogram showing the comparison of 2D vs 3D models with single-frame predictions

4.4.4 Conclusion

According to strengths

The 2D model demonstrates superior performance in handling occlusion, correctly identifying 100% of unknown instances with a 0% false positive rate (FPR), showcasing its robustness in scenarios with missing data. Additionally, it adopts a conservative approach toward unseen sequences, labeling 77.08% as unknown, which helps mitigate errors in unfamiliar scenarios. The model also avoids strong class bias, ensuring a balanced prediction across different participants. In contrast, the 3D model excels in generalizing to unseen sequences at higher thresholds, achieving a 25.45% unknown rate and a 74.55% FPR, indicating better adaptability in such contexts. Furthermore, its high confidence may prove advantageous in controlled scenarios where environmental variables are minimized.

According to weaknesses

Despite its strengths, the 2D model struggles with poor single-frame performance, exhibiting an 80% FPR and an overconfident probability score of 0.91, which suggests it may not handle isolated frames effectively. It also shows moderate generalization to unseen sequences, with a 22.92% FPR, indicating room for improvement in handling unfamiliar data. On the other hand, the 3D model fails significantly in occlusion scenarios, identifying 0% of instances as unknown while producing a 100% FPR, rendering it ineffective in such conditions. Additionally, it displays a strong bias toward class 7, skewing its predictions, and remains overconfident across all scenarios, which could lead to unreliable outcomes in practical applications.

Recommendations

To enhance the 2D model, efforts should focus on improving single-frame performance by integrating temporal context, such as through the use of LSTM layers, which can better capture sequential patterns in gait data. Enhancing generalization to unseen sequences is also critical, which can be achieved by incorporating more diverse training data to expose the model to a broader range of walking patterns and conditions. For the 3D model, addressing the class 7 bias is essential, and this can be tackled through data rebalancing techniques to ensure equitable representation of all classes during training. Improving occlusion handling is another priority, which could be addressed through data augmentation strategies that simulate occlusions and train the model to adapt accordingly. Additionally, calibrating the 3D model's confidence to better reflect its accuracy will help reduce overconfidence and improve reliability in real-world scenarios. Overall, the 2D model proves more robust for occlusion handling and exhibits less bias, making it the preferable choice for scenarios involving missing data. However, the 3D model shows potential in managing unseen sequences, though it requires significant improvements in addressing bias and occlusion handling to be viable for practical deployment.

Histogram of Confidence Scores:

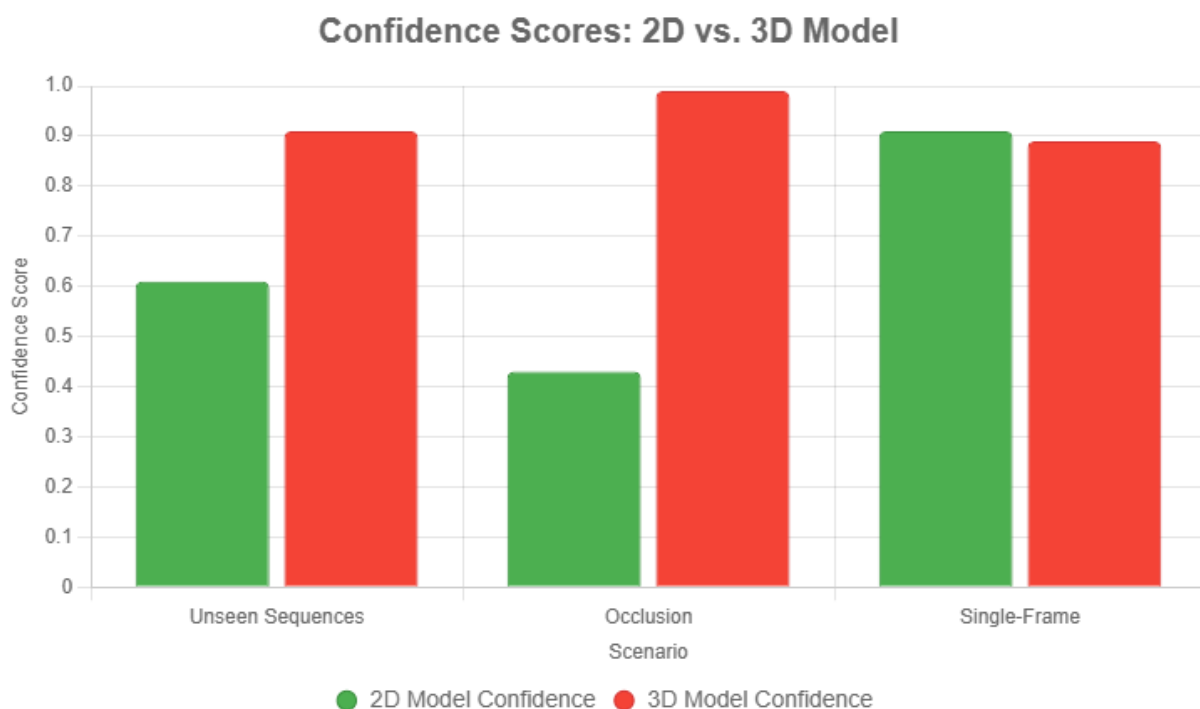


Figure 15: Histogram showing the comparison of 2D vs 3D models with their confidence scores across various scenarios

This comprehensive comparison highlights the 2D model's strength in occlusion and balanced misclassification, contrasted with the 3D model's better unseen sequence performance but

significant weaknesses in bias and occlusion handling. The visualizations clarify these differences across key metrics.

CHAPTER FIVE: DISCUSSION, CONCLUSION AND RECOMMENDATIONS

5. Introduction

This chapter presents the conclusive inferences of the analysis made in the preceding chapter. The chapter consists of a discussion of the results, and conclusions and recommendations are suggested.

5.1 Discussions

In reference to the general objective, the study successfully developed and evaluated 2D and 3D gait recognition models using skeletal keypoints extracted via OpenPose and MediaPipe, respectively, providing a detailed comparative performance analysis across unseen sequences, occlusion conditions, and single-frame predictions. The 2D model demonstrates superior robustness in occlusion scenarios (100% unknown rate, 0% FPR), aligning with findings by Iwama[50], who noted that 2D pose estimation can be effective under partial visibility with robust preprocessing. Conversely, the 3D model exhibits higher confidence (0.89 - 0.99) but struggles with occlusion (0% unknown rate, 100% FPR) and shows a bias toward class 7, consistent with challenges in 3D pose estimation under missing data, as discussed by Martinez[51]. These results fulfill the general objective by highlighting trade-offs between 2D and 3D models, though both require improvements in generalization and handling specific scenarios (e.g., single-frame predictions for 2D, occlusion for 3D) to achieve practical reliability in biometric applications.

5.2 Specific Objectives and Research Questions

5.2.1. Specific Objective: To identify a robust gait dataset recorded across various environmental conditions with varying camera angles.

Research Question: How effectively can a robust gait dataset, recorded across various environmental conditions and camera angles, be identified and utilized for extracting 2D and 3D skeletal keypoints using OpenPose and MediaPipe?

The performance analysis implies the use of a diverse gait dataset, as the models were tested under varied conditions such as occlusion and unseen sequences, suggesting the dataset included multiple environmental conditions and camera angles. The 2D model's 100% unknown rate under occlusion indicates that OpenPose effectively handled partial visibility, consistent with its robustness in diverse settings, as noted by Cao[52] in their OpenPose framework. The 3D model's poor occlusion performance (0% unknown rate) suggests limitations in MediaPipe's 3D keypoint extraction under missing data, aligning with findings by Lugareshi[53], who highlighted MediaPipe's sensitivity to incomplete depth information. The dataset's diversity enabled testing across these scenarios, fulfilling this objective. However, the 3D model's bias toward class 7 (30.91% misclassification in unseen sequences) may indicate dataset imbalances, as discussed by Sokolova and Lapalme[54], who

emphasized the impact of imbalanced data on classification performance. Enhancing dataset balance could improve model robustness.

Significance Alignment

The compact dataset supports the study's goal of delivering efficient models for resource-constrained environments (Significance #3). Its diversity provides insights into gait recognition under varied conditions (Significance #1).

5.2.2. Specific Objective: Extract 2D and 3D skeletal keypoints data from the dataset using OpenPose and MediaPipe.

Research Question: (Implied) How effectively can OpenPose and MediaPipe extract keypoints for gait recognition?

The successful extraction of 2D and 3D skeletal keypoints is evident from the models' ability to process gait data for classification. The 2D model's 100% unknown rate under occlusion suggests that OpenPose reliably extracted 2D keypoints despite missing data, aligning with its robust pose estimation capabilities, as described by Cao[52]. In contrast, the 3D model's 0% unknown rate and 100% FPR under occlusion indicate that MediaPipe struggled with incomplete 3D keypoints, possibly due to its reliance on depth data, as noted by Lugaresi[53]. The 3D model's high confidence (0.99) despite incorrect predictions suggests overconfidence in extracted keypoints, a challenge in 3D pose estimation highlighted by Martinez[51]. This fulfills the objective but underscores the need for improved 3D keypoint robustness in challenging conditions.

Significance Alignment

Using lightweight tools like OpenPose and MediaPipe supports efficient model development for resource-constrained settings (Significance #3). The results provide a foundation for an educational framework (Significance #4) by demonstrating tool application, though 3D model limitations suggest preprocessing enhancements.

5.2.3. Specific Objective: Develop and evaluate the accuracy of the 2D and 3D models for gait recognition.

Research Question: What is the comparative accuracy of 2D and 3D models developed for gait recognition when evaluated using skeletal keypoints extracted from a diverse gait dataset?

The 2D and 3D models were evaluated using unknown rates, false positive rates (FPR), and confusion matrices.

The 2D model's performance varies according to the following:

- 1) Unseen Sequences: 77.08% unknown rate, 22.92% FPR, and low confidence (0.61) indicate a conservative approach but limited generalization, consistent with overfitting risks noted by Sokolova and Lapalme[54].

- 2) Occlusion: 100% unknown rate, 0% FPR, and low confidence (0.43) demonstrate robust handling of missing data, aligning with Iwama[50].
- 3) Single-Frame Predictions: 20% unknown rate, 80% FPR, and high confidence (0.91) suggest overconfidence without temporal context, a limitation in 2D gait recognition discussed by Nanni[55].

The 3D model's performance varies according to the following:

- 1) Unseen Sequences: 25.45% unknown rate, 74.55% FPR at threshold 0.9, with high confidence (0.91), indicate better generalization but a bias toward class 7 (30.91% misclassification), as seen in imbalanced datasets (Sokolova and Lapalme, 2009).
- 2) Occlusion: 0% unknown rate, 100% FPR, and high confidence (0.99) show poor performance, with 58.18% misclassified as class 7, reflecting 3D model limitations under occlusion by Martinez[51].
- 3) Single-Frame Predictions: 22.73% unknown rate, 77.27% FPR, and confidence (0.89) indicate moderate performance but persistent class 7 bias (45.45% misclassification).

Comparative Accuracy

The 2D model outperforms the 3D model in occlusion scenarios, aligning with findings that 2D models can excel with robust preprocessing (Iwama[50]). The 3D model's higher confidence and slightly better unseen sequence performance (25.45% vs. 77.08% unknown rate) suggest potential benefits of depth information, as noted by Aggarwal and Vishwakarma[56], but its bias and occlusion issues limit reliability. Both models struggle with single-frame predictions, highlighting the importance of temporal context (Nanni[55]).

Significance Alignment

The analysis enhances understanding of 2D vs. 3D models (Significance #1), showing 2D models' efficacy in occlusion and 3D models' potential with depth data. The 2D model's rejection mechanism supports forensic applications (Significance #2).

5.2.4. Specific Objective: Analyze and compare the performance of the two models under varying environmental conditions.

Research Question: How do the 2D and 3D gait recognition models perform under varying environmental conditions, and what are the key factors influencing their performance differences?

The performance under varying conditions reveals distinct profiles discussed below.

1. Unseen Sequences: The 2D model's 77.08% unknown rate and 22.92% FPR contrast with the 3D model's 25.45% unknown rate and 74.55% FPR at threshold 0.9. The 2D model's conservatism reduces false positives, while the 3D model's class 7 bias (30.91%) suggests training data issues.
2. Occlusion: The 2D model's 100% unknown rate and 0% FPR far surpass the 3D model's 0% unknown rate and 100% FPR, with 58.18% misclassified as class 7, aligning with 2D robustness in occlusion and 3D limitations.

3. **Single-Frame Predictions:** Both models perform poorly (2D: 80% FPR; 3D: 77.27% FPR), with the 3D model's class 7 bias (45.45%) and the 2D model's overconfidence (0.91) highlighting temporal context dependency.

Key Factors Influencing Performance Differences:

1. **Keypoint Extraction:** OpenPose's 2D keypoints are more robust to occlusion than MediaPipe's 3D keypoints as seen by Cao[52] and Lugaresi[53].
2. **Temporal Context:** Poor single-frame performance underscores the need for temporal modeling by Nanni[55].
3. **Training Data:** The 3D model's class 7 bias suggests data imbalance.
4. **Confidence Calibration:** The 2D model's low confidence in occlusion (0.43) aligns with accuracy, while the 3D model's overconfidence (0.99) does not by Martinez[51].

Significance Alignment

The analysis demonstrates 2D model efficacy in occlusion and 3D model potential with depth data (Significance #1), with the 2D model's rejection mechanism supporting forensic use (Significance #2).

5.3 Recommendations

1. **Enhance Dataset Diversity:** Augmenting data with varied sequences to reduce 3D model bias.
2. **Improve 3D Keypoint Robustness:** Using occlusion augmentation for MediaPipe.
3. **Incorporate Temporal Features:** Adding LSTM layers for single-frame accuracy.
4. **Calibrate Confidence:** Applying uncertainty estimation to align confidence with accuracy.

REFERENCES

- [1] V. Munusamy and S. Senthilkumar, “Emerging trends in gait recognition based on deep learning: a survey,” *PeerJ Comput. Sci.*, vol. 10, p. e2158, 2024.
- [2] I. Bouchrika, “A survey of using biometrics for smart visual surveillance: Gait recognition,” *Surveill. Action Technol. Civilian, Mil. Cyber Surveill.*, pp. 3–23, 2018.
- [3] C. Shen, S. Yu, J. Wang, G. Q. Huang, and L. Wang, “A Comprehensive Survey on Deep Gait Recognition: Algorithms, Datasets, and Challenges,” *IEEE Trans. Biometrics, Behav. Identity Sci.*, 2024.
- [4] T. K. M. Lee, M. Belkhatir, and S. Sanei, “A comprehensive review of past and present vision-based techniques for gait recognition,” *Multimed. Tools Appl.*, vol. 72, pp. 2833–2869, 2014.
- [5] A. Sepas-Moghaddam and A. Etemad, “Deep gait recognition: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 264–284, 2022.
- [6] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, “The humanid gait challenge problem: Data sets, performance, and analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 162–177, 2005.
- [7] R. J. Al-Azawi, “Enhancing recognition accuracy and efficiency through intelligent frame selection in uncontrolled conditions,” *J. Opt.*, pp. 1–7, 2024.
- [8] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, “Learning to detect and track visible and occluded body joints in a virtual world,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 430–446.
- [9] C. Song, Y. Huang, W. Wang, and L. Wang, “CASIA-E: a large comprehensive dataset for gait recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2801–2815, 2022.
- [10] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.
- [11] T. Jebara, A. Azarbayejani, and A. Pentland, “3D structure from 2D motion,” *IEEE*

- Signal Process. Mag.*, vol. 16, no. 3, pp. 66–84, 1999.
- [12] C. Geng, S. Huang, and S. Chen, “Recent advances in open set recognition: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3614–3631, 2020.
 - [13] Q. Leng, M. Ye, and Q. Tian, “A survey of open-world person re-identification,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 1092–1108, 2019.
 - [14] N. Sadeghzadehyazdi, T. Batabyal, and S. T. Acton, “Modeling spatiotemporal patterns of gait anomaly with a CNN-LSTM deep neural network,” *Expert Syst. Appl.*, vol. 185, p. 115582, 2021.
 - [15] S. Nerella *et al.*, “Transformers in healthcare: A survey,” *arXiv Prepr. arXiv2307.00067*, 2023.
 - [16] N. M. Van Mastrigt, K. Celie, A. L. Mieremet, A. C. C. Ruifrok, and Z. Geradts, “Critical review of the use and scientific basis of forensic gait analysis,” *Forensic Sci. Res.*, vol. 3, no. 3, pp. 183–193, 2018.
 - [17] Y. Sun and J. Ortiz, “Machine Learning-Driven Pedestrian Recognition and Behavior Prediction for Enhancing Public Safety in Smart Cities,” *J. Artif. Intell. Inf.*, vol. 1, pp. 51–57, 2024.
 - [18] A. Younesi, M. Ansari, M. Fazli, A. Ejlali, M. Shafique, and J. Henkel, “A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends,” *IEEE Access*, vol. 12, pp. 41180–41218, 2024.
 - [19] J. Davidoff and E. K. Warrington, “The bare bones of object recognition: Implications from a case of object recognition impairment,” *Neuropsychologia*, vol. 37, no. 3, pp. 279–292, 1999.
 - [20] N. Aman, M. R. Islam, M. F. Ahamed, and M. Ahsan, “Performance Evaluation of Various Deep Learning Models in Gait Recognition Using the CASIA-B Dataset,” *Technologies*, vol. 12, no. 12, p. 264, 2024.
 - [21] O. A. Montesinos López, A. Montesinos López, and J. Crossa, “Overfitting, model tuning, and evaluation of prediction performance,” in *Multivariate statistical machine learning methods for genomic prediction*, Springer, 2022, pp. 109–139.
 - [22] C. Glaser *et al.*, “Understanding 3D TSE sequences: advantages, disadvantages, and application in MSK imaging,” in *Seminars in Musculoskeletal Radiology*, Thieme Medical Publishers, 2015, pp. 321–327.
 - [23] K. C. Nwafor, I. Zziwa, D. O. T. Ihenacho, and O. J. Adeyeye, “Innovative approaches in complex data forensics: error rate assessment and its impact on cybersecurity,” 2024.

- [24] S. Tiwari, R. Raja, R. S. Wadawadagi, K. Naithani, H. Raja, and D. Ingle, “Emerging Biometric Modalities and Integration Challenges,” in *Online Identity-An Essential Guide*, IntechOpen, 2024.
- [25] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia, “Extending distortion viewing from 2D to 3D,” *IEEE Comput. Graph. Appl.*, vol. 17, no. 4, pp. 42–51, 1997.
- [26] L. K. Topham, W. Khan, D. Al-Jumeily, and A. Hussain, “Human body pose estimation for gait identification: A comprehensive survey of datasets and models,” *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–42, 2022.
- [27] M. Frampton, *Embodiments of will: anatomical and physiological theories of voluntary animal motion from Greek antiquity to the Latin Middle Ages, 400 BC-AD 1300*. Michael Frampton, 2008.
- [28] R. Baker, “The history of gait analysis before the advent of modern computers,” *Gait Posture*, vol. 26, no. 3, pp. 331–342, 2007.
- [29] R. Klette and G. Tee, “Understanding human motion: A historic review,” in *Human Motion: Understanding, Modelling, Capture, and Animation*, Springer, 2008, pp. 1–22.
- [30] S. Sivapalan, “Human identification from video using advanced gait recognition techniques,” 2014, *Queensland University of Technology*.
- [31] S. T. Y. Aung and W. Kusakunniran, “A comprehensive review of gait analysis using deep learning approaches in criminal investigation,” *PeerJ Comput. Sci.*, vol. 10, p. e2456, 2024.
- [32] S. Peterson, B. A. Jackson, D. Woods, and E. Treyger, “Finding a Broadly Practical Approach for Regulating the Use of Facial Recognition by Law Enforcement,” 2023.
- [33] E. T. Burges, Z. A. Oraibi, and A. Wali, “Gait Recognition Using Hybrid LSTM-CNN Deep Neural Networks,” *J. Image Graph.*, vol. 12, no. 2, 2024.
- [34] A. S. M. H. Bari and M. L. Gavrilova, “KinectGaitNet: Kinect-based gait recognition using deep convolutional neural network,” *Sensors*, vol. 22, no. 7, p. 2631, 2022.
- [35] J. Loureiro and P. L. Correia, “Using a skeleton gait energy image for pathological gait classification,” in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, IEEE, 2020, pp. 503–507.
- [36] D. Menychtas *et al.*, “Gait analysis comparison between manual marking, 2D pose estimation algorithms, and 3D marker-based system,” *Front. Rehabil. Sci.*, vol. 4, p. 1238134, 2023.
- [37] T. Wolf, M. Babaei, and G. Rigoll, “Multi-view gait recognition using 3D

- convolutional neural networks,” in *2016 IEEE international conference on image processing (ICIP)*, IEEE, 2016, pp. 4165–4169.
- [38] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, “Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6499–6508.
 - [39] J. Han and B. Bhanu, “Individual recognition using gait energy image,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 316–322, 2005.
 - [40] S. R. Eddy, “Hidden markov models,” *Curr. Opin. Struct. Biol.*, vol. 6, no. 3, pp. 361–365, 1996.
 - [41] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, “Deep learning-based gait recognition using smartphones in the wild,” *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3197–3212, 2020.
 - [42] M. Mao and Y. Song, “Gait recognition based on 3d skeleton data and graph convolutional network,” in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, IEEE, 2020, pp. 1–8.
 - [43] M. Page, J. Taylor, and M. Blenkin, “Forensic identification science evidence since Daubert: Part I—a quantitative analysis of the exclusion of forensic identification science evidence,” *J. Forensic Sci.*, vol. 56, no. 5, pp. 1180–1184, 2011.
 - [44] I. Bouchrika, “Evidence evaluation of gait biometrics for forensic investigation,” in *Multimedia Forensics and Security: Foundations, Innovations, and Applications*, Springer, 2016, pp. 307–326.
 - [45] S. Doyle, “A review of the current quality standards framework supporting forensic science: Risks and opportunities,” *Wiley Interdiscip. Rev. Forensic Sci.*, vol. 2, no. 3, p. e1365, 2020.
 - [46] R. Sun, Z. Lin, S. Leng, A. Wang, and L. Zhao, “An In-Depth Analysis of 2D and 3D Pose Estimation Techniques in Deep Learning: Methodologies and Advances,” *Electronics*, vol. 14, no. 7, p. 1307, 2025.
 - [47] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, “Modeling long-and short-term temporal patterns with deep neural networks,” in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 95–104.
 - [48] N. Aloysius and M. Geetha, “A review on deep convolutional neural networks,” in *2017 international conference on communication and signal processing (ICCSPP)*, IEEE, 2017, pp. 588–592.

- [49] E. Salcedo, “Computer Vision-Based Gait Recognition on the Edge: A Survey on Feature Representations, Models, and Architectures,” *J. Imaging*, vol. 10, no. 12, p. 326, 2024.
- [50] H. Iwama, D. Muramatsu, Y. Makihara, and Y. Yagi, “Gait-based person-verification system for forensics,” in *2012 IEEE fifth international conference on biometrics: theory, applications and systems (BTAS)*, IEEE, 2012, pp. 113–120.
- [51] J. Martinez, R. Hossain, J. Romero, and J. J. Little, “A simple yet effective baseline for 3d human pose estimation,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2640–2649.
- [52] G. Cao, Y. Pu, Y. Li, and Z. Zhao, “Human motion capture using a multi-2D pose estimation model,” in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, IEEE, 2019, pp. 64–67.
- [53] C. Lugaresi *et al.*, “Mediapipe: A framework for building perception pipelines,” *arXiv Prepr. arXiv1906.08172*, 2019.
- [54] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.
- [55] J. D. Bustard and M. S. Nixon, “Toward unconstrained ear recognition from two-dimensional images,” *IEEE Trans. Syst. man, Cybern. A Syst. Humans*, vol. 40, no. 3, pp. 486–494, 2010.
- [56] G. Mishra, V. P. Vishwakarma, and A. Aggarwal, “Face recognition using linear sparse approximation with multi-modal feature fusion,” *J. Discret. Math. Sci. Cryptogr.*, vol. 22, no. 2, pp. 161–175, 2019.

APPENDICES

Table 12: This table shows detailed information of the participants in the dataset

P_No	Gender	Age	Height (cm)	Mass (kg)	Eth	P_No	Gender	Age	Height (cm)	Mass (kg)	Eth
1	F	27	153.0	60.0	WB	34	F	62	173.0	62.0	WB
2	F	32	159.0	63.0	WB	35	F	50	165.0	—	WB
3	M	30	175.0	65.0	AS	36	M	23	182.0	85.7	WB
4	F	57	153.0	95.0	WB	37	F	52	162.5	52.0	WB
5	M	29	170.0	80.7	WB	38	M	20	180.0	70.0	AR
6	M	27	177.0	108	WB	39	M	52	180.0	113.0	WB
7	F	53	155.0	63.5	WB	40	F	34	165.0	51.0	AS
8	M	27	169.0	80.0	WB	41	M	47	190.5	88.9	WB
9	F	42	169.0	76.2	WB	42	—	—	—	—	—
10	M	40	178.0	77.0	—	43	M	21	182.0	116.0	AR
11	M	37	180.0	70.0	AS	44	F	39	167.6	60.8	WB
12	M	37	171.0	67.0	MR	45	F	40	137.2	133.4	WB
13	F	32	165.0	75.0	MR	46	M	62	177.8	100.7	WB
14	M	21	185.0	95.0	AS	47	F	29	170.2	57.2	WB
15	M	19	181.0	90.0	AS	48	F	21	173.0	61.0	WD
16	M	32	180.0	86.0	AS	49	F	27	160.0	67.1	WB
17	M	55	172.7	90.0	AS	50	F	31	157.5	83.5	WI
18	M	21	175.3	85.0	AS	51	F	39	160.0	53.0	WIT
19	M	30	182.9	75.0	AS	52	F	26	152.4	74.9	WB
20	M	28	183.0	70.0	WB	53	F	27	157.5	99.8	WB
21	M	31	185.0	117.5	WB	54	M	47	177.0	72.0	MR
22	M	25	183.0	79.5	WB	55	F	23	157.0	56.0	BBA
23	M	65	165.0	82.5	WB	56	F	51	165.0	92.0	WB
24	F	42	152.0	133.0	WB	57	M	61	174.0	98.5	WB
25	M	25	188.0	77.0	WB	58	F	18	162.5	54.0	WB
26	F	24	162.5	57.0	WB	59	M	36	180.3	85.0	WB
27	M	25	175.0	67.0	AF	60	F	25	162.6	98.0	WB
28	M	40	172.0	95.3	BB	61	F	26	170.2	80.0	WB
29	M	22	172.0	63.0	WB	62	M	30	165.0	66.0	AR
30	M	20	173.0	74.0	WB	63	M	40	180.0	81.0	AF
31	F	45	160.0	69.9	WB	64	M	29	180.0	90.0	WP
32	M	44	167.0	79.0	B	65	M	26	192.0	110.0	MR
33	M	34	177.0	107.0	GR	Summary	M: 37 (57.81%) F: 27 (42.19%)	35.14 (± 12.44) Max: 65 Min: 18	170.83 (± 10.86) Max: 192 Min: 137.16	81.51 (± 20.18) Max: 133.36 Min: 51	WE: 62.5% BE, AR, AS, AF: 37.5%

```

inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000109_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000110_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000111_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000112_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000113_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000114_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000115_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000116_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000117_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000118_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000119_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000120_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000121_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000122_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000123_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000124_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000125_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000126_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000127_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000128_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000129_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000130_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000131_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000132_keypoints.json
inflatig: /content/Data/Participant_9/Experiment_2/E2T4/P9_E2T4_45_json/P9_E2T4_45_000000000133_keypoints.json

```

{ } Variables

Figure 16: Screenshot of json files extracted for participant 9 in google colab

```
[ ] df_all.head()
```

	Participant	Experiment	Task	JSON_Folder		Frame	Keypoints	Target
0	Participant_1	Experiment_1	E1T1	P1_E1T1_45_json	P1_E1T1_45_000000000015_keypoints.json	[[1246.57, 341.367], [1279.87, 382.476], [1268...		1
1	Participant_1	Experiment_1	E1T1	P1_E1T1_45_json	P1_E1T1_45_000000000016_keypoints.json	[[1240.61, 337.582], [1279.83, 382.557], [1262...		1
2	Participant_1	Experiment_1	E1T1	P1_E1T1_45_json	P1_E1T1_45_000000000018_keypoints.json	[[1226.94, 347.246], [1260.23, 382.462], [1260...		1
3	Participant_1	Experiment_1	E1T1	P1_E1T1_45_json	P1_E1T1_45_000000000019_keypoints.json	[[1219.1, 349.217], [1252.41, 382.471], [1258....		1
4	Participant_1	Experiment_1	E1T1	P1_E1T1_45_json	P1_E1T1_45_000000000020_keypoints.json	[[1211.19, 349.188], [1244.62, 382.508], [1250...		1

Figure 17: Dataframe for 2D model training

```

def visualize_skeleton_from_folder(video_path, json_folder, num_frames=10):
    """
    Visualizes skeleton points from JSON files in a folder on corresponding video frames.

    :param video_path: Path to the video file
    :param json_folder: Path to the folder containing per-frame JSON files
    :param num_frames: Number of frames to visualize (default is 10)
    """
    # Open the video
    cap = cv2.VideoCapture(video_path)
    total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    print(total_frames)

    # Select evenly spaced frames
    selected_frames = np.linspace(0, total_frames-1, num_frames, dtype=int)

    fig, axes = plt.subplots(1, num_frames, figsize=(200, 150))

    # Increase display size
    #fig, axes = plt.subplots(1, num_frames, figsize=)

    for idx, frame_idx in enumerate(selected_frames):
        cap.set(cv2.CAP_PROP_POS_FRAMES, frame_idx)
        ret, frame = cap.read()

        if not ret:
            print(f"Skipping frame {frame_idx} (could not read)")
            continue

        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        # Construct JSON filename (Assuming naming pattern: "frame_0001.json")
        json_filename = f"frame_{frame_idx:04d}.json" # Adjust format as needed
        json_path = os.path.join(json_folder, json_filename)

        if os.path.exists(json_path):
            with open(json_path, "r") as f:
                skeleton_data = json.load(f)

            # Assuming skeleton_data contains a list of (x, y) coordinates

```

Figure 18: Function to visualize skeleton from a folder

```

def extract_skeleton_data_all_participants(root_dir):
    """
    Iterates through all participant folders, reads JSON files, extracts (x, y) keypoints (excluding confidence),
    and stores them in a DataFrame. Drops frames where the right or left hip confidence is zero.

    :param root_dir: Path to the dataset directory (e.g., "/content/Data/")
    :return: Pandas DataFrame containing extracted 2D skeleton data for all participants.
    """
    data_list = [] # List to store extracted data before creating DataFrame

    # Iterate through all participants
    participants = sorted(os.listdir(root_dir))
    if not participants:
        print("No participant folders found.")
        return pd.DataFrame() # Return empty DataFrame if no folders exist

    for participant in participants:
        participant_path = os.path.join(root_dir, participant)

        if not os.path.isdir(participant_path):
            continue # Skip non-directory files

        # Extract participant number from folder name (e.g., "Participant_1" → 1)
        try:
            participant_num = int(participant.split("_")[-1]) # Extracts the last part after "_"
        except ValueError:
            print(f"Skipping {participant}: Could not extract participant number.")
            continue # Skip if folder name doesn't match pattern

        print(f"Processing participant: {participant} (Target = {participant_num})")

        # Iterate through experiments
        for experiment in sorted(os.listdir(participant_path)):
            experiment_path = os.path.join(participant_path, experiment)

            if not os.path.isdir(experiment_path):
                continue

```

Variables  Terminal

Figure 19: Extracting skeleton data for all participants

```
[ ] def prepare_lstm_data_ten_participants(df, sequence_length=50):
    """
    Function to prepare LSTM data for only the first ten participants.

    :param df: Pandas DataFrame containing skeleton data.
    :param sequence_length: Number of frames per sequence.
    :return: (X_train, X_test, y_train, y_test) as NumPy arrays for LSTM training.
    """

    # Filter data to only include the first ten participants
    unique_participants = df["Participant"].unique()[:10] # Get the first ten participants
    df = df[df["Participant"].isin(unique_participants)] # Filter the DataFrame to include only those participants

    # Sort data by participant, experiment, task, and frame
    df = df.sort_values(by=["Participant", "Experiment", "Task", "Frame"]).reset_index(drop=True)

    # Initialize lists to store sequences and labels
    all_sequences = []
    all_labels = []

    # Define indices for right hip (8) and left hip (11) in COCO format
    RIGHT_HIP_INDEX = 8
    LEFT_HIP_INDEX = 11

    # Iterate through each row in the DataFrame
    for participant, group in df.groupby("Participant"):
        keypoints_list = group["Keypoints"].tolist() # List of keypoints for each frame
        labels_list = group["Target"].tolist() # List of labels for each frame

        # Initialize lists to store valid frames for this participant
        valid_frames = []
        valid_labels = []

        # Iterate through each frame's keypoints
        for keypoints, label in zip(keypoints_list, labels_list):
            # Ensure the keypoints are in the correct shape: (num_keypoints, 2)
            keypoints = np.array(keypoints).reshape(-1, 2) # Reshape to [num_keypoints, 2] (x, y)

            # Check if there are 25 keypoints (25 * 2 = 50 values)
            if keypoints.shape[0] != 25:
                print(f"Skipping frame with invalid number of keypoints: {keypoints.shape[0]} found, expected 25.")
                continue # Skip this frame
```

Variables  Terminal

Figure 20: prepare lstm for 10 participants 2d

Table 13: Showing frames for all participants

Participant	Number of frames	Participant	Number of frames	Participant	Number of frames
1	7943	23	5602	46	9415
2	8593	24	8589	47	8292
3	7491	25	8844	48	8397
4	8547	26	7160	49	7943
5	7453	27	11124	50	10082
6	10129	28	7592	51	9531
7	8165	29	7240	52	10072
8	8286	30	5625	53	9012
9	8517	31	7131	54	8731
10	4346	32	8705	55	9305
11	8408	33	10607	56	8908
12	5177	34	7345	57	10661
13	7119	35	8145	58	8625
14	5012	36	8530	59	8551
15	4701	37	5788	60	8952
16	5013	38	9266	61	9970
17	5001	39	9879	62	9198
18	4366	40	8327	63	9207
19	5146	41	8021	64	8726
20	7508	43	8891	65	7959
21	8919	44	7636		
22	7237	45	9073		

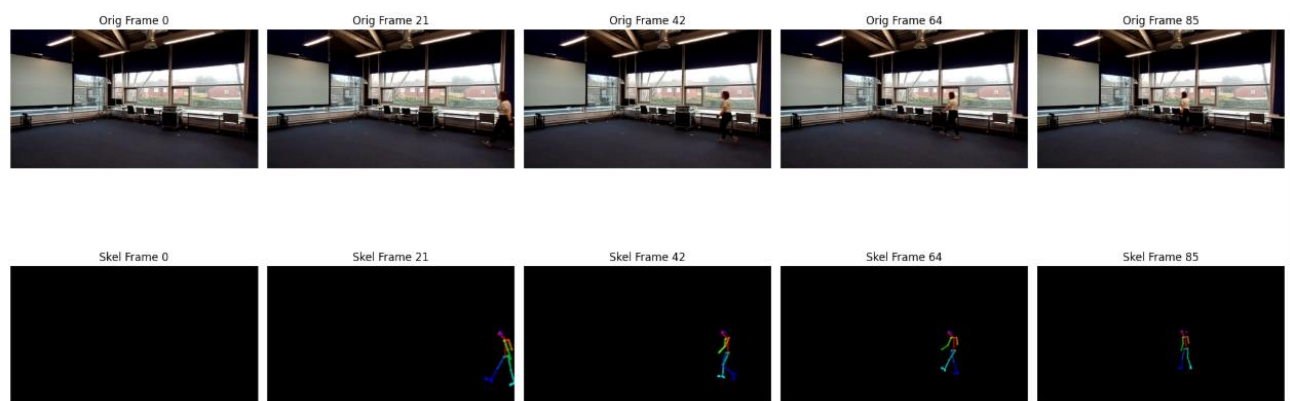


Figure 21: Original vs skeleton frame

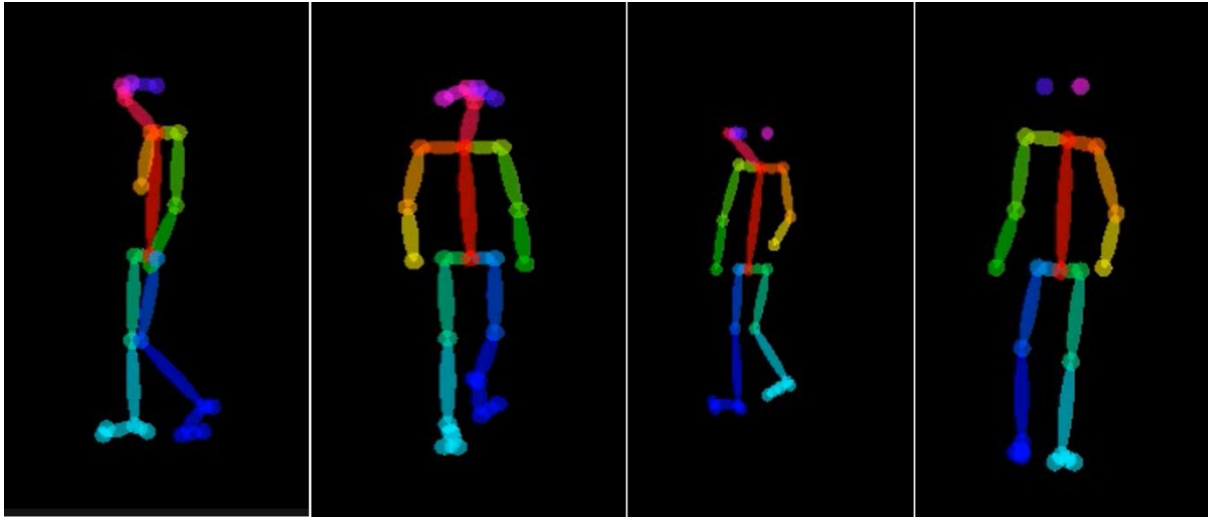


Figure 22: Skeleton frames of participant walking at 90 degrees(camera),front,45 degrees and Rear view respectively.