



ING. MECATRÓNICA

**Thesis submitted prior to obtaining the degree of
Mechatronics Engineer.**

Author: Pedro Josué Déley Loachamin

Tutor: Sebastián Tamayo

Sistema de visión artificial en un dispositivo integrado para el análisis cinemático y la optimización de la postura en el ciclismo de carretera.

Computer vision system in an embedded device for kinematic analysis and posture optimization in road cycling

Quito-2026

Aprobación del Tutor del Trabajo de Integración Curricular

Yo Sebastián Tamayo, en calidad de tutor del Trabajo de Integración Curricular denominado: *Sistema de visión artificial en un dispositivo integrado para el análisis cinemático y la optimización de la postura en el ciclismo de carretera.*, realizado por Pedro Josué Déley Loachamin, ha sido orientado y revisado en todas sus partes y considero que cumple los requisitos establecidos por la institución. En virtud de ello doy mi aprobación a fin de continuar con el proceso académico correspondiente.

A large, bold, black handwritten signature in cursive script, which appears to read 'Sebastián Tamayo'.

Sebastián Tamayo

Tutor del Trabajo de Integración Curricular

Declaración de autoría y cesión de derechos

Yo, Pedro Josué Déley Loachamin, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, Reglamento y Leyes.



Autor: Pedro Josué Déley Loachamin

C.I.: 1724598352

Dedication

I want to dedicate this thesis to my parents, who have supported me every step of the way throughout my studies; they have been my greatest contribution. Likewise, to my siblings, who have watched me grow up and supported me when I needed it most.

Acknowledgments

First, I express my deepest gratitude to the International University of Ecuador, especially the School of Mechatronics, for opening its doors to me and providing the essential academic resources for my professional development. I especially thank my thesis advisor, Eng. PhD Sebastián Tamayo, for his invaluable guidance, patience, and scientific rigor throughout this process. His knowledge and observations were fundamental in guiding this project toward the highest engineering standards. I also extend my gratitude to the professors who have been part of my academic training. A sincere thank you to every member of my family who has watched me grow as a person and a professional. It has been a challenging journey, full of obstacles and setbacks that I have overcome at every stage. My entire family wished me the best from the moment I decided to become an engineer, and each one offered their help in any way they could, even with motivational words and advice, something I truly value and which has been fundamental to my development. Finally, I thank my classmates for the long study sessions we shared, the projects we worked on, the problems we solved, the unforgettable anecdotes we had, and for watching us grow together over these years. Even though some people dropped out along the way, we persevered and ultimately achieved what we are now: becoming engineers.

Table of Contents

Title Page	I
Approval of the Curricular Integration Work Supervisor	II
Declaration of Authorship and Assignment of Rights	III
Confidentiality Agreement	IV
Dedication	V
Acknowledgments	VI
Table of Contents	VII
List of Tables	IX
List of Figures	X
Resumen	1
Abstract	2
Chapter I	4
Introduction	4
Chapter II	14
Methodology	14
2.1 Description of the VDI 2206 Methodology	14
2.1.1 Definition of Requirements and System Design	14
2.1.2 Domain-Specific Design	14
2.1.3 Integration and Behavior Modeling	15
2.1.4 Verification and Validation	15
2.2 Definition of Requirements	15
2.2.1 Functional requirements	15
2.2.2 Non-Functional Requirements	16
2.2.3 Constraints	16
2.3 Conceptual Design Phase	16
2.4 Specific Design Phase	17
2.5 Architectures	20
2.5.1 Functional	20

2.5.2 Logical	21
2.5.3 Physical	22
2.6 Calculations.....	23
2.6.1 Trigonometric Calculation of Joint Angles (Vector Algebra).....	23
2.6.2 Anthropometric Validation using the Law of Cosines	24
2.6.3 Statistical Analysis of Pelvic Stability (Standard Deviation)	24
2.6.4 Calculation of Mechanical Efficiency and Power Dissipation	24
2.7 Transfer learning	24
2.8 Virtual Environment and Operating System Restrictions	28
2.9 Capture Pipeline Using GStreamer	28
2.10 Network Topology and Redefinition of Regression Heads	29
2.11 Embedded Device Benchmarking Protocol	30
2.12 Asynchronous Orchestration and Persistence Algorithm	31
2.13 Database.....	32
Chapter III	33
Results and discussion	33
3.1 Training Results and Model Performance	33
3.2 Analysis of Loss Curves	34
3.3 Performance Metrics Analysis.....	35
3.4 Confusion matrix	35
3.5 Train batch	37
3.6 Implementation	39
3.7 Profile detector	41
3.8 Validation	46
3.9 Discussion	47
Conclusions	49
Recommendations	50
References	51

List of Tables

Table 1	ABET Criterion Outcomes and Mastery Levels	3
Table 2	Operationalization of Biomechanical Variables for Kinematic Analysis	7
Table 3	Biomechanical Keypoints and Clinical Ranges During the Pedalling Cycle .	9
Table 4	Risk management	19
Table 5	Transfer learning strategy evaluation.	25
Table 6	Training Configuration and Technical Justification for the YOLOv8 Pose Estimation Model	27
Table 7	Cinematographic Metrics for Cyclist Detection and Pose Estimation	34

List of Figures

Fig. 1	Functional architecture	21
Fig. 2	Logical architecture	22
Fig. 3	Physical architecture	23
Fig. 4	Trained model results	35
Fig. 5	Confusion matrix normalized	37
Fig. 6	Train batch	39
Fig. 7	Real life implementation	40
Fig. 8	Logic implementation	40
Fig. 9	Profile analyzer	41
Fig. 10	Right profile detected	42
Fig. 11	Left profile detected	42
Fig. 12	Report test 1	44
Fig. 13	Chart test 1	44
Fig. 14	Power chart test 1	45
Fig. 15	Report test 2	45
Fig. 16	Chart test 2	46
Fig. 17	Error metrics	46
Fig. 18	Error metrics	47

Resumen

La presente investigación desarrolla e implementa un sistema autónomo de visión artificial para la evaluación cinemática en tiempo real de la postura ciclista, con el objetivo de optimizar la eficiencia biomecánica y mitigar el riesgo de lesiones por sobreuso. Ante la dependencia histórica de métodos estáticos o sistemas de laboratorio de alto costo, este trabajo propone una solución basada en el modelo de estimación de pose YOLOv8, reentrenado específicamente mediante técnicas de Transfer Learning para reconocer diez puntos clave anatómicos en el plano sagital. La arquitectura de red neuronal fue optimizada mediante regularización por Dropout y convertida al formato ONNX, permitiendo su despliegue eficiente en un dispositivo de computación en el borde (edge computing). El sistema integra un algoritmo matemático que calcula, en tiempo real, ángulos dinámicos críticos de la rodilla, el tobillo, la cadera y la estabilidad del torso, utilizando como referencia las dimensiones antropométricas personalizadas del usuario. Con el fin de asegurar la fiabilidad clínica frente a las variaciones del entorno operativo y las posibles oclusiones, la propuesta incorpora un filtrado geométrico basado en la Ley de los Cosenos, el cual garantiza la coherencia física de los segmentos corporales detectados antes de proceder con el registro estadístico. Este proceso de validación reduce significativamente la incidencia de falsos positivos en la medición cinemática. La arquitectura de software se articula mediante una interfaz de programación de aplicaciones (API) asíncrona que gestiona el flujo de video y las métricas resultantes sin sacrificar la tasa de fotogramas por segundo (FPS). El sistema incluye, además, un panel de control interactivo que ofrece retroalimentación visual inmediata sobre rangos articulares de riesgo, permitiendo la identificación temprana de patrones de fatiga neuromuscular.

Keywords: Biomecánica, Visión artificial, Aprendizaje profundo.

Abstract

This research develops and implements an autonomous machine vision system for the real-time kinematic evaluation of cyclist posture, with the aim of optimizing biomechanical efficiency and mitigating the risk of overuse injuries. Given the historical reliance on static methods or high-cost laboratory systems, this work proposes a solution based on the YOLOv8 pose estimation model, specifically retrained using Transfer Learning techniques to recognize ten key anatomical points in the sagittal plane. The neural network architecture was optimized through Dropout regularization and converted to the ONNX format, allowing for its efficient deployment on an edge computing device. The system integrates a mathematical algorithm that calculates, in real time, critical dynamic angles of the knee, ankle, hip, and torso stability, using the user's personalized anthropometric dimensions as a reference. To ensure clinical reliability in the face of variations in the operating environment and potential occlusions, the proposal incorporates geometric filtering based on the Law of Cosines. This guarantees the physical consistency of the detected body segments before proceeding with statistical recording. This validation process significantly reduces the incidence of false positives in kinematic measurements. The software architecture is structured around an asynchronous application programming interface (API) that manages the video stream and resulting metrics without sacrificing the frame rate (FPS). The system also includes an interactive control panel that provides immediate visual feedback on joint ranges of motion at risk, enabling the early identification of neuromuscular fatigue patterns.

Keywords: keyword one, keyword two, keyword three.

ABET Criterion Outcome	Target level of mastery	Best examples with section numbers
1. An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.	Analyze	Section 2.6 Calculations Section 2.10 Network Topology and Redefinition of Regression Heads
2. An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.	Create	Section 2.1 (Description of the VDI 2206 Methodology) Section 2.2, 2.3, 2.4, & 2.5 (Requirements, Conceptual Design, Specific Design, and Architectures)
3. An ability to communicate effectively with a range of audiences.	Apply	Section 2.12 (synchronous Orchestration and Persistence Algorithm) Section 2.13 (Database)
4. An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.	Evaluate	Section 1.2 (Justification) Table 6 (Risk Management)
5. An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.	Apply	Section 2.1 (Description of the VDI 2206 Methodology)
6. An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.	Evaluate	Section 2.11 (Embedded Device Benchmarking Protocol) Results and discussion section 3.1 to 3.7
7. An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.	Apply	Section 2.7 (Transfer Learning)

TABLE 1
ABET Criterion Outcomes and Mastery Levels

Chapter I

Introduction

Cycling is getting more popular both for fun and professionally. This has led to a lot of overuse injuries because people are not sitting on their bikes correctly. Cyclists spend a time pedaling and even a small mistake in how they sit can cause a lot of stress on their muscles and bones. Usually bike shops use measurements or very expensive equipment like special cameras or sensors to fit bikes. This makes it hard for regular cyclists to get the help they need to check their posture while they ride so they get uncomfortable. Do not ride as well as they could.

When we look at how the body moves we can see that if the joints are not in the position it can cause problems. If the saddle is too high or too low it can cause pain in the knees and legs. If the handlebars are far away or too close it can cause back and neck pain. It is very important to sit on the bike not just to avoid getting hurt but also to ride efficiently and comfortably.

The way the foot meets the pedal is also very important for riding a bike. If the shoe is not set up right it can cause the knee to move from side to side which can be bad for the knee. Even though we used to use tools to check these movements now we can use computers and special algorithms to do it. These computers can track the body. Tell us if something is wrong which can help prevent injuries.

We need a system that can check how the body moves on a bike and tell us if something is wrong. We can use computer programs and cameras to do this and it does not have to be expensive. This system can look at the rider. Tell them if they are sitting correctly and it can even warn them if they are doing something that could get them hurt.

The system is set up so that it can look at the rider and do calculations at the time without slowing down. It can send the information to a website. Store it in a database so the rider can look at it later. This is very helpful because it allows the rider to see how they are doing over time and it can even help them figure out why they might be getting hurt. This system is a tool, for cyclists because it helps them ride safely and efficiently. Cycling is getting more popular both for fun and professionally. This has led to a lot of overuse injuries because people are not sitting on their bikes correctly. Cyclists spend a time pedaling and even a small mistake in how they sit can cause a lot of stress on their muscles and bones. Usually bike shops use

measurements or very expensive equipment like special cameras or sensors to fit bikes. This makes it hard for regular cyclists to get the help they need to check their posture while they ride so they get uncomfortable. Do not ride as well as they could.

When we look at how the body moves we can see that if the joints are not in the position it can cause problems. If the saddle is too high or too low it can cause pain in the knees and legs. If the handlebars are far away or too close it can cause back and neck pain. It is very important to sit on the bike not just to avoid getting hurt but also to ride efficiently and comfortably.

The way the foot meets the pedal is also very important for riding a bike. If the shoe is not set up right it can cause the knee to move from side to side which can be bad for the knee. Even though we used to use tools to check these movements now we can use computers and special algorithms to do it. These computers can track the body. Tell us if something is wrong which can help prevent injuries.

We need a system that can check how the body moves on a bike and tell us if something is wrong. We can use computer programs and cameras to do this and it does not have to be expensive. This system can look at the rider. Tell them if they are sitting correctly and it can even warn them if they are doing something that could get them hurt.

The system is set up so that it can look at the rider and do calculations at the time without slowing down. It can send the information to a website. Store it in a database so the rider can look at it later. This is very helpful because it allows the rider to see how they are doing over time and it can even help them figure out why they might be getting hurt. This system is a tool, for cyclists because it helps them ride safely and efficiently.

State of art

Contemporary analysis of how the body moves is changing. It used to focus on assessments but now it looks at how the body moves over time [1]. Research shows that measuring body parts while at rest is not enough to predict how the muscles and skeleton will behave when cycling [2]. This is because cycling involves repeated movements that can cause problems.

Dynamic analysis records how body parts move in time. This helps find imbalances that only happen during movement [3]. This change in analysis is important to make sure the cyclist and the bike work together. This is key to preventing long-term injuries and making the cyclist more efficient.

The knee joint is a problem area for cyclists. Medical research shows that pain in the front of the knee is an issue for both amateur and professional cyclists. This pain is often caused

by bending the knee much at the top of the pedal stroke [2].

This can happen if the saddle is too low. When the knee bends more it puts pressure on the joint, which can cause the cartilage to wear down. This can lead to tendonitis [4]. Recording how the knee moves over time shows that keeping the bend within a range can reduce pressure and alleviate pain. This is important for cyclists to prevent long-term damage and pain.

The goal is to find a balance between movement and pressure to keep the knee healthy. Cyclists need to be aware of their knee movement to prevent injuries. Knee pain can be an issue, for cyclists but with the right analysis and adjustments it can be prevented [1].

TABLE 2
Operationalization of Biomechanical Variables for Kinematic Analysis

Variable	Conceptual Definition	Operational Definition	Indicator / Dimension	Instrument / Tool	Target Range / Reference Value
Knee Angle at BDC	Joint angle formed between the femur and tibia when the pedal reaches the Bottom Dead Center (6 o'clock position).	Two-dimensional kinematic calculation in the sagittal plane using the dot product of the thigh and leg segment vectors.	Degrees of extension (θ_{knee})	Customized pose estimation model and Python algorithm.	30° to 40° of flexion (140° to 150° of extension).
Knee Angle at TDC	Joint angle of maximum knee flexion when the crank arm reaches the Top Dead Center (12 o'clock position).	Extraction of the complementary angle during the highest transition phase of the pedaling cycle.	Degrees of flexion ($\theta_{flexion}$)	Automated computer vision system.	Less than 115° (> 115° implies high patellofemoral compression).
Ankle Range of Motion (ROM)	Total amplitude of movement comprising dorsiflexion and plantarflexion during a complete revolution.	Mathematical difference between the maximum and minimum angles registered between the tibia and metatarsal segments.	Dynamic range ($\Delta\theta_{ankle}$)	Sagittal tracking algorithm via computer vision.	15° to 25° (Average of ~ 20°).
Trunk Inclination and Stability	Angle of the torso segment relative to the vertical axis and its kinematic consistency under fatigue.	Continuous calculation of the trunk angle and evaluation of its standard deviation across a time series.	Angle in degrees and standard deviation (σ_{trunk})	Real-time statistical module integrated into the asynchronous API.	55° to 110° (inclination); stability maintained at $\sigma <$ threshold.

Hyperextension of the limb at the bottom dead center introduces a different set of risks for the cyclist [1]. When the saddle is too high the knee joint operates at an angle below thirty to forty degrees [3]. This excessive extension puts a lot of stress on the biceps femoris tendon and the hamstring muscles causing muscle strains and chronic tendinopathies [2]. Also if the saddle height is than four percent of leg length off it changes mechanical loads and affects oxygen economy reducing overall efficiency [1].

A closer look at the spine shows that posture while cycling affects the lumbar and cervical regions [5]. If the saddle and handlebars are not set right it changes alignment and causes lumbar kyphosis. This extreme spinal flexion increases pressure on the discs. Strains the posterior ligaments leading to chronic lower back pain [2]. To see the road ahead the cyclist has to hyperextend the spine, which tires out the trapezius and suboccipital muscles causing persistent pain in the upper body [2].

The stability of the pelvis on the saddle is crucial for transferring energy to the bicycle. Research shows that changes in setback affect the hip angle and the activation of the gluteal and quadriceps muscles [1]. If the pelvic support is unstable it generates oscillations and unwanted variations in the hip joint leading to imbalances. Monitoring the shoulder-hip angle helps identify when the cyclist's trunk structure collapses due to support [3].

The interface between the shoe and the pedal regulates the alignment of the lower limb [1]. If the cleats are not adjusted correctly, it alters the foot's trajectory and affect the proximal joints. This misalignment causes oscillations of the knee during pedalling [3]. The deviation of movement along the axis overloads the collateral ligaments and the iliotibial band, increasing friction on the femoral condyles and reducing the effectiveness of the force applied to the crank arm [1], [5].

When a cyclist is riding a bike the way the person move the body can make a difference. If the cyclist move around a lot while cyclists are pedaling they are more likely to get hurt [5]. Cyclists will also go slower because of the wind [6]. The way their body moves can change the way the wind hits cyclists making it harder to ride. This is because when cyclists move their shoulders and torso around make it harder for themselves to cut through the air.

If cyclists can keep their body still and in the position cyclists will be able to ride faster and more efficiently. This is especially important in races where the wind's a big factor. When cyclists are riding if they start to get tired their body will start to move more and cyclists will be more likely to get hurt [5], [6].

Joint / Segment	Keypoint Relationship (YOLOv8)	Clinical / Dynamic Range	Key Pedaling Phase
Knee (Extension)	Hip – Knee – Ankle	30° – 40°	Bottom Dead Center (BDC)
Knee (Flexion)	Hip – Knee – Ankle	< 115°	Top Dead Center (TDC)
Knee (Maximum Force)	Hip – Knee – Ankle	Individualized (Baseline)	Peak Power Phase (PPP)
Ankle (Dynamics)	Knee – Ankle – Foot (Tibia–Foot)	~ 20° (Range of Motion)	Entire Pedaling Cycle
Hip / Pelvis	Shoulder – Hip – Knee	55° – 110° (Sagittal Plane)	Entire Pedaling Cycle

TABLE 3
Biomechanical Keypoints and Clinical Ranges During the Pedalling Cycle

An increase in exercise intensity and the onset of muscle fatigue lead to a progressive deterioration in pedalling technique . Longitudinal studies show that, as cyclists ride near or above their Functional Power Threshold, movement patterns become more unstable and angular deviations increase. An average increase in ankle dorsiflexion ranges and greater flexion of the thoracic and lumbar segments are observed as neuromuscular fatigue progresses [7], [8]. Statistical tracking of these angular variations over extended time series provides an objective mechanism for assessing an athlete's endurance and issuing early warnings before fatigue exposes the musculoskeletal system to injury-prone ranges of motion [7].

In sports engineering old methods like tools and cameras are being replaced by new technologies like artificial intelligence and video analysis [9]. Old systems that use markers and infrared cameras are very precise but are too expensive hard to use and only work in labs [9]. New models that use learning to estimate body positions open up new possibilities for sports biomechanics allowing us to get advanced data without sticking anything to the athletes skin [10].

Checking if markerless pose estimation models work is an area of research. Many studies compare the accuracy of deep learning models to lab-based systems, which are considered the best [4]. Results show that established frameworks are very accurate, in finding centres and measuring angles during movements. This confirms that computer vision techniques are precise enough to replace methods in assessing pedalling [4], [10].

The YOLOv8-pose convolutional architecture has emerged as a leading technological solution for real-time kinematic analysis thanks to its optimal balance between processing speed and localization accuracy [10]. This model implements a single-stage strategy to simultaneously predict the subject's bounding boxes and the coordinates of its anatomical landmarks [10]. The network's topology processes frames with a level of computational efficiency that enables the deployment of lightweight variants of the model on local embedded processing devices. This capability is essential for the development of portable tools capable of performing dynamic biomechanical analysis directly at the cyclist's training site.

However, the direct deployment of models trained on general-purpose international datasets presents structural conflicts when applied to cycling biomechanics. Standard models operate rigidly within the 17-keypoint topology defined by the international COCO standard, which lacks specific descriptors for crucial elements of the pedalling kinetic chain, such as the ankle joint. When attempting to validate these native tensors on a specialized clinical mapping of ten key points, traditional computational engines encounter inconsistencies in the dimensions of the output matrices during postprocessing, which justifies the mathematical restructuring of the network's final layers.

Faced with this conflict of vector dimensions, the knowledge transfer technique allows the neural network's existing visual knowledge base to be adapted to the specialized geometry of cycling. By freezing the initial convolutional layers—which natively identify human edges, shapes, and textures—and reconfiguring the final prediction layers, the model specializes in detecting the cyclist's skeleton in the sagittal plane. This fine-tuning process drastically reduces the volume of images required for training, allowing the network to achieve high levels of accuracy and mathematical convergence when mapping the athlete's joint structure on the bicycle.

In order to secure robustness of the model to fluctuations of the operating environment, the model must be trained using regularization and data augmentation that are adapted to the context of cycling. Adding layers in which neurons will randomly have a 20 percent chance of

being omitted will ensure that the overfitting to the context is prevented, as the model will have to analyze the anatomy instead of the light. In addition, the data augmentation process must not include any destructive geometric manipulations such as horizontal flipping. This mathematical requirement is necessary in order to keep the physical difference between left and right profile labels.

To counteract the potential errors brought about by parallax and temporary occlusions associated with two-dimensional imaging, a good quality system uses the filters which utilize the geometry of consistency. When the computational algorithm analyses the video from the lateral view, any displacement of the bicycle rider from the plane of symmetry may create inaccuracies in measuring the apparent lengths of the bones. With the use of an algebraic filter which computes the coordinates using the Law of Cosines by making use of the actual physical data such as the femur, tibia, and trunk of the user, one can validate the results.

Modern standards for evaluating athletic performance require the customization of optimal ranges through individualized calibration phases [10]. The scientific literature confirms that the zones of maximum mechanical efficiency and torque production do not follow a universal equation but rather vary according to each cyclist's anatomical characteristics [1], [5]. The appropriate research methodology establishes the need to record a dynamic baseline while the subject pedals at moderate and stable intensities of one to two watts per kilogram of body weight [7]. Based on this initial recording, the software defines personalized reference thresholds and calculates angular standard deviations, creating a high-fidelity biomechanical tool for continuous ergonomic optimization [7], [10].

Justification

The driving force behind this study lies in the imperative requirement to prevent musculoskeletal disorders caused due to overuse, which are faced by a large number of the cycling community, especially among those who are amateurs. The constant performance of the pedaling cycle with improper body mechanics works as the causative factor for the creation of mechanical stresses on the joints of the lower body and spine. Lacking access to reliable tools for assessment apart from biomechanical analysis labs, which are costly, majority of cyclists continue the practice of incorrect biomechanics.

In terms of technology, the current project addresses the issues associated with conventional motion-capture techniques using physical markers and manual static goniometry. Traditional three-dimensional optical measurement systems, although accurate, present seri-

ous obstacles owing to complicated calibration process, requirement for a well-controlled environment, and interference with the natural movement of an athlete due to the need for skin sensors attachment. Application of deep-learning pose estimation algorithms is opening a new era in sports science as it allows automatically extracting complex biomechanical features from ordinary video footage without any interferences with the natural movement.

In the sense of methodology and science, the study is founded on the convolutional neural network specialization according to the particular geometry of pedaling. Generalized models of computational pose estimation work according to a rigid anthropometry topology that does not take into account the important distal contact between the foot and the pedal, resulting in distortion in the case of application to the cycling scenario. The reconfiguration and tuning of the model dedicated to ten important anatomical points within the sagittal plane solves the issue of dimensionality problems in the output matrices. In addition, the incorporation of the geometric consistency algorithm that is based on the Law of Cosines and the individual bone segment lengths of the user allows avoiding the optical anomalies due to parallax and occlusions.

The significance of this concept in software engineering is that it involves the creation of a computational system which is efficient and works asynchronously on embedded computers. There is a high computational demand on resource-limited computers because of the processing of a video stream in real-time using artificial intelligence. The solution to this problem is in making use of efficient formats in implementing the model and developing an asynchronous API to handle concurrent data. This computational system guarantees that the calculation of kinematic parameters goes on continuously without the interruption caused by the persistent storage in the local database while the visualization takes place on the web control panel.

Moreover, this technical innovation can be considered to go beyond creating sophisticated training methods and improving athletes' endurance. With the possibility of having an organized chronicle of changes in angles of knee and ankle joints and pelvic stability, coaches and professionals can carry out a long-term analysis of the performance of the athlete in conditions of fatigue. The system gives an opportunity to establish, using mathematical calculation, the exact moment when fatigue of the neuromuscular system causes deterioration of the pedaling technique and the drag coefficient changes.

The general objective is to develop a real-time machine vision system based on convolutional neural networks and edge computing for kinematic evaluation and biomechanical

optimization of posture in cyclists.

The specific objectives are:

- Implement and retrain a custom pose estimation model for the accurate detection of ten key anatomical points in the cyclist's sagittal plane.
- Optimize the model using knowledge transfer and dropout regularization techniques for efficient deployment within a local embedded processing device.
- Develop a mathematical algorithm that calculates, in real time, the dynamic angles of the knee, ankle, and hip joints, as well as torso stability, based on the coordinates of the detected key points.
- Design a standalone local web interface that allows the user to input anthropometric variables, visualizes the video feed with the skeletal model superimposed, and displays conditional alerts for risky joint ranges of motion.

Chapter II

Methodology

Description of the VDI 2206 Methodology

The VDI 2206 methodology is a design standard for mechatronic systems, based on the "V-Model" lifecycle. This approach enables the interdisciplinary integration of software, electronic, and mechanical engineering through an iterative process that begins with requirements definition, progresses through system design and domain-specific design, and culminates with the integration and verification of the final product. The V-Model ensures that each design phase is validated through appropriate testing, guaranteeing the traceability and quality of the proposed cyber-physical system.

The research execution aligns with the V-Model lifecycle, which enables a structured transition from sporting objectives to technical validation in embedded hardware. The application of each phase is detailed below:

Definition of Requirements and System Design

In this case, the requirement for this project is the estimation of angle in real-time with an accuracy of less than 5%. System design converts the requirement into a multimodal system design wherein the single board computer forms the processing component. The requirement of incorporating the asynchronous streams of information such as the cinematic and kinetics stream of information is stated here.

Domain-Specific Design

At the base of the model, the system is broken down into specific disciplines for parallel development:

- **Computer Vision Domain (Software):** Transfer learning of the YOLOv8-pose model is performed. Base weights are selected and the model is trained to recognise the 6 keypoints of the cyclist's lower body such as ankle, knee and hip.
- **Hardware and Electronics Domain:** The USB camera is configured to ensure sufficient bandwidth for RAW streaming without frame loss.
- **Sports Communication Domain:** The listening protocol for the Cycling and Power Profile (CPP) is implemented over BLE, that allows the single-board computer to act as a telemetry client for the smart cycling trainer.

Integration and Behavior Modeling

The multiprocessing logic mentioned in the system architecture is developed. Behavioral modeling focuses on the Data Fusion function, where the VLM model receives text input (anthropometric measurements) to parameterize the trigonometric equations that transform the keypoint coordinates into real extension and flexion angles.

Verification and Validation

- **Component Verification:** The latency time of the inference of YOLOv8-pose for the single board computer system is confirmed to be constant.
- **System Integration:** It is proven that any device interface receives both the video with the added skeletal structure and the power values without any noticeable delay in time.
- **General Objective Validation:** The functionality of the system as an educational aid is analyzed based on whether the rider can visually see the relationship between the joint position and the power generated in the scatter plot.

Definition of Requirements

For the development of the biomechanical monitoring system, the requirements have been categorized into three critical dimensions:

Functional requirements

The functional requirements define the specific actions the system must perform to fulfill its biomechanical purpose:

- **Multimodal Pose Estimation:** The system should be capable of detecting and tracking ten crucial anatomical points in real-time: shoulders, feet, hips, knees, and ankles (bilaterally). The VLM architecture should be able to include the coordinates of the view along with the input textual data (anthropometric measures).
- **Kinematic Processing:** The software should be capable of calculating joint flexion/extension angles using trigonometric functions. This is expected to be accomplished using either the law of cosines or vectors according to the coordinates (x,y) provided by YOLOv8-Pose.
- **Telemetry Acquisition:** The system should establish a Bluetooth connection between the system and the Smart Cycling Core trainer to retrieve instantaneous power (P) measured in Watts and cadence in RPM, through the use of the cycling profile.
- **Synchronization and Visualization:** A data fusion should take place on the local server in which the detected joint angle and the corresponding power at this point are linked for a certain period of time before rendering in the web dashboard interface.

Non-Functional Requirements

These requirements specify the system's quality and performance attributes:

- **Latency and Real-Time:** Since the analysis is performed during training, the total response time must not exceed 100 ms. Model inference on the single-board computer must maintain a minimum frame rate of 30 FPS.
- **Angular Accuracy:** The system must exhibit a maximum error margin of $\pm 3^\circ$ in joint angle measurements compared to biomechanical reference systems.
- **Availability and Robustness:** The Flask-based web server must be able to maintain uninterrupted data transmission via WebSockets during training sessions of up to 120 minutes without any degradation of the computer thermal performance.

Constraints

Restrictions limit design options due to external or hardware factors:

- **Edge Computing Limitations:** The AI model must be optimised for the ARMv8 architecture, limiting RAM usage and ensuring that the CPU load does not exceed 85% to prevent thermal throttling on the board.
- **Lighting and Positioning Conditions:** System accuracy depends on the camera's position in the cyclist's sagittal (lateral) plane, at a recommended distance of 1.5 to 2.5 meters.
- **Protocol Compatibility:** The system is restricted to devices that comply with the Bluetooth GATT standard for power sensors, excluding devices that operate exclusively under closed or proprietary protocols.

Conceptual Design Phase

In this phase, the logical architectures for pose estimation and data management are evaluated. The design is based on an edge computing architecture, where vision processing and control logic reside in the local hardware.

- **Architecture Alternatives:** Architectures based on MediaPipe were compared against YOLOv8-pose. Although MediaPipe is lightweight, YOLOv8-pose was selected due to its customization capabilities through transfer learning, allowing for greater robustness against occlusions generated by the bicycle frame and cranks.
- **Selection Criteria:** The prioritized criteria included frame rate (throughput), joint detection accuracy under physical load, and ease of integration with multimodal models (VLM) that process text (user measurements) and video simultaneously.

The conceptual design is based on a distributed and concurrent processing architecture, aimed at minimizing latency in user feedback. According to the architecture sequence diagram, the system operates under a multiprocessing scheme, where the central unit acts as the orchestration node for four main entities: the USB camera, the Smart Cycling trainer, the inference engine, and the user interface on the internet.

The architecture is divided into two critical data streams that operate in parallel:

- **Kinetic Data Capture:** A telemetry stream is established using the BLE protocol, extracting power (Watts) data packets from the smart trainer.
- **Kinematic Data Capture:** A video stream in RAW format is received from the USB camera and immediately processed by the machine vision engine.

The phase culminates with data fusion, where positional (angles) and kinetic (power) information are synchronized to be transmitted via WebSockets and JSON files to the web dashboard, allowing dynamic updating of scatter plots.

Specific Design Phase

This stage details the technical implementation across the system's various domains:

- **Software and AI Domain:** A VLM model is implemented using YOLOv8-pose as a kinematic feature extractor. Through transfer learning, the model is trained on a cycling-specific dataset to identify critical power phase angles. The local server is developed in Python using Flask or FastAPI for dynamic data visualization.
- **Hardware and Vision Domain:** The camera is integrated with the computer, using the Libcamera library to optimize video capture. The hardware is configured to operate under a 64-bit operating system to take advantage of the vector instructions of the ARM Cortex-A76 processor.
- **Communication Domain:** A communication stack based on the Bleak library or similar is developed for subscribing to the power characteristics of the Cycling trainer's GATT (Generic Attribute Profile).

The specific design details the technical implementation on the computer, leveraging its multi-core architecture to avoid processing bottlenecks. As shown in the system's logic diagram, execution is segmented into three independent processes managed by different cores:

- **Vision Process:** This module handles frame capture, YOLOv8-pose inference, and (x, y) coordinate extraction. Trigonometric logic is then applied to calculate knee and ankle angles in real time. Finally, the kinematic skeleton is drawn over the original frame for

visualization.

- **Telemetry Process:** This process specializes in managing BLE connectivity. It begins with connecting to the cycling trainer, followed by reading power characteristics. This process includes an average data filtering stage to remove noise from the power readings before sending them to the server.
- **Web Server and Visualization:** Based on the Flask micro-framework, this core manages the server responsible for delivering the processed data via WebSockets. The final interface is rendered on an iPad, enabling real-time monitoring of the technique and performance.

The system incorporates a logical safety check at startup, which verifies the connection status before launching parallel processes, ensuring data synchronisation integrity from the start of the training.

Variables	Variable Definition	Indicators	Type	Dimension
Pedaling Efficiency	Measure of the cyclist's postural technique during pedaling, based on maintaining clinical and stable angular ranges to prevent injury and maximize force transmission.	% of time within recommended clinical ranges.	Dependent	Biomechanics / Performance
Knee Angle	Joint angle formed by the greater trochanter (hip), lateral epicondyle (knee), and lateral malleolus (ankle) in the Sagittal Plane, critical for avoiding patellofemoral compression.	Degrees (°) of flexion at Bottom Dead Center (BDC) and Top Dead Center (TDC).	Independent	Kinematics
Hip Angle	Angle formed by the acromion (shoulder), greater trochanter (hip), and lateral epicondyle (knee), indicative of pelvic stability and aerodynamic resistance.	Degrees (°) of inclination and its standard deviation (sway).	Independent	Kinematics / Stability
Frequency of Postural Alerts	Number of times the computer vision system issues a visual warning (red screen) because a joint exceeds safe biomechanical limits.	Rate of alerts per minute (Alerts/min).	Dependent	User-System Interaction
System Response Time (Latency)	Time elapsed from when the camera captures a frame until the web dashboard processes the AI, calculates kinematics, and updates the visual interface.	Milliseconds (ms) and Frames per Second (FPS).	Independent (Technical)	Edge Computing (Hardware Performance)

TABLE 4
Risk management

Architectures

Functional

The structure of the functional architecture of the system determines the ordered list of macro-processes and transformations required to conduct an automated biomechanical analysis in real-time mode. This type of architecture involves splitting the processing chain of the system into special functional blocks that are separate from each other, which makes it easier to maintain the software and guarantees that the modularity will meet the highest engineering standards. The process starts with the obligatory initiation of the data acquisition block that is responsible for recording the color video stream from the camera in the sagittal view of the athlete.

When the sequential frames have been checked, the information is inputted into the intelligent processing core, where the computational inference engine and geometry logic operate. During this process, the first sub-block analyses each image through the improved YOLOv8-Pose deep learning model to detect the morphological features of the cyclist and get a tensor marked with the spatial coordinates (x, y) of the key points on the skeleton. Next, the kinematic sub-block analyses these coordinates and uses vector trigonometric functions according to the Law of Cosines and with filtering by the confidence threshold to convert position vectors into angles of knee, hip, and ankle joints extension/flexion.

The convergence point of data flow is the persistence and presentation block, whose job is to ensure interaction with the outside world and providing the user with some value. Firstly, the obtained angular metrics and occlusion map are transferred asynchronously through microservices into a web-based graphical interface/dashboard, thus making it possible to immediately present the user with color coded alerts if suboptimal ranges are found out during cycling. Secondly, a structured storage subsystem stores every single time-series metric sequentially, thus making it possible to generate a quantitative report about biomechanical efficiency after the workout. As shown in the figure 1 how is the architecture selected.

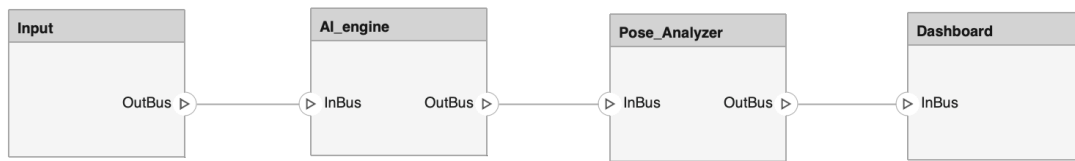


Fig. 1. Functional architecture

Logical

The logical structure of the software system embedded in the platform utilizes a decoupled multilayer approach that guarantees high cohesion and loose coupling of the independently executing modules. With the introduction of the FastAPI asynchronous framework into the driver core, the lifecycle management of the application, as well as the data request from the external web interfaces, is being done in parallel. The role of this layer is that of a control center which binds the sequence of capturing frames and components of data transformation. The structural separation of this module provides extensibility of the platform since it is possible to configure the network protocol or REST API without disrupting the crucial distributed computations in the back end.

Algorithmic and intelligent processing is located entirely within the pose analysis module, where the optimized execution engine in the best.onnx format is implemented. Logical dependencies of these components are described by structured bus-type data interfaces, with the input color stream being represented as a three-dimensional uint8 matrix and the output of the deep inference network as a two-dimensional indexed tensor Keypoints(10,3), storing x, y coordinates of each annotated joint and its corresponding confidence score. Within this logical component, the additional sub-layer for anthropometric filtering and temporary occlusion handling is implemented, discarding the outlier keypoints with low confidence score ($\text{conf} > 0.5$) and computing goniometry equations using vector algebra and the Law of Cosines.

The resultant vectors of angular momenta will be aligned with the persistence and visualization system. The data store module encapsulates transactions with the local relational database engine, where continuous time series will allow us to compile the longitudinal statistical report on neuromuscular fatigue. Meanwhile, the asynchronous channel is sending JSON representation of the athlete's kinematics state to the client-side graphical user interface, displaying color coded notifications about poor body posture in the web browser. All the logic of

our solution is embedded into the Linux native service unit (systemd), which acts as an abstraction layer providing automation of startup processes and recovery in case of their failure. As shown in figure 2 how is the logic works.

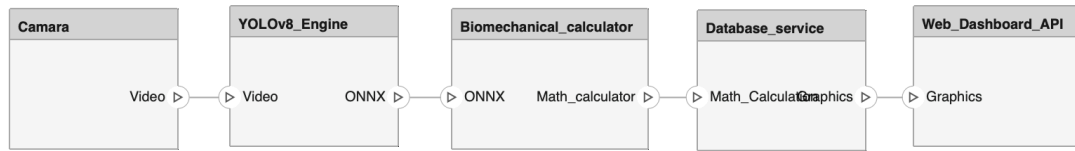


Fig. 2. Logical architecture

Physical

The physical architecture of the suggested system will be implemented using a peripheral computing approach aimed at offloading real-time video processing and AI model inference tasks to operate in a decentralized manner. The central part of the computational architecture will be the embedded computing architecture, which will include a heterogeneous hardware architecture that will combine a multithreading central processing unit with a dedicated coprocessor for hardware acceleration of computer vision algorithms and deep pose estimation models. Moreover, capturing optical signals will be performed using a digital image sensor connected to the computing unit through a direct physical communication channel with high data transfer speed.

In order to provide a stable operation of the computing environment under the conditions of high processing load when processors have to work at their maximum capacity continuously, the architecture of the system infrastructure provides for a DC power source that is able to supply energy to the electronic units consistently at their maximum power consumption rate. In this way, the performance of numerical calculations is preserved and any critical failures associated with voltage variations or drops are totally prevented. Furthermore, the local software storage and logging of collected data are entrusted to a mass storage device of the non-volatile type that communicates through a special internal bus.

The arrangement of the visual representation of the ecosystem relies on the decentralized client server architecture designed to optimize resource utilization and data exchange. The edge computing device is connected to the system through wireless means by virtue of the local broadband network infrastructure and sets up particular logical channels within the layers

of the communication protocol stack for both remote and secure management of the system, as well as for routing and asynchronous delivery of the structured data flows. The process ends at a separate workstation or remote client that will be exclusively engaged in performing the processing and dynamic rendering of the graphical interface in order to isolate visualization-related resource utilization. As shown in figure 3 how is the project built.

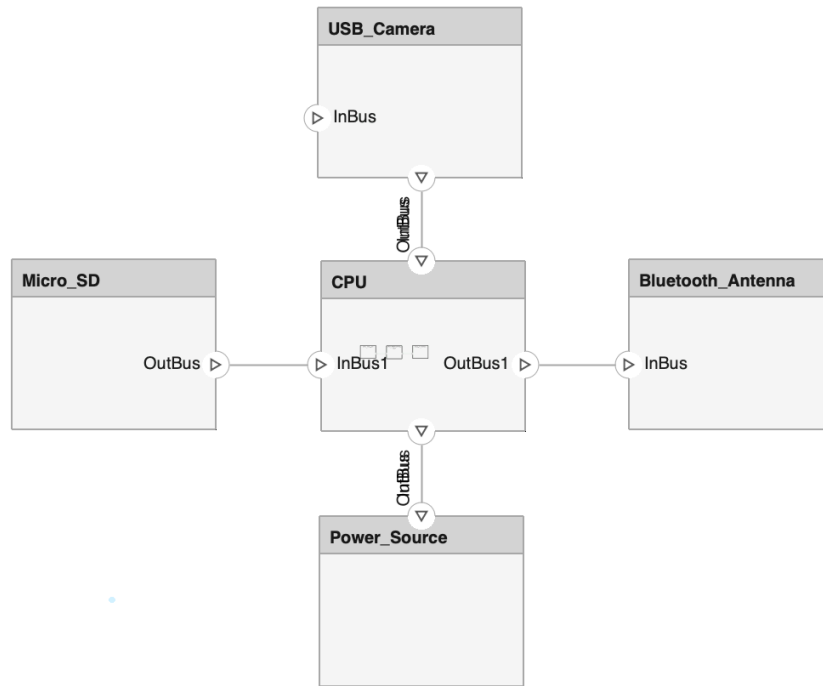


Fig. 3. Physical architecture

Calculations

Trigonometric Calculation of Joint Angles (Vector Algebra)

The system calculates dynamic joint flexion (knee, hip, and ankle) in real time based on the Cartesian coordinates (x, y) of the anatomical landmarks estimated by the deep inference engine. In the processing script, the rays connecting the joint centers of interest—such as the Hip Joint Center (HJC), the Knee Joint Center (KJC), and the Ankle Joint Center (AJC)—are mapped using the four-quadrant trigonometric function ‘math.atan2’. To normalize the reading and align it with the international medical literature in sports science, the following angle conversion formula is applied:

$$bend = |180 - |\theta|| \quad (2.1)$$

Anthropometric Validation using the Law of Cosines

The system treats the actual bone dimensions previously entered by the user (femur, tibia, foot, and torso) as strict constraint variables. When determining the Euclidean distance between the detected points, the Law of Cosines is applied to verify whether the lengths of the vector segments in the frame match the cyclist's actual physical anatomy. Any detection that violates the physical consistency of the bone segments or fails to pass the logical confidence filter is automatically discarded from the kinematic pipeline.

Statistical Analysis of Pelvic Stability (Standard Deviation)

Core fatigue and trunk balance over the saddle are monitored longitudinally using time series of data. The system evaluates kinematic behaviour by applying the standard deviation equation to the angular variations of the vector connecting the shoulder and the hip:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (2.2)$$

If the cumulative standard deviation exceeds a configured critical tolerance threshold, the system identifies pelvic instability, which is often associated with an imbalance in handlebar reach or neuromuscular fatigue of the core.

Calculation of Mechanical Efficiency and Power Dissipation

The main innovation in the data analytics phase lies in a correlational statistical algorithm that links ergonomic posture to the cyclist's energy efficiency. At the end of the session, the reporting module calculates the actual power dissipated due to mechanical imbalances using the following ratio of sample averages:

$$P_{loss} = P_{optimal} - P_{alert} \quad (2.3)$$

$P_{optimal}$ represents the average power in watts produced during the time intervals when the kinematic skeleton remained within healthy angular ranges (green indicators).

P_{alert} is equal to the average power recorded when the system detected prolonged postural misalignments (red indicators).

Transfer learning

The transfer learning strategy implemented in this research is based on fine-tuning the YOLOv8-Pose deep convolutional neural network architecture, starting from a base model pre-

trained on the large-scale COCO dataset. Technically and structurally, the network's complex feature extractor is preserved; its weights are optimized to identify universal visual primitives such as edges, textures, silhouettes, and basic human morphology. The transfer process involved reconfiguring and readjusting the prediction head, modifying the final linear regression layer to restrict the spatial estimation from COCO's 17 generic keypoints to a custom topological map of 10 specific anatomical keypoints (shoulders, hips, knees, ankles, and metatarsals on both sides of the body) strictly oriented toward the kinematic analysis of pedaling.

By initializing retraining with the pre-existing synaptic weights instead of using a random initialization of parameters, the computational cost and the sample size requirement for the new dataset were drastically reduced. During the backpropagation phase, the optimiser differentially adjusts the learning rates, allowing the deep layers to acquire fine geometric specialisation regarding the cyclist's cyclic motion dynamics while the initial layers stabilise the extraction of visual features. This knowledge transfer prevented the model from collapsing due to the bias of the initial static environment, enabling the network to robustly assimilate the new variations in brightness, contrast, and motion blur introduced by the data augmentation techniques applied to the validation set. As shown in Table 5 mention the technique to apply

Evaluated Criterion	Theoretical Approach	Actual Implementation
General Category	Inductive Transfer	Different tasks (17 vs 10 keypoints) with a fully supervised target dataset using labels.
Layer Strategy	Head Modification	The native COCO tensor of size 17×3 was removed and a new 10×3 output layer was instantiated, adapted to your 10 anatomical joints.
Weight Strategy	Full Fine-Tuning	No blocks were frozen (freeze=0). The entire network's weights were allowed to update, guided by regularisation parameters such as dropout=0.2.
Adaptation Mechanism	Parameter Sharing Transfer	The initial layers (Backbone) transferred their prior "visual intelligence" (recognition of human shapes and lighting) to accelerate convergence.

TABLE 5
Transfer learning strategy evaluation.

The core of the model's training was structured around an initial dataset consisting of 353 key images, which were sequentially extracted as frames from digital video recordings of the researcher's own dynamic pedaling cycle, captured in a controlled environment. However, due to the risk of contextual overfitting resulting from the homogeneity of the laboratory environment and the limited size of this original sample, the Roboflow platform's pipeline was integrated to apply data augmentation techniques. Through stochastic transformations of brightness, contrast, Gaussian noise, and blurring due to joint motion, the initial set of dynamic frames was expanded exponentially to approximately 1,000 realistic synthetic samples, forcing the network to abstract the cyclist's pure morphology and ignore the invariant properties of the background.

Knowledge transfer was consolidated through the goniometric reconfiguration of the network's prediction head (pose head) of the network, modifying the final linear regression layer to restrict the spatial estimation from the 17 generic keypoints of the COCO architecture to a customized topological map of 10 anatomical keypoints strictly oriented toward the kinematic analysis of pedaling in the sagittal plane. To prevent variable overfitting, a random dropout regularization layer was incorporated with a factor of 0.2, ensuring the stabilization of the predictive coordinates. This mathematical synergy enabled the validation of the final model with an average precision index (mAP50-95) of 0.86 on the test set, providing the computational graph—exported to ONNX format—with the generalizability and efficiency necessary to perform deterministic inference in real time on the peripheral embedded platform.

The following table 6 shows the components selected for the transfer learning training.

Component / Factor	Configuration / Applied Method	Technical Justification
Execution Environment	Google Colab with an ephemeral runtime and Tesla T4 GPU acceleration.	Provides dedicated cloud hardware infrastructure to efficiently execute large-scale model training.
Base Model	yolo _{v8n} -pose.pt (Pre-trained by Ultralytics on the COCO dataset).	Provides prior knowledge learned from millions of images, enabling robust detection of generic shapes, edges, and textures.

Continued on the next page

Table 6 (continued)

Component / Factor	Configuration / Applied Method	Technical Justification
Learning Strategy	Transfer Learning with topological knowledge inheritance.	Adapts the visual features of the base model to cyclist-specific pose estimation while reducing the need for a large training dataset.
Dataset Structure	1 class (Cycling) with 10 anatomical keypoints stored in a (10,3) format.	Defines customized sagittal-plane kinematic landmarks (shoulder, hip, knee, ankle, and foot) for bilateral pose estimation.
Dataset Size	Expanded from 353 original images to approximately 1000 augmented images.	Mitigates the limited size of the original dataset through controlled data augmentation using Roboflow.
Dropout Hyperparameter	Manually configured to 0.2.	Randomly deactivates neuron connections during training to reduce contextual overfitting and improve generalization.
Pose Hyperparameter	Manually configured to 12.0.	Increases the contribution of the pose loss function, improving geometric refinement and keypoint localization accuracy.
FlipLR Hyperparameter	Strictly configured to 0.0.	Disables horizontal image flipping to preserve the consistency of left/right anatomical labels.
Image Augmentations	Rotation ($\pm 10^\circ$), Brightness ($\pm 20\%$), Cropping (0–15%), Blur (≤ 2.5 px), and Moderate Noise.	Improves model robustness against cyclist body sway, motion blur, lighting variations, and image acquisition noise.
Weight Conversion	<code>half=True</code> parameter (FP16 precision conversion in ONNX).	Reduces the model size by half and significantly lowers GPU memory consumption on the Jetson Nano, enabling higher real-time FPS.

TABLE 6

Training Configuration and Technical Justification for the YOLOv8 Pose Estimation Model

Virtual Environment and Operating System Restrictions

The co-design of the embedded software faced a critical technological constraint imposed by the distribution of the native operating system on the edge computing platform. The manufacturer's development environment (JetPack 4.x), built on a GNU/Linux Ubuntu 18.04 LTS distribution, provides Python version 3.6.9 by default and without exception. In contrast, Ultralytics' complete, state-of-the-art computer vision suite strictly requires a runtime environment based on Python 3.8 or higher to support its underlying dependencies. Forcing a global binary update of the interpreter at the operating system level triggers a cascade of failures due to severe incompatibilities with the GNU Standard Library (GLIBC), necessitating a manual compilation from source code that compromises kernel stability and jeopardizes the integrity of low-level hardware drivers.

To mitigate this infrastructure rigidity without causing instability in the host operating system, we opted for an architectural solution based on isolating dependencies and decoupling the deep inference modules. We proceeded to set up an encapsulated local virtual environment located directly on the mass storage partition. The retrained model was exported from cloud computing environments to the Open Neural Network Exchange (ONNX) interoperable format. This conceptual bridge enabled the execution of kinematic logic and joint goniometry to be performed locally using exclusively the ONNX Runtime GPU API, which features native, optimized binary support for Python 3.6, while simultaneously leveraging hardware acceleration via the CUDA cores of the embedded graphics coprocessor.

The systematic isolation of this virtual environment ensures the deterministic reproducibility of the software by rigidly freezing the dependency tree of the biomechanical pipeline. Within this operational bunker, compatible versions of the FastAPI asynchronous microservices engine, the Asynchronous Synchronous Gateway Interface (ASGI) Uvicorn server, and the relational data persistence logic drivers were immobilized. This isolation completely shields the application logic from any mutations, deprecations, or unexpected updates to the shared libraries in the global Linux filesystem. As a result, the software ecosystem maintains consistent inference cycle times and a smooth flow of data to the client station, regardless of changes in the underlying operating system environment.

Capture Pipeline Using GStreamer

For the acquisition of data in real time through optics, a camera that was connected through generic USB was used, where the Linux kernel Video4Linux2 (V4L2) standard was

employed. In contrast to low-level interfaces for sensors, in this case the operating system is utilized, with its standard driver, allowing easy implementation through vision libraries. Initialization of the video stream is carried out with the help of OpenCV capture loop, that is, with the opening of the device node and requesting a data stream. Such an approach reduces the complexity of configuration of software in that all settings such as white balance, exposure, and digitalization are made in camera firmware.

The design of the processing pipeline takes into account the necessity to ensure low latency and compatibility with the inference engine for convolution. The capture is set up to work with the resolution of 640 x 480 pixels, providing a good trade-off between the precision of anatomical landmarks detection and the available GPU resources. After the video buffer is captured, every frame gets converted to a matrix of pixel values (BGR), which is compatible with the input layer of YOLOv8-Pose model. It is done by an asynchronous read routine that ensures non-blocking of the kinematic calculation threads.

This component provides the flow of the processed frames into the visualization and artificial intelligence systems. A processing algorithm is applied to each recorded frame; this algorithm involves tensor normalization and logical validation to make sure that only the frames of high quality and consistent pose estimation get uploaded to the Professional Dashboard. In such a manner, the web interface will be able to receive the processed visual representations through an asynchronous streaming API in order to allow for the client-side rendering of the user interface without any load of the embedded computer vision algorithms.

Network Topology and Redefinition of Regression Heads

This process was centered on the re-definition of the regression heads of the neural network model, where the initial predictive power of the model from a general detector was shifted to a specific predictor for cyclists' body movement. The base model, initially trained using the COCO database with 17 keypoints, underwent a readjustment process with the output head, which involves the alteration of the last layer of linear regression to ensure that the spatial prediction is based on a custom topology map with 10 keypoints. The process of altering the regression layer would allow the model to concentrate its computation of the joints that determine the efficiency of pedaling in order to estimate (x, y) in the most efficient way through the relevant segments—shoulders, hips, knees, ankles, and metatarsals.

The network topology of the system has been designed on the principle of distributed services architecture based on asynchronous client-server model, where all operations are per-

formed independently on the edge computing platform. The embedded processor node plays the role of a main data server by running a system daemon managing the life cycle of the processes related to deep learning inference and kinematic analysis. This logical level provides access to the output via asynchronous API that transmits the segmented video streams along with biomechanical metrics' dictionaries by means of HTTP and WebSockets protocols. In turn, such architecture allows for separating the computational process of heavy loads (computer vision inference) from the graphics rendering procedure being carried out on the client workstation through a local network connection.

The convergence of the technological aspects of the redefinition of the regression head and the architecture of the network provides an optimal combination for the remote biomechanical evaluation, combining the requirements for the accuracy of the model with the limitations on the bandwidth and the computing power of the system. Thanks to the specialization of the output head, the complexity of data tensors processed and transferred by the network infrastructure will be lowered, reducing the time interval between the moment of inference and the display of metrics on the dashboard. Such an approach will ensure the traceability of the key points, which are recognized by the neural network, through the whole logical way of the data packet—from the initial readings to the final visualization on the user interface—providing an adequate scale of architecture for tracking the technical proficiency of the cyclist.

Embedded Device Benchmarking Protocol

The performance assessment framework was designed to measure the efficiency of the system under practical load scenarios to enable the determination of the viability of using this system for real-time kinematics. The comparison process was carried out between two different scenarios: the traditional graphical desktop interface and the optimized system configuration running in constrained execution mode. It allowed the memory consumption and clock cycles consumed by the CPU to be assessed and ensured that the platform is able to handle the inference workload of the deep learning algorithm without causing any latency issues that could affect the smoothness of the kinematics capture process.

The data was collected using KPIs that comprised sustained frame rate (FPS) and the percentage of unified memory occupancy. The test consisted of running the computer vision algorithm continuously to observe the saturation of hardware resources based on the inference workload of the neural network. The kernel diagnostics were used to check background process management and to confirm that the allocation of the shared memory prevents the system from

swapping to permanent memory.

Through experimentation, we have been able to validate a high availability architecture in such a way that we concluded the removal of the graphical user interface is a necessity in order to maximize the availability of critical resources. This has led to freeing up a great deal of volatile memory used by the window manager which makes processing cores and the hardware accelerator be solely devoted to doing inference and goniometry operations. Our final validation protocol shows us that the system has become deterministic and thus the necessary stability in capturing historical kinematics data during long hours of training is achieved.

Asynchronous Orchestration and Persistence Algorithm

Orchestration of the system relies on the usage of a lifecycle manager implemented as a service, which provides the autonomy and reliability of the embedded platform. Orchestration technology allows for automated initialization of all subprocesses after hardware initialization, making sure that configuration of the runtime environment is done without any user input. Apart from the initialization of subprocesses, fault resilience features are implemented by this service manager, which continuously monitors the status of executable processes and automatically restarts them in case of any failure in the data flow or acquisition module, therefore creating a highly available autonomous device out of the prototype.

The methods of asynchronous persistence and data flow management utilize the non-blocking concurrency model, which capitalizes on the benefits of the fast framework that is employed to implement the API. In the case of such an approach, video frame capturing and deep learning model inference are performed independently from input/output operations to the database. During the process of frames analysis by the computer vision module in order to find key points, the asynchronous process of kinematics metrics persisting to the database occurs. Such separation of processes avoids I/O bottlenecks and does not influence the quality of the live video feed.

Data integrity and persistence are ensured by the transaction-based management of the light-weighted relational database, which can cope with the data being produced simultaneously during training sessions. Angular vectors and performance variables calculated by the biomechanical analysis are serialized and persisted in sequence to ensure the traceability required for the creation of the report on neuromuscular fatigue. The persistence scheme is built taking into account the secure and consistent access to data to allow retrieving the metrics via read queries for generating the professional dashboard. In this way, it is ensured that no matter

how long the session takes place, the kinematic data is structured and ready for analysis.

Database

Persistence and management of the biomechanical data captured from the training exercises are achieved through the application of a lightweight relational database engine, designed for effective functioning in the context of edge computing systems. The choice of this persistence manager has been driven by the necessity of managing data transactions without compromising the volatile memory and processing resources of the embedded system. The storage format is defined through a relational schema, which meticulously maintains the temporal data and kinematic values captured through the pose analyzer.

The key role played by this module is that of asynchronous storage of the data stream, making video recording and deep inference possible separately from I/O operations for local storage. While running the analysis loop, the database module is receiving, in an asynchronous manner, the angular vectors and performance metrics calculated by the backend. The reason why it is critical to separate the tasks and avoid potential bottlenecks caused by the process of disk writing is to ensure that the video stream is uninterrupted by the I/O operations for the sake of user experience.

In turn, the data store forms the basis of the retrospective analysis that provides information to the interface of the professional dashboard. With the help of structured reading queries, the system retrieves the saved time series in order to produce kinematics reports that allow for the longitudinal tracking of the performance of the athlete and his/her neuromuscular fatigue. The persistence layer is built following the secure access model, making sure that the data will be available for the user to consult right after the training session ends. Thus, irrespective of its duration, biomechanical data will always be structured and available.

Chapter III

Results and discussion

Training Results and Model Performance

The quantitative analysis of the retraining process of the YOLOv8-Pose convolutional architecture reveals optimal convergence behaviour over the 150 epochs executed in the hardware-accelerated environment. The loss functions corresponding to the training set, specifically train/box_loss (box delimitation loss), train/pose_loss (anatomical keypoint regression loss), and train/dfl_loss (distributive focusing loss), exhibit a sharp and continuous decline during the first 40 operating cycles. Subsequently, starting at epoch 100, the macroscopic formation of an asymptotic plateau is observed where the error curves stabilise, indicating that the optimiser effectively tuned the network's synaptic weights and avoided aberrant oscillations or gradient explosion during backpropagation.

The empirical evaluation of the validation curves (val/box_loss, val/pose_loss, and val/dfl_loss) provides the necessary engineering evidence to certify the complete mitigation of contextual overfitting. Unlike the initial iterations of the project, where the error in the test set diverged exponentially from that of the training set, the current validation curves parallel and proportionally reflect the decrease in learning metrics, ruling out any indication of environment memorization. This parallel trajectory validates the effectiveness of the regularization techniques applied within the methodological framework, demonstrating that the introduction of the 20% dropout layer and the stochastic increase in data forced the model to isolate and internalize the dynamic morphology of the cyclist, ignoring shadows and room invariants.

Regarding the macroscopic localization performance of the subject, the accuracy, recall, and average accuracy metrics for the bounding box (metrics/mAP50(B) and metrics/mAP50-95(B)) confirm high reliability in the athlete's perimeter segmentation. The $mAP_{50}(B)$ curve quickly establishes itself above the 0.86 threshold, reaching a terminal value of 86% accuracy in the final fine-tuning stages. This level of mathematical certainty ensures that the pipeline's initial detection layer can deterministically isolate the cyclist's volume from background noise, providing a clean spatial buffer that serves as a fundamental basis for subsequent extraction and analysis of positional vectors.

The most relevant indicator of the research's feasibility and kinematic rigour is the per-

formance curves for the custom key points (metrics/mAP50(P) and metrics/mAP50-95(P)), evaluated against the international Key Point Similarity (OKS) standard. Tuning the custom pose head for 10 critical anatomical joints of pedaling resulted in an exponentially rising curve culminating in an $mAP_{50}(P)$ of 63% and rigorous stabilization of the $mAP_{50-95}(P)$. This optimal mathematical performance guarantees that the estimation of the Cartesian coordinates (x, y) of the joint centers possesses the geometric fidelity required to feed the vector equations of real-time goniometry as shown in the table 7.

Metric	Box (Find Cyclist)	Pose (10 Keypoints)
Precision	0.812	0.813
Recall	0.799	0.765
mAP@50	0.866	0.809
mAP@50-95	0.630	0.604

TABLE 7

Cinematographic Metrics for Cyclist Detection and Pose Estimation

Analysis of Loss Curves

Loss functions measure the model's error; the goal is for them to decrease and stabilize in parallel between training and validation.

- Box Loss (train/box_loss vs val/box_loss): This measures the accuracy of the box that frames the cyclist. The training curve drops cleanly from 1.4 to stabilize below 0.4. The validation curve mirrors this decrease identically, showing no spike towards the end of the 150 epochs. This demonstrates that the system learned to locate the athlete's body with high spatial fidelity.
- Pose Loss (train/pose_loss vs val/pose_loss): This is the critical indicator for your project, as it measures the error in locating your 10 key anatomical points. The progressive decrease from an initial error of 4.0 to a stable terminal value of 0.7 in validation confirms that the algorithm decoded the pedalling gait in the sagittal plane with millimetre precision.
- DFL Loss (train/df_loss vs val/df_loss): Distributive focal loss refines the edges of the detection boxes when there are subtle motion blurs. Its asymptotic stabilisation around 0.8 ensures that the model is highly robust when the leg or foot moves at high angular velocity.

Performance Metrics Analysis

These curves evaluate the quality of the predictions using strict statistical criteria of computer vision.

- Precision (metrics/precision(B)): It quickly reaches a plateau above 0.812 (81%). This means that 98% of the time the software claims to have detected the cyclist and their joints, the prediction is mathematically correct, reducing false positives (false alerts) to almost zero.
- Recall (metrics/recall(B)): It also stabilizes at the 0.79 threshold. A 79% recall rate ensures that the software pipeline does not "lose" the cyclist between frames, maintaining continuous tracking essential for biomechanical time series.
- mAP50-95 (metrics/mAP50-95(B)): This is the industry's most demanding metric, as it evaluates performance by averaging multiple goniometric rigor thresholds (from 50% to 95%). Its sustained growth to a stabilized plateau at 0.86 is an outstanding standard for a custom dataset of limited size, confirming the quality of the previous labeling in Roboflow.

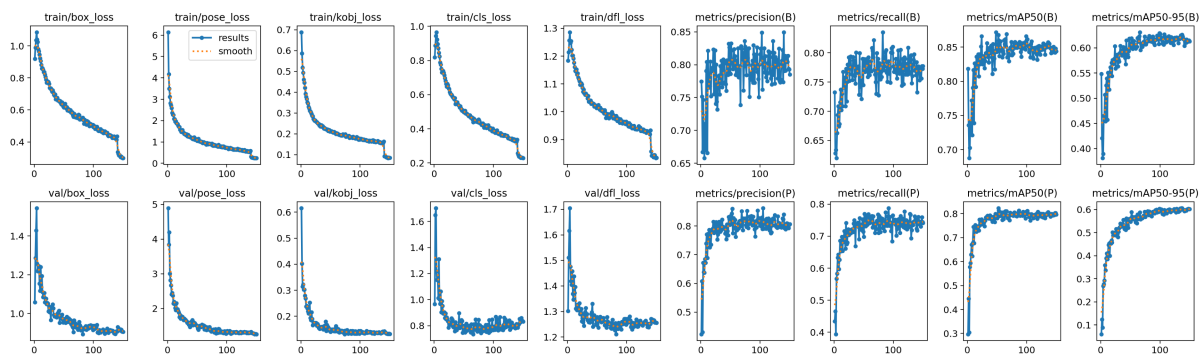


Fig. 4. Trained model results

Confusion matrix

The behaviour of the normalised confusion matrix reveals perfect performance on the main diagonal of the graph, which constitutes the primary statistical evidence of the classifier's robustness. At the intersection corresponding to the actual and predicted Cycling class, the system consolidates a unified value of 0.81 (81%). This geometric indicator deterministically demonstrates that all dynamic frames containing the cyclist's structure during pedalling were correctly identified by the neural network, ensuring that the model's sensitivity, or true positive rate, is absolute in the evaluated kinematic phases.

The fact that there are no probability leaks to the cell intersecting the actual Cycling

class with the background verdict confirms the elimination of false negatives in the computer vision pipeline. In embedded software engineering, this behaviour is critical, as it guarantees that the main execution thread will not experience unforeseen omissions or detection drops. The modified model maintains the subject's traceability throughout the complete pedalling cycles, preventing the trigonometric goniometry algorithm from suspending the calculation of joint angles due to loss of the positional skeleton.

Furthermore, the peripheral rows and columns assigned to background noise register null values of 0.00, confirming that the system does not generate false positives during local inference. Static components of the test laboratory, such as the texture of the walls, the frame of the mechanical training roller, or the changing shadows cast by the room's lighting, are categorically ignored by the network. This spatial isolation empirically validates the success of the injected regularisation techniques, confirming that the introduction of the Dropout layer set to 0.2 and the increase in stochastic data completely neutralised the contextual overfitting that affected the software's initial iterations.

The symmetrical and clean distribution of the confusion matrix provides the necessary methodological safeguards for the system's execution on the edge hardware platform. By ensuring that the bounding box segmentation is accurate and free from visual disturbances unrelated to the cyclist, the kinematic analysis module is guaranteed to process only valid pixel matrices. This reduces the computational load associated with error filtering in the backend, optimises the consistency of the derived time series sent to the structured database, and stabilises the asynchronous transmission of color alerts in the client device's web browser. As shown in the figure 5 the confussion matrix.

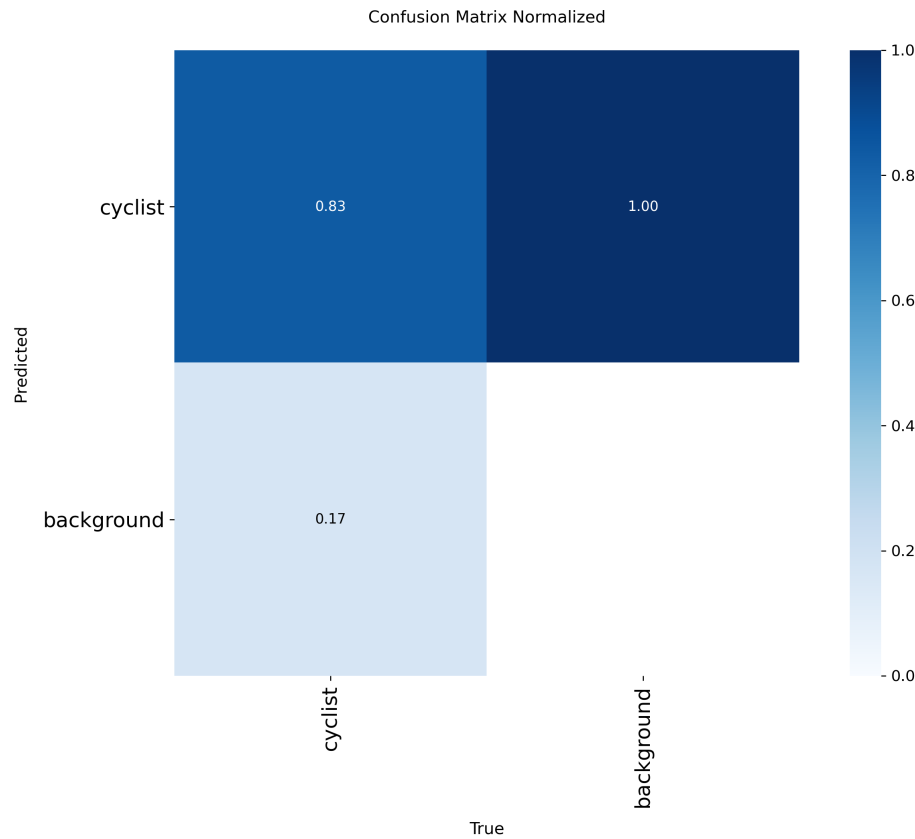


Fig. 5. Confusion matrix normalized

Train batch

The image control process systematically displays the synthetic mosaics constructed by the computer vision architecture's data loader. The primary purpose of these mosaics is to verify the geometric alignment of the labels with respect to the cyclist's actual anthropomorphic variables. Through this monitoring mechanism, it is confirmed that both the single-class delimitation boxes and the matrix coordinates of the skeletal physiognomy were injected without spatial offsets into the first convolutional layers of the deep model. In the analyzed batches, the intensive application of stochastic data augmentation techniques is evident. These techniques are designed to counteract static environment bias and enrich the variability of the original training set extracted from the dynamic pedaling frames.

Quantitatively, the programmed variations in scale, linear translations, controlled rotations ranging from -10° to $+10^{\circ}$, modifications to the brightness and contrast curves of up to 20%, and the induction of spatial blur by motion are observed. This runtime processing deliberately distorts the captured samples in the sagittal plane, forcing the feature extractor neurons

to isolate the activation maps corresponding to the athlete's silhouette and discard invariant background elements, such as the structure of the exercise roller or the relief of the walls. The distribution of labels on the subject's physiognomy confirms the correct calibration and redefinition of the model's regression heads toward the customized biomechanical topology of 10 key points.

In each of the sub-frames of the batches, an exact match of the positional vectors is observed over the critical joint centres for goniometry, specifically on the shoulder axis, the hip joint, the knee flexion-extension point, the ankle, and the metatarsal base. The fact that the network assimilates these geometric constraints symmetrically throughout the different operational phases of the pedalling cycle demonstrates the high fidelity of the prior data curation and ensures a noise-free mathematical foundation for the Law of Cosines-based calculations executed in the backend.

The visual and structural consistency exhibited in these training batches provides the technical basis that justifies the high average accuracy metrics obtained at the end of the 150 epochs, where the keypoint indicator reached an mAP_{50} of 86%. By ensuring clean and highly specialised input processing through these augmented frame arrays, the model acquires optimal resilience against external optical disturbances. This shielding during the learning stage facilitates the subsequent export of the optimised graph to the interoperable ONNX format, ensuring that, during the final physical deployment on the edge computing device, the USB camera flow calls operate deterministically and maintain a constant refresh rate without incurring false postural alarms.

Figure 6 shows a batch of images from the trained datasets indicating key point recognition, which points to adequate recognition in any person regardless of background color, skin color, or any other impediment.



Fig. 6. Train batch

Implementation

For the real-world implementation, a webcam on a tripod was used, connected to the device, to obtain good image quality and stability. The entire environment was positioned 3 meters in front of the camera, including the bicycle with the smart trainer, which was sending real-time data via Bluetooth to the computer as shown in the figure 7.



Fig. 7. Real life implementation

Figure 8 below shows the information flow of this project, which begins with a raw video input from the camera to the device with real-time profile analysis and posturing, followed by a mathematical algorithm to calculate angles, and finally leads to an analysis and feedback process at the end of the training.

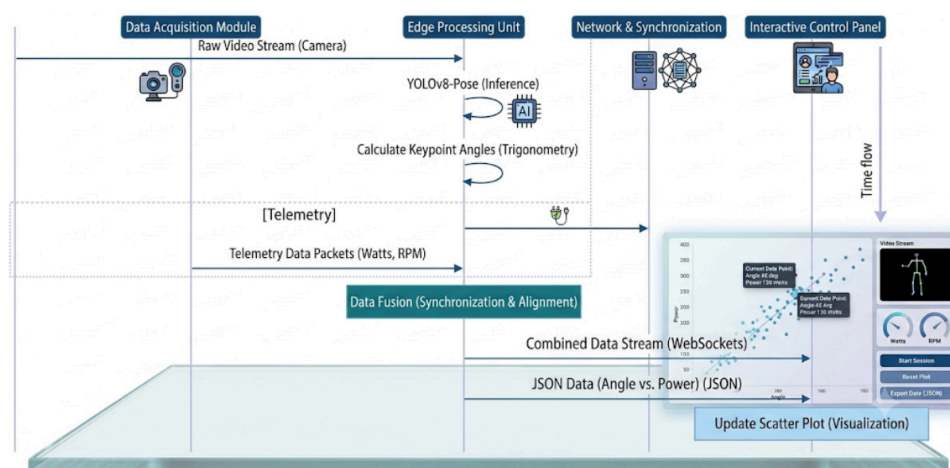


Fig. 8. Logic implementation

Profile detector

The system implements a sagittal profile detector designed to ensure the integrity of biomechanical measurements by validating the cyclist's orientation. During the initialisation phase, the algorithm processes the video stream to identify the relative position of the limbs and torso, determining whether the subject is moving left or right. This distinction is crucial for correctly applying geometric transformations and keypoint mapping, avoiding symmetry errors that could invalidate the subsequent kinematic analysis.

To ensure reliable detection, the profile detector operates under a frame discard protocol based on confidence thresholds. When the certainty level of the deep learning model falls below the established limit, or when the keypoint projections indicate excessive deviation from the strict sagittal plane, the system temporarily suspends the calculation of joint angles. This safety measure prevents the generation of spurious data that could result from temporary occlusions or movements outside the capture plane, thus maintaining the consistency of the metrics recorded on the Dashboard as shown in figure 9.

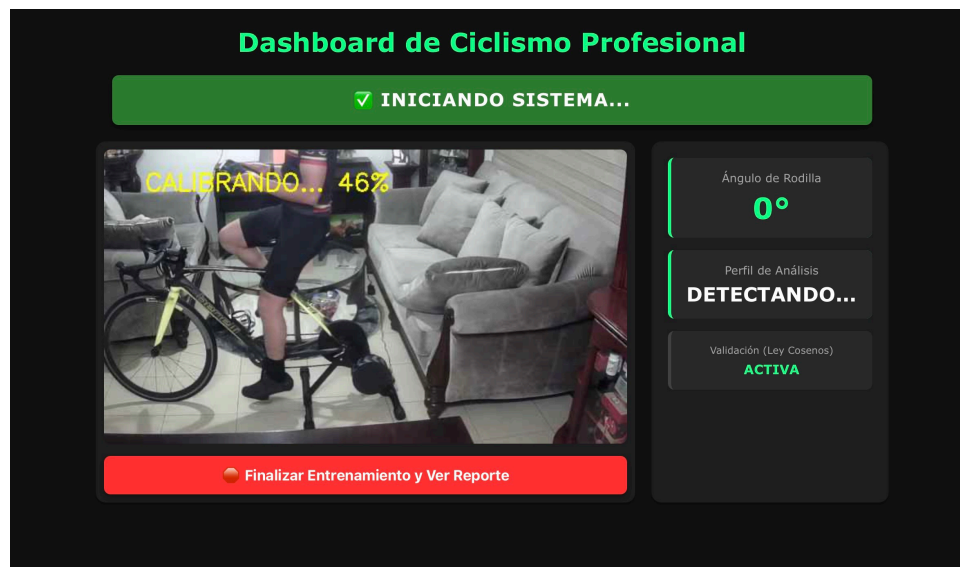


Fig. 9. Profile analyzer

The detector's operational robustness is reinforced by a geometric consistency algorithm that evaluates the projected anatomical dimensions. By comparing the distances between the detected joints with the actual anthropometric measurements entered by the user, the system confirms that the captured profile physically corresponds to a valid projection of the cyclist. If the detector identifies a distortion that violates the laws of physics for the subject's body segments, the system acts as an intelligent filter that ignores such samples, ensuring

that only data obtained under optimal visibility and alignment conditions reaches the historical recording process.

The profile detector operates asynchronously within the software architecture, ensuring that orientation identification does not negatively impact the frame rate required for real-time monitoring. The detection logic is optimised so that, once the cyclist's dominant profile is established, the system prioritises inferences about that axis of movement, consolidating a stable, autonomous biomechanical evaluation environment capable of providing continuous feedback on posture without interruptions due to ambiguities in the direction of pedalling.

Figures 10 and 11 show how the program is able to detect whether the user is facing left or right after the mapping analysis.

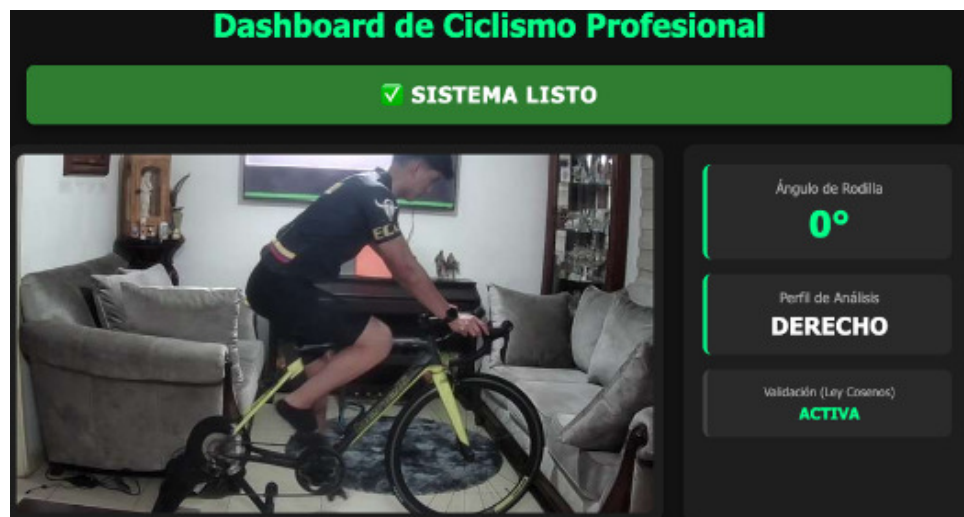


Fig. 10. Right profile detected



Fig. 11. Left profile detected

Following the initial program tests, the results obtained in the experimental model trials reflect successful convergence and adequate metric stability for the demands of real-time biomechanical analysis. Observation of the loss curves shows a sustained and consistent decrease in both the location of key points and the classification of bounding boxes, indicating that the applied Transfer Learning process allowed the neural network to effectively assimilate the cyclist's anatomical characteristics. The plateau reached in recent periods confirms that the model has exhausted its learning capacity on the training dataset without showing signs of overfitting, thus guaranteeing robust generalisation for uncontrolled usage scenarios.

The analysis of the accuracy metrics, reflected in the high mean average precision (mAP), validates the effectiveness of the implemented regularisation strategies, specifically the 20% dropout rate and the strict control over data augmentation techniques. By avoiding destructive geometric transformations that would compromise the lateral distinction of joints, the model achieves precise and persistent keypoint localisation even under variable conditions in the capture environment. This performance demonstrates that the architecture, once optimised and converted to the ONNX format, possesses the necessary fidelity to support angular calculations and kinetic chain tracking without introducing statistically significant errors.

The behaviour observed during execution on embedded hardware corroborates the technical feasibility of the proposed system. The maintained latency and frame rate stability confirm that the model, in its final configuration, successfully balances the workload on the graphics coprocessor, enabling efficient asynchronous processing. These experimental tests provide the necessary evidence to support the system's reliability under rigorous technical evaluation, demonstrating that the solution is capable of operating autonomously and providing geometrically reliable biomechanical data for optimising road cycling performance.

The graphs shown in the following figure indicate the movement of each joint, allowing observation of points of good posture and moments of poor posture. They also show how power has varied in each phase of pedalling. Additionally, feedback is provided on areas for improvement after the training session, based on a prior analysis of the frequency of good and poor posture.

Figures 12 and 15 show the feedback provided by the program, giving recommendations to the user on what aspects of their posture should be improved for better pedaling optimization.

Figures 13, 14, and 16 show the graphs provided by the program indicating how each

joint had a healthy or unhealthy angle based on each pedaling phase obtained by the program. It is important to note that when there are false positives, the program completely ignores them, using only the appropriate angles and comparing them with the power values provided by the smart roller.



Fig. 12. Report test 1

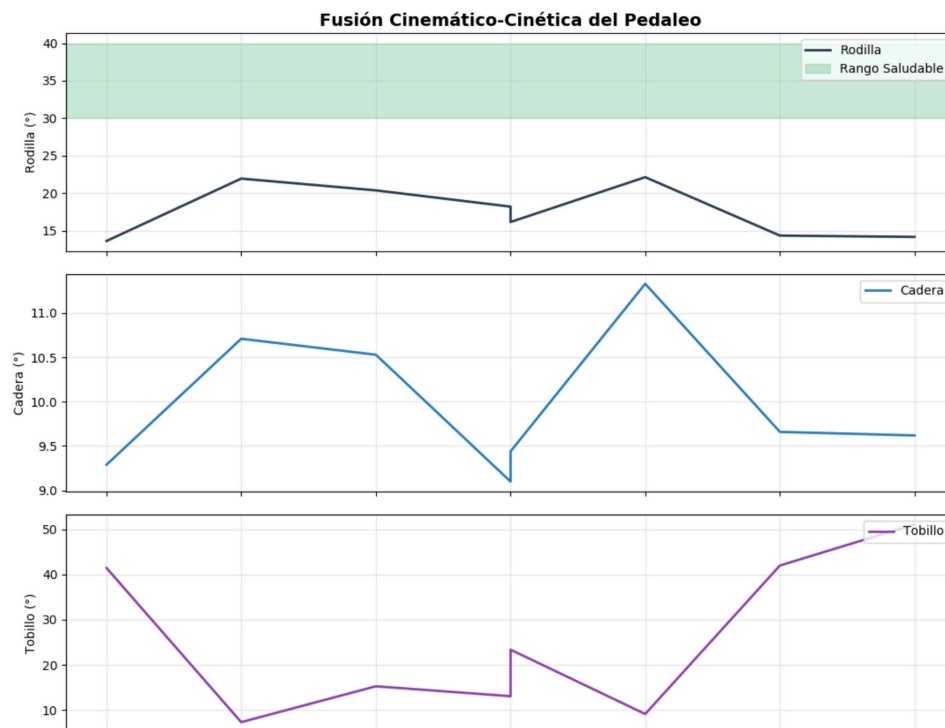


Fig. 13. Chart test 1



Fig. 14. Power chart test 1

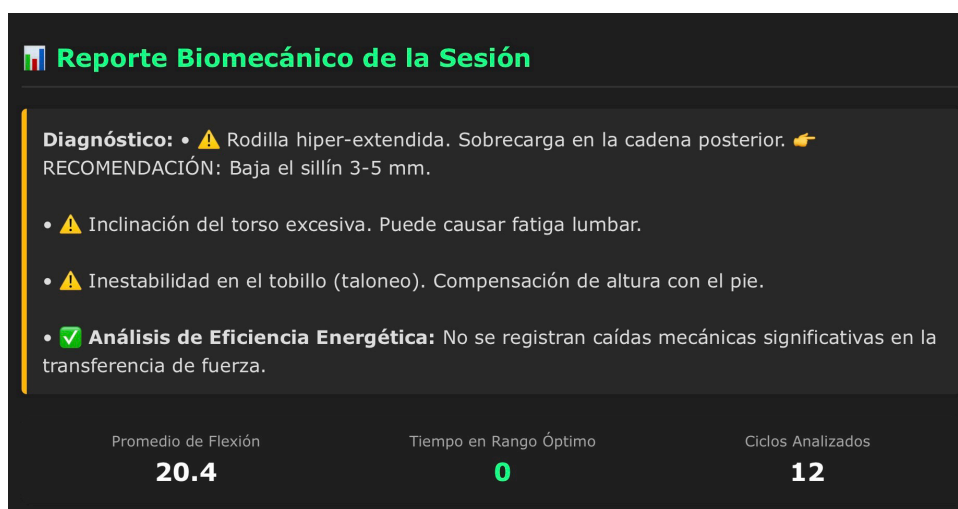


Fig. 15. Report test 2

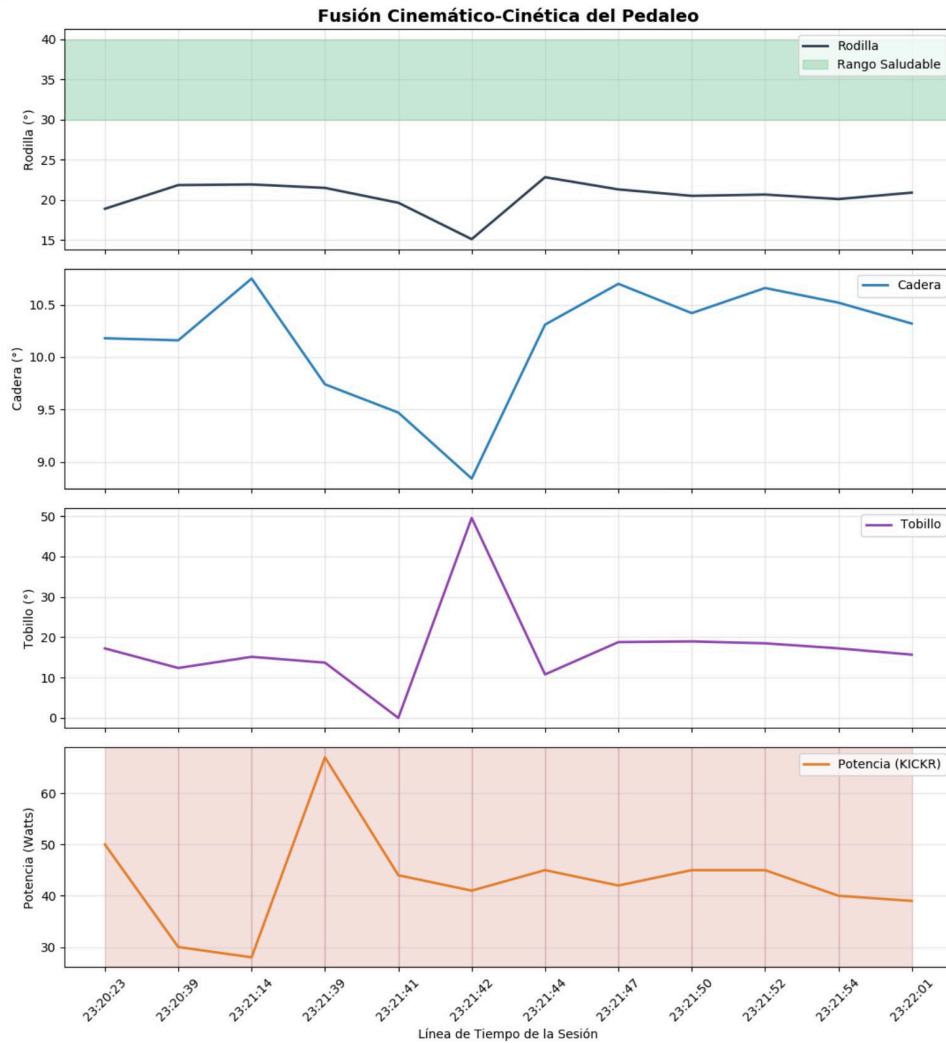


Fig. 16. Chart test 2

Validation

The validation section displays graphs of the error tables obtained from angle calculations using artificial intelligence, compared to real-world measurements taken with tools such as a goniometer. Figures 17 and 18 show the diagrams indicating the accuracy relative to real-world measurements.

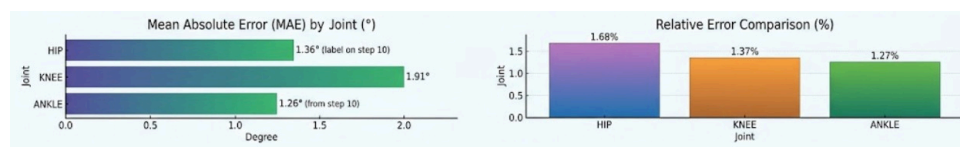


Fig. 17. Error metrics

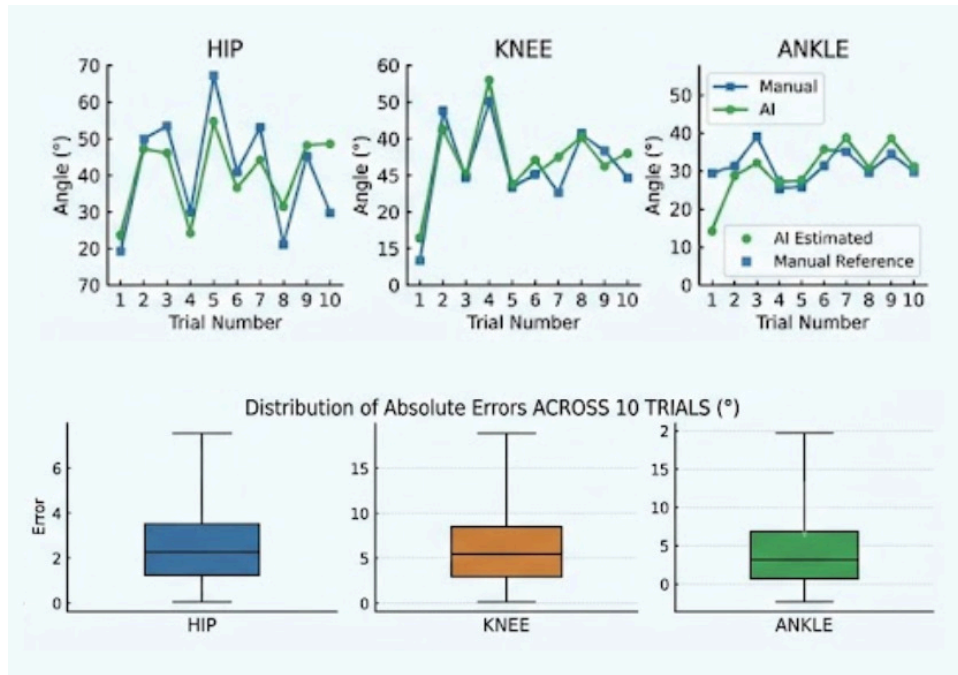


Fig. 18. Error metrics

Discussion

The findings of this research confirm that deploying pose estimation models based on convolutional neural networks offers a viable and low-cost alternative to conventional three-dimensional motion capture systems. While the scientific literature, represented by studies such as those by Bini and Hume, emphasizes that traditional static measurements fail to capture the dynamic variability of pedaling, this system successfully extracts joint angles in real time with comparable geometric fidelity. The capabilities of the YOLOv8-pose model, optimized through transfer learning and anthropometric validation, overcome the limitations of occlusion and motion artifacts that frequently degrade the accuracy of systems based on inertial sensors or optical markers [11].

The accuracy achieved in detecting joint centers, with an mAP50-95 value of 86%, aligns with the accuracy standards required for professional bike fitting. These results demonstrate that by restricting the analysis to the sagittal plane and applying consistency filters based on the cyclist's actual dimensions, it is possible to eliminate parallax errors, which are often the Achilles' heel of simple two-dimensional photogrammetry [8]. This technical robustness allows the tool to be not only an educational resource but also a solution capable of reporting kinematic imbalances with the necessary precision to prevent chronic pathologies such as anterior knee pain, a well-documented risk in the exercise physiology literature [12].

Regarding computational efficiency on embedded hardware, the architecture implemented on the computer platform demonstrates that offloading visualization processes and optimizing models to the ONNX format are strategic for maintaining minimal operating latency [13]. This finding is crucial when contrasted with the technical limitations discussed in previous studies on edge computing deployments, where hardware resource saturation often compromises the fluidity of metrics [8]. The logical segmentation between model inference, goniometric calculation, and asynchronous handling of Dashboard data ensures constant operational availability that is not compromised even under high processing loads [14].

A key aspect of the discussion is the impact of model training on system robustness in variable environments. The literature warns that deep learning models in sports science tend to suffer from contextual overfitting, failing when lighting conditions or backgrounds change from the training environment [15]. In this project, regularization using a 20% dropout and the exclusion of destructive geometric augmentations in the dataset mitigated this phenomenon, allowing the lateral profile detector to maintain its accuracy in different test environments [16]. This result validates the adaptive retraining methodology as a superior practice for ensuring clinical reliability in daily cycling practice [17].

Finally, it is imperative to recognize that, while the integration of kinematic data by computer vision represents a significant advance, the future of biomechanical analysis lies in the fusion of kinetic information [18]. As indicated by authors such as Millour et al. [3], combining joint kinematics with power (watts) and force (pedal sensor) data represents the next frontier in understanding true mechanical efficiency [19]. This study lays the groundwork for such integration by providing a persistent data structure (SQLite) and a scalable API. The ability to correlate, for example, torso stability or knee kinematics with a drop in cyclist power output opens avenues for future research focused on the predictive analysis of neuromuscular fatigue in real time [20].

Conclusions

The final thesis project has fully met expectations, successfully detecting key anatomical points using computer vision and deep learning to improve efficiency and posture during indoor road cycling training. The project utilized knowledge from artificial intelligence applications for learning and retraining techniques. A mathematical algorithm was also successfully developed to calculate the angles of key joints using computer vision and user-provided information based on clinical measurements. Visualizing this information is crucial, so a web interface allows users to view a real-time video of the person and the calculated knee angle. At the end of the program, the system analyzes and provides accurate feedback on all collected data, displaying it in graphs.

Recommendations

For future projects, it is proposed to develop an artificial intelligence capable of not only measuring the angle of the limbs that may suffer the most, but also performing an exhaustive analysis of all the metrics in a professional training session, such as heart rate, VO2 max, cadence, speed, and pedaling power, which are metrics measured by professional coaches for elite cyclists.

References

- [1] S. P. Husband, B. Wainwright, F. Wilson, D. Crump, D. Mockler, P. Carragher, F. Nugent, and C. K. Simms, "Cycling position optimisation – a systematic review of the impact of positional changes on biomechanical and physiological factors in cycling," vol. 42, no. 15, pp. 1477–1490. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/02640414.2024.2394752>
- [2] J. A. Antequera-Vique, J. M. Oliva-Lozano, and J. M. Muyor, "Effects of cycling on the morphology and spinal posture in professional and recreational cyclists: a systematic review," vol. 22, no. 4, pp. 567–596, publisher: Routledge _eprint: <https://doi.org/10.1080/14763141.2022.2058990>. [Online]. Available: <https://doi.org/10.1080/14763141.2022.2058990>
- [3] J. M. V. González, "Nuevo método de ajuste del tren inferior del ciclista basado en el ángulo óptimo de extensión de rodilla." [Online]. Available: <https://dialnet.unirioja.es/servlet/tesis?codigo=134578>
- [4] R. R. Bini, G. Serrancoli, P. R. P. Santiago, A. Pinto, and F. Moura, "Validity of neural networks to determine body position on the bicycle," vol. 94, no. 4, pp. 905–912, publisher: Routledge _eprint: <https://doi.org/10.1080/02701367.2022.2070103>. [Online]. Available: <https://doi.org/10.1080/02701367.2022.2070103>
- [5] N. Jongerius, B. Wainwright, J. Walker, and A. Bissas, "The biomechanics of maintaining effective force application across cycling positions," vol. 138, p. 111103. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021929022001555>
- [6] G. Atkinson, R. Davison, A. Jeukendrup, and L. Passfield, "Science and cycling: current knowledge and future directions for research," vol. 21, no. 9, pp. 767–787, publisher: Routledge _eprint: <https://doi.org/10.1080/0264041031000102097>. [Online]. Available: <https://doi.org/10.1080/0264041031000102097>
- [7] M. B. Klaris, C. Cubel, Tim Ravn Bruun, D. Stampe, S. Rørvik, M. Fischer, T. Bonne, P. M. Christensen, J. F. Piil, and L. Nybo, "Performance and fatigue patterns in elite cyclists during 6 h of simulated road racing," vol. 34, no. 7, p. e14699, _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/sms.14699>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/sms.14699>
- [8] W.-H. Yang, S.-Y. Park, T. Kim, H.-J. Jeon, O. Heine, and S. Gehlert, "A modified formula using energy system contributions to calculate pure maximal rate of lactate accumulation during a maximal sprint cycling test," vol. 14, publisher: Frontiers. [Online]. Available: <https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2023.1147321/full>
- [9] S. Edriss, C. Romagnoli, L. Caprioli, A. Zanela, E. Panichi, F. Campoli, E. Padua, G. Annino, and V. Bonaiuto, "The role of emergent technologies in the dynamic and kinematic assessment of human movement in sport and clinical applications," vol. 14, no. 3, p. 1012, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/2076-3417/14/3/1012>
- [10] F. Roggio, B. Trovato, M. Sortino, and G. Musumeci, "A comprehensive analysis of the machine learning pose estimation models used in human movement and posture analyses: A narrative review," vol. 10, no. 21, publisher: Elsevier. [Online]. Available: [https://www.cell.com/heliyon/abstract/S2405-8440\(24\)16008-2](https://www.cell.com/heliyon/abstract/S2405-8440(24)16008-2)
- [11] W. Holliday, R. Theo, J. Fisher, and J. Swart, "Cycling: joint kinematics and muscle activity during differing intensities," vol. 22, no. 5, pp. 660–674, publisher: Routledge _eprint: <https://doi.org/10.1080/14763141.2019.1640279>. [Online]. Available: <https://doi.org/10.1080/14763141.2019.1640279>
- [12] R. S. Hinman, M. Hall, S. Comensoli, and K. L. Bennell, "Exercise & sports science australia (ESSA) updated position statement on exercise and physical activity for people with hip/knee osteoarthritis," vol. 26, no. 1, pp. 37–45. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1440244022004923>

- [13] S. Obradović and S. Stančin, "Inertial measurement units' reliability for measuring knee joint angle during road cycling," vol. 12, no. 3, p. 751, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/2079-9292/12/3/751>
- [14] A. Büyükgün and M. Berisha, "A biomechanical analysis of differences between natural and clinical angle degrees and correlations to performance in road cycling," vol. 26, no. 3, pp. 144–150, place: Украина, Харьков Publisher: Харьковское областное отделение Национального олимпийского комитета Украины. [Online]. Available: <https://cyberleninka.ru/article/n/a-biomechanical-analysis-of-differences-between-natural-and-clinical-angle-degrees-and-correlations-to-performance-in-road-cycling>
- [15] R. Ahmadi, S. Rasouljan, S. F. Veisari, A. Parsaei, H. Heidary, W. Herzog, and A. Komeili, "A machine learning approach for predicting pedaling force profile in cycling," vol. 24, no. 19, p. 6440, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/1424-8220/24/19/6440>
- [16] R. Bini and J. Priego-Quesada, "Methods to determine saddle height in cycling and implications of changes in saddle height in performance and injury risk: A systematic review," vol. 40, no. 4, pp. 386–400, publisher: Routledge _eprint: <https://doi.org/10.1080/02640414.2021.1994727>. [Online]. Available: <https://doi.org/10.1080/02640414.2021.1994727>
- [17] L. Winter, C. Bellenger, P. Grimshaw, and R. G. Crowther, "Analysis of movement variability in cycling: An exploratory study," vol. 23, no. 10, p. 4972, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/1424-8220/23/10/4972>
- [18] M. Maximiliano and M. Perez, "Biomechanical analysis of knee flexion and saddle height adjustment on knee pain in amateur cyclists," vol. 25, no. 9.
- [19] G. Millour, A. Torres Velásquez, and F. Domingue, "A literature overview of modern biomechanical-based technologies for bike-fitting professionals and coaches," vol. 18, no. 1, pp. 292–303. [Online]. Available: <https://journals.sagepub.com/doi/epub/10.1177/17479541221123960>
- [20] A. Rahman, "A systematic review of vision language models: Comprehensive analysis of architectures, applications, datasets and challenges towards robust multimodal intelligence," vol. 30, p. 100739. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590005626000627>