

**REAL-TIME EXERCISE POSE DETECTION AND FORM CORRECTION USING
COMPUTER VISION**

By

Jeffrey William
12202014

BACHELOR'S DEGREE
in

IT Technopreneurship
Faculty of Engineering and Information Technology

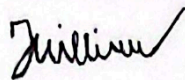


SWISS GERMAN UNIVERSITY
The Prominence Tower
Jalan Jalur Sutera Barat No. 15, Alam Sutera
Tangerang, Banten 15143 - Indonesia

[Revision after the Thesis Defense on 10 June 2026]

STATEMENT BY THE AUTHOR

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in the thesis.



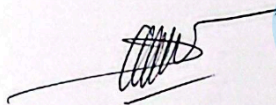
Jeffrey William

Student

18-6-2026

Date

Approved by:

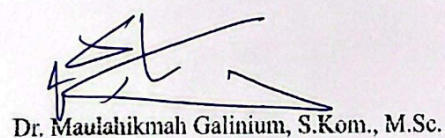


James Purnama, S.Kom, M.Sc

Thesis Advisor

18-6-2026

Date

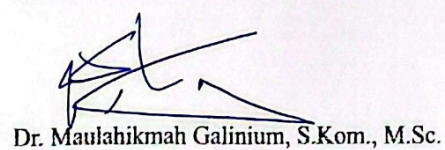


Dr. Maulahikmah Galinium, S.Kom., M.Sc.

Thesis Co-Advisor

18.6.2026

Date



Dr. Maulahikmah Galinium, S.Kom., M.Sc.

Dean

18.6.2026

Date

ABSTRACT

REAL-TIME EXERCISE POSE DETECTION AND FORM CORRECTION USING COMPUTER VISION

By

Jeffrey William

James Purnama, S.Kom, M.Sc., Advisor

Dr. Maulahikmah Galinium, S.Kom., M.Sc., Co-Advisor

SWISS GERMAN UNIVERSITY

Independent exercisers frequently develop incorrect movement patterns without realising it, as most fitness applications track repetitions and duration but do not evaluate movement quality. Existing solutions that provide form analysis either require expensive proprietary hardware or depend on server-side GPU infrastructure. This thesis designed, implemented, and evaluated AIGym, a browser-accessible web application for real-time exercise form correction targeting four foundational bodyweight exercises: squats, push-ups, lunges, and planks. The system runs pose estimation entirely in the browser using TensorFlow.js MoveNet SINGLEPOSE_THUNDER, eliminating server-side GPU resources and reducing total round-trip latency from 200–400 ms to 30–60 ms. A dual-path architecture retains YOLO11 for local offline video testing while the browser-based MoveNet path serves production deployment, both feeding into an identical FastAPI classification backend. Form classification uses biomechanically validated angle thresholds combined with an EvidenceGate temporal classifier requiring six consecutive frames of confirmed error before flagging a variation, eliminating feedback flicker. The system detects up to three named form error variations per exercise across fourteen test scenarios. Experimental evaluation achieved 100% accuracy across all scenarios. User Acceptance Testing with three participants confirmed all primary requirements, and a User Satisfaction Survey with nine respondents yielded a mean score of 3.84 out of 5.0.

Keywords: TensorFlow.js, MoveNet, computer vision, WebSocket, EvidenceGate.

© Copyright 2026

by Jeffrey William

All rights reserved



DEDICATION

To every person in this world who wakes up each day and chooses to invest in their health, despite the demands, the weariness, and the countless reasons not to: this work is for you. The quiet discipline of those who carve out time to stay fit, who push through the struggles of daily life without a trainer or a team beside them, is the very reason this research exists.

To my parents: no words are adequate to express the depth of my gratitude. You gave me life, raised me with love and sacrifice, and provided me with the opportunity to stand at this point today. Every step forward I have taken in this university journey has been made possible by your unwavering support, your belief in me, and the countless ways you have poured into my life. This work belongs to you as much as it belongs to me.



ACKNOWLEDGEMENTS

Above all, I give my greatest thanks to God. It is His strength, His wisdom, and His faithfulness that have carried me to this point. Every moment of clarity in the midst of difficulty, every breakthrough when progress felt impossible, and every resource that arrived at the right time, all of it came from Him. Without God, I am nothing, and this thesis is a testament to what He makes possible.

I wish to express my deepest gratitude to my thesis advisor, Mr. James Purnama, S.Kom, M.Sc., and my co-advisor, Dr. Maulahikmah Galinium, S.Kom., M.Sc., for their patient guidance, thoughtful feedback, and unwavering support throughout this research process. Their direction shaped the rigor and depth of this work, and their willingness to challenge and encourage me in equal measure made me a better researcher.

I also wish to sincerely thank the friends who volunteered their time to test the AIGym application as participants in the User Acceptance Testing phase. Your honest feedback, your willingness to move in front of a camera and be assessed by a system I built, and the insights you shared have contributed in a way that no technical evaluation alone could replicate. This thesis is more complete and more grounded in real experience because of your participation.

TABLE OF CONTENTS

STATEMENT BY THE AUTHOR.....	2
ABSTRACT.....	3
DEDICATION.....	5
ACKNOWLEDGEMENTS.....	6
TABLE OF CONTENTS.....	7
LIST OF FIGURES.....	12
LIST OF TABLES.....	13
CHAPTER 1 - INTRODUCTION.....	14
1.1. Background.....	14
1.2. Problem Statement.....	15
1.3 Research Objectives.....	15
1.4 Research Questions.....	16
1.5 Hypotheses.....	16
1.6 Scope and Limitations.....	17
1.6.1 Scope.....	17
1.6.2 Limitations.....	18
1.7 Significance of the Study.....	19
CHAPTER 2 - LITERATURE REVIEW.....	21
2.1. Theoretical Framework.....	21
2.1.1 Human Pose Estimation.....	21
2.1.2 Pose Estimation Algorithms: YOLO11 and MoveNet.....	22
2.1.3 Browser-Based Machine Learning with TensorFlow.js.....	23
2.1.4 OpenCV for Local Video Processing.....	24
2.1.5 WebSocket-Based Real-Time Communication.....	25
2.1.6 Exercise Form Correction Techniques.....	25
2.1.7 Biomechanics of Bodyweight Exercises.....	26
2.1.8 Supporting Web Framework and Deployment Stack Theory.....	27
2.1.9 Tool Selection and Comparative Analysis.....	29
Table 2.1: Comparison of Pose Estimation Models.....	29
2.1.10 Selection of Exercises.....	31
2.1.11 Agile Software Development Methodology.....	32
2.1.12 Conceptual Framework.....	33
Figure 2.1: AIGym Conceptual Framework.....	34
2.2. Review of Previous Studies.....	36
2.2.1 Chen and Yang (2020) – Pose Trainer: Correcting Exercise Posture Using Pose Estimation.....	36
2.2.2 Khadtare et al. (2025) – Real-Time Machine Learning-Based Posture Correction for Enhanced Exercise Performance.....	37
2.2.3 Mishra (2025) – AI Human Fitness Tracker Using Computer Vision with MediaPipe.....	38

2.2.4	Vigo (2024) – Leveraging Pose Estimation for the Analysis of Human Posture	38
2.2.5	Praveen Kumar et al. (2026) – Real-Time Yoga Pose Correction Using AI	39
2.2.6	Summary of Related Studies	40
	Table 2.2: Comparison of Related Studies	40
2.3.	Research Gap and Novelty	42
2.4.	Comparison with Existing Applications	43
	Table 2.3: Comparison of Existing Applications	43
CHAPTER 3 - RESEARCH METHODOLOGY		47
3.1.	Research Overview	47
	Figure 3.1: Research Methodology Overview	47
3.2.	Requirements Gathering	48
3.2.1	User Interviews	49
	Table 3.1: User Interview Profiles and Primary Requirements	49
3.2.2	Certified Fitness Instructor Consultation	52
	Table 3.2: Certified Fitness Instructor Consultation Summary	52
3.2.3	System Requirements Derived from Stakeholder Feedback	53
3.3.	Literature Review	54
3.4.	Selection and Preparation of Tools	54
3.4.1	System Architecture Overview	55
3.4.1.1	Overall System Pipeline	55
	Figure 3.2: Detailed System Processing Flowchart	57
3.4.1.2	Technology Stack	58
	Table 3.3: Full System Technology Stack	58
3.4.1.3	Pose Model Keypoint Index Reference	59
	Table 3.4: COCO-17 Keypoint Index Reference (used by both YOLO11 and MoveNet)	60
3.4.1.4	Joint Angle Computation	60
	Figure 3.3: angle_at_b() Code Snippet	61
	Figure 3.4: signed_deviation() Code Snippet	61
	Figure 3.5: Detailed System Architecture Design	62
3.4.2	Dual-Path Architecture: YOLO Local and MoveNet Web	63
	Figure 3.6: USE_YOLO Code Snippet	64
	Figure 3.7: USE_YOLO Code Snippet (2)	64
	Table 3.5: Latency Comparison: YOLO Local Path vs MoveNet Web Path	65
3.4.3	YOLO11 Pose Model	66
3.4.4	Use of Pre-trained Model Without Additional Training	67
3.4.5	OpenCV (cv2)	67
3.4.6	NumPy	68
3.4.7	Python 3.10 and Visual Studio Code	68
3.4.8	React 18.3.1	69

3.4.9 TensorFlow.js 4.22.0 and MoveNet.....	69
Figure 3.8: MoveNet model initialization and confidence gating from usePoseDetector.js.....	71
Figure 3.9: MoveNet model initialization and confidence gating (2).....	72
Figure 3.10: MoveNet model initialization and confidence gating (3).....	72
Figure 3.11: Browser-side MoveNet processing flow.....	72
3.4.10 FastAPI 0.115.5, Uvicorn 0.32.1, and WebSocket.....	73
Figure 3.12: WebSocket endpoint from backend/main.py.....	74
Figure 3.13: FastAPI WebSocket connection lifecycle.....	75
3.4.11 Vite 5.4.10.....	76
3.4.12 Deployment Stack.....	76
Figure 3.14: Deployment request routing.....	77
3.5. System Development.....	78
3.5.1 Development Methodology.....	78
Figure 3.15: AIGym Agile Development Sprint Cycle.....	80
Table 3.6: Sprint Deliverables.....	81
3.5.2 Exercise Definitions and Form Criteria.....	82
3.5.2.1 Squat.....	82
Table 3.7: Squats Correct Form Metrics.....	83
3.5.2.2 Push-Up.....	84
Table 3.8: Push-Up Correct Form Metrics.....	84
3.5.2.3 Lunge.....	85
Table 3.9: Lunges Correct Form Metrics.....	86
3.5.2.4 Plank.....	87
Table 3.10: Plank Correct Form Metrics.....	88
3.5.3 Detection Logic.....	89
3.5.3.1 EvidenceGate: Temporal Classification Mechanism.....	89
3.5.4 Testing Scenarios.....	90
3.5.4.1 Scenario Structure.....	90
3.5.4.2 Squat Scenarios.....	91
Table 3.11: Squat Scenarios Overview.....	91
3.5.4.3 Push-Up Scenarios.....	91
Table 3.12: Push-Up Scenarios Overview.....	91
3.5.4.4 Lunge Scenarios.....	92
Table 3.13: Lunge Scenarios Overview.....	92
3.5.4.5 Plank Scenarios.....	93
Table 3.14: Plank Scenarios Overview.....	93
3.6. Experimental Testing.....	93
3.6.1 Test Subject.....	94
3.6.2 Camera Setup and Positioning.....	94
3.6.3 Environmental Conditions.....	95

3.6.4 Test Protocol: Form Classification Accuracy.....	95
3.6.5 Test Protocol: YOLO Local vs MoveNet Web.....	96
3.6.6 Test Protocol: Environmental Robustness.....	97
3.6.7 Test Protocol: Cross-Platform Performance.....	98
3.7. System Implementation.....	99
3.7.1 Production Deployment Platform.....	99
3.7.2 Deployment Process.....	100
3.7.3 User Acceptance Testing (UAT).....	100
3.7.3.1 UAT Participants.....	101
3.7.3.2 UAT Cross-Check Protocol.....	101
3.7.4 User Satisfaction Survey.....	101
3.7.4.1 Survey Framework.....	102
3.7.4.2 Survey Structure.....	102
3.7.4.3 Survey Distribution and Collection.....	103
3.8. Chapter Summary.....	103
CHAPTER 4 - RESULTS AND DISCUSSION.....	105
4.1. Results of Development.....	105
4.1.1 Implemented Exercises.....	105
4.1.2 Dual-Path Architecture.....	105
4.1.3 Key System Components.....	106
4.1.4 System Interface.....	107
Figure 4.1: Exercise Selection Screen (Main Dashboard).....	108
Figure 4.2: Exercise Selection Screen (Push-Up Start Screen).....	108
Figure 4.3: Exercise Selection Screen (Lunge Foot Selection).....	109
Figure 4.4: Exercise Selection Screen (Lunge Start Screen).....	109
Figure 4.5: Countdown Screen (Lunge Screen Left Foot).....	110
Figure 4.6: Countdown Screen (Lunge Screen Right Foot).....	110
Figure 4.7: Active Session Screen (Plank Exercise).....	110
Figure 4.8: Session Summary Screen.....	111
4.1.5 Deployment Confirmation.....	111
4.2. Results of Experimental Testing.....	112
4.2.1 Form Classification Accuracy (H1).....	112
Table 4.1: Squat Form Classification Accuracy (MoveNet Web Path, Controlled Environment).....	113
Table 4.2: Push-up Form Classification Accuracy (MoveNet Web Path, Controlled Environment).....	113
Table 4.3: Lunge Form Classification Accuracy (MoveNet Web Path, Controlled Environment).....	114
Table 4.4: Plank Form Classification Accuracy (MoveNet Web Path, Controlled Environment).....	115
Table 4.5: H1 Summary Across All Exercises.....	116

4.2.2 YOLO Local Path vs MoveNet Web Path Comparison.....	118
Table 4.6: YOLO vs MoveNet Classification Agreement (Per Exercise, Controlled Environment).....	118
Table 4.7: Average Processing Latency: YOLO Local Path vs MoveNet Web Path.....	120
4.2.3 Environmental Robustness.....	122
4.2.3.1 Phase 1: Distance Range Testing.....	122
Table 4.8: Distance Range Testing Results (MoveNet Web Path, Correct Form, Normal Lighting).....	122
4.2.3.2 Phase 2: Lighting Condition Testing.....	124
Table 4.9: Lighting Condition Testing Results (MoveNet Web Path, Correct Form, at Optimal Distance D*).....	124
4.2.4 Cross-Platform Performance.....	126
Table 4.10: Cross-Platform Performance Results.....	126
4.3. User Acceptance Testing Results.....	128
4.3.1 UAT Participants.....	128
4.3.2 UAT Cross-Check Results.....	129
Table 4.11: UAT Cross-Check Against Primary Requirements.....	130
4.3.3 UAT Discussion.....	131
4.4. User Satisfaction Survey Results.....	132
4.4.1 Respondent Profile.....	132
4.4.2 Survey Results.....	133
Table 4.12: User Satisfaction Survey Results by Requirement Group (Likert Scale: 1 to 5, n = 9).....	133
4.4.3 Alignment with Original Requirements.....	136
Table 4.13: Requirements Alignment: Section 3.2 vs Survey Results.....	137
CHAPTER 5 - CONCLUSION AND FUTURE WORKS.....	140
5.1. Conclusion.....	140
5.2 Future Works and Development Roadmap.....	143
5.2.1 Phase 1: Core Development (January – June 2026) — Completed.....	143
5.2.2 Phase 2: Immediate Enhancements (July – December 2026).....	144
5.2.3 Phase 3: Moderate Enhancements (January – June 2027).....	146
5.2.4 Phase 4: Advanced Features (July – December 2027).....	147
GLOSSARY.....	149
APPENDIX.....	153
REFERENCES.....	160
CURRICULUM VITAE.....	163

LIST OF FIGURES

Figure	Page
Figure 2.1: AIGym Conceptual Framework.....	34
Figure 3.1: Research Methodology Overview.....	47
Figure 3.2: Detailed System Processing Flowchart.....	57
Figure 3.3: angle_at_b() Code Snippet.....	61
Figure 3.4: signed_deviation() Code Snippet.....	61
Figure 3.5: Detailed System Architecture Design.....	62
Figure 3.6: USE_YOLO Code Snippet.....	64
Figure 3.7: USE_YOLO Code Snippet (2).....	64
Figure 3.8: MoveNet model initialization and confidence gating from usePoseDetector.js....	71
Figure 3.9: MoveNet model initialization and confidence gating (2).....	72
Figure 3.10: MoveNet model initialization and confidence gating (3).....	72
Figure 3.11: Browser-side MoveNet processing flow.....	72
Figure 3.12: WebSocket endpoint from backend/main.py.....	74
Figure 3.13: FastAPI WebSocket connection lifecycle.....	75
Figure 3.14: Deployment request routing.....	77
Figure 3.15: AIGym Agile Development Sprint Cycle.....	80
Figure 4.1: Exercise Selection Screen (Main Dashboard).....	108
Figure 4.2: Exercise Selection Screen (Push-Up Start Screen).....	108
Figure 4.3: Exercise Selection Screen (Lunge Foot Selection).....	109
Figure 4.4: Exercise Selection Screen (Lunge Start Screen).....	109
Figure 4.5: Countdown Screen (Lunge Screen Left Foot).....	110
Figure 4.6: Countdown Screen (Lunge Screen Right Foot).....	110
Figure 4.7: Active Session Screen (Plank Exercise).....	110
Figure 4.8: Session Summary Screen.....	111

LIST OF TABLES

Table	Page
Table 2.1: Comparison of Pose Estimation Models.....	29
Table 2.2: Comparison of Related Studies.....	40
Table 2.3: Comparison of Existing Applications.....	43
Table 3.1: User Interview Profiles and Primary Requirements.....	49
Table 3.2: Certified Fitness Instructor Consultation Summary.....	52
Table 3.3: Full System Technology Stack.....	58
Table 3.4: COCO-17 Keypoint Index Reference (used by both YOLO11 and MoveNet).....	60
Table 3.5: Latency Comparison: YOLO Local Path vs MoveNet Web Path.....	65
Table 3.6: Sprint Deliverables.....	81
Table 3.7: Squats Correct Form Metrics.....	83
Table 3.8: Push-Up Correct Form Metrics.....	84
Table 3.9: Lunges Correct Form Metrics.....	86
Table 3.10: Plank Correct Form Metrics.....	88
Table 3.11: Squat Scenarios Overview.....	91
Table 3.12: Push-Up Scenarios Overview.....	91
Table 3.13: Lunge Scenarios Overview.....	92
Table 3.14: Plank Scenarios Overview.....	93
Table 4.1: Squat Form Classification Accuracy (MoveNet Web Path, Controlled Environment).....	113
Table 4.2: Push-up Form Classification Accuracy (MoveNet Web Path, Controlled Environment).....	113
Table 4.3: Lunge Form Classification Accuracy (MoveNet Web Path, Controlled Environment).....	114
Table 4.4: Plank Form Classification Accuracy (MoveNet Web Path, Controlled Environment).....	115
Table 4.5: H1 Summary Across All Exercises.....	116
Table 4.6: YOLO vs MoveNet Classification Agreement (Per Exercise, Controlled Environment).....	118
Table 4.7: Average Processing Latency: YOLO Local Path vs MoveNet Web Path.....	120
Table 4.8: Distance Range Testing Results (MoveNet Web Path, Correct Form, Normal Lighting).....	122
Table 4.9: Lighting Condition Testing Results (MoveNet Web Path, Correct Form, at Optimal Distance D*).....	124
Table 4.10: Cross-Platform Performance Results.....	126
Table 4.11: UAT Cross-Check Against Primary Requirements.....	130
Table 4.12: User Satisfaction Survey Results by Requirement Group (Likert Scale: 1 to 5, n = 9).....	133
Table 4.13: Requirements Alignment: Section 3.2 vs Survey Results.....	137