Digital Twin Coaching for Edge Computing Using Deep Learning Based 2D Pose Estimation

by

Rogelio Gamez Diaz

Thesis submitted to the University of Ottawa

in partial Fulfillment of the requirements for the

Master of Computer Science degree.

School of Electrical Engineering and Computer Science

Faculty of Engineering,

University of Ottawa

Abstract

In these challenging times caused by the COVID-19, technology that leverages Artificial Intelligence potential can help people cope with the pandemic. For example, people looking to perform physical exercises while in quarantine. We also find another opportunity in the widespread adoption of mobile smart devices, making complex Artificial Intelligence (AI) models accessible to the average user.

Taking advantage of this situation, we propose a Smart Coaching experience on the Edge with our Digital Twin Coaching (DTC) architecture. Since the general population is advised to work from home, sedentarism has become prevalent. Coaching is a positive force in exercising, but keeping physical distance while exercising is a significant problem. Therefore, a Smart Coach can help in this scenario as it involves using smart devices instead of direct communication with another person. Some researchers have worked on Smart Coaching, but their systems often involve complex devices such as RGB-Depth cameras, making them cumbersome to use. Our approach is one of the firsts to focus on everyday smart devices, like smartphones, to solve this problem.

Digital Twin Coaching can be defined as a virtual system designed to help people improve in a specific field and is a powerful tool if combined with edge technology. The DTC architecture has six characteristics that we try to fulfill: adaptability, compatibility, flexibility, portability, security, and privacy.

We collected training data of 10 subjects using a 2D pose estimation model to train our models since there was no dataset of Coach-Trainee videos. To effectively use this information, the most critical pre-processing step was synchronization. This step synchronizes the coach and the trainee’s poses to overcome the trainee's action lag while performing the routine in real-time.

We trained a light neural network called “Pose Inference Neural Network” (PINN) to serve as a fine-tuning architecture mechanism. We improved the generalist 2D pose estimation model with this trained neural network while keeping the time complexity relatively unaffected. We also propose an Angular Pose Representation to compare the trainee and coach's stances that consider the differences in different people's body proportions.

For the PINN model, we use Random Search Optimization to come up with the best configuration. The configurations tested included using 1, 2, 3, 4, 5, and 10 layers. We chose the 2-Layer Neural Network (2-LNN) configuration because it was the fastest to train and predict while providing a fair tradeoff between performance and resource consumption. Using frame synchronization in pre-processing, we improved 76% on the test loss (Mean Squared Error) while training with the 2-LNN. The PINN improved the R^2 score of the PoseNet model by at least 15% and at most 93% depending on the configuration. Our approach only added 4 seconds (roughly 2% of the total time) to the total processing time on average. Finally, the usability test results showed that our Proof of Concept application, DTCoach, was considered easy to learn and convenient to use. At the same time, some participants mentioned that they would like to have more features and improved clarity to be more invested in using the app frequently.

We hope DTCoach can help people stay more active, especially in quarantine, as the application can serve as a motivator. Since it can be run on modern smartphones, it can quickly be adopted by many people.

Acknowledgments

Thank you, Brenda, Mamá, Papá, Familia, Friends, Professor El Saddik, Dr. Laamarti, the MCRLab, the University of Ottawa, and Canada.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Networks
CAVE	Cave Automatic Virtual Environment
CNN	Convolutional Neural Network
CPS	Cyber-Physical Systems
DL	Deep Learning
DT	Digital Twin
DTC	Digital Twin Coaching
EC	Edge Computing
KNN	K-Nearest Neighbors
IoT	Internet of Things
MII	Multimodal Interaction Interface
ML	Machine Learning
MSE	Mean Squared Error
NLP	Natural Language Processing
NN	Neural Network
PINN	Pose-Inference Neural Network
POC	Proof of Concept
RGB-D	RGB-Depth
RSO	Random Search Optimization
RT	Real Twin
RTC	Real Twin Coach
RTT	Real Twin Trainee

SVM	Support Vector Machine
TF	TensorFlow
WCF	Weighted Coordinate Function
WNN	Wavelet Neural Network.

1. Introduction

As defined by El Saddik, Digital Twins (DT) are “digital replications of living as well as nonliving entities that enable data to be seamlessly transmitted between the physical and virtual worlds” [1]. This model was introduced by Michael Grieves in 2002 [2] as a part of the Product Lifecycle Management, but it was not known as “Digital Twin” until 2010 by John Vickers of NASA [3].

DT was designed with the idea of creating a virtual representation of any physical entity, such as an aircraft engine or a wind turbine. The DT then uses computational resources to optimize or keep track of the operation in a production environment. Hence, communication is the central pillar of Digital Twins. We need to have real-time data to make instant decisions on the Real Twin (RT), which is the physical entity that we are *digitalizing*. The DT needs to interchange data with its RT continuously and without delays. For this reason, the sensors and the quality of communication channels become an essential part of the DT.

Gartner has classified Digital Twins as an emerging technology, emphasizing DT's integration into peoples' everyday lives, from health to biometric authentication [4]. DTs are currently rising in expectations, expecting to plateau in 5 to 10 years, which will bring an increasing amount of research interest in the following years. Based on a Market Research Future report, the DT technology market is estimated to reach 15 billion dollars by 2025 [5]. Current companies working with Digital Twins include General Electric, Philips, Sisco Systems, Microsoft, Oracle, and IBM [6]. We can see that major players in the world of technology are not staying behind this trend, and we will keep seeing it more and more in the future.

External factors such as the rise in popularity of Machine Learning (ML) in recent years, the ubiquity of smart devices such as smartphones and smartwatches, and the global adoption of

IoT infrastructure are making DT's development a reality for the general population. Bringing Digital Twins to everyday use has positive repercussions regarding privacy, security, usability, availability, etc. We can leverage the power of different technologies and paradigms such as Edge Computing and Deep Learning (DL), to name a few, to provide a complete solution to various fields involving a personal DT.

While this technology has roots in the manufacturing industry, it has made its way to other fields such as health and wellbeing [7]. Scheuermann et al. [8] designed smart clothes with sensors for body temperature, humidity, and heart rate. They had 72 total sensors collecting data to predict stress levels using Random Forest Regression. Changing the approach, Nikitina et al. [9] propose food DT to extract nutrition information and meet a specific consumer's individual needs, applying this new knowledge into recipes, product design, clinical factors, etc. Calderita et al. [10] took Cyber-Physical Systems (CPS) advantages to design an assisted living environment. They integrate Internet-of-Things (IoT), robotics, and AI to add intelligence to caregiving centers and help healthcare assistants make better decisions regarding patients, their medication, therapies, or any other object such as robots inside the CPS.

Another field that DT technology can benefit is virtual coaching, more specifically, sports or physical coaching. Barricelli et al. [11] propose a system called SmartFit, a collection of DTs for health and fitness tracking, from nutrition and sleep hours, to physical training. They provide suggestions to the trainee using Machine Learning (ML) algorithms such as Support Vector Machine (SVM) and K-Nearest-Neighbors (KNN). When applied to coaching, DT's advantages are similar to those described above for manufacturing: optimizing and providing real-time data and feedback between others. It also offers several benefits for coaches. For example, coaches can receive quality data for future routine planning, for immediate feedback to avoid injuries or improve the training. There is also the possibility of working remotely with the trainee's DT as they can be confident that it is a faithful representation of its Real Twin (RT).

Current research involves specialized hardware such as RGB-D Cameras, complex motion sensors, or computational resource-heavy processes. While positive results are reported, mass implementation becomes a challenge. There is a gap in Digital Twin's literature for physical activities in low-resource environments suitable for personal use. This environment has hardware and processing limitations. For example, they could only have a single camera without any other sensors or low-energy efficient chips designed to run for long periods independently.

Edge Computing is another field where research focused on optimizing low-resource consumption models is always welcomed and works exceptionally for Digital Twin Coaching. There are many advances in AI and ML for pose estimation and pose recognition. These two tasks go hand in hand with DT for sports or any other physical activities. One of the main hurdles that AI engineers face is bringing these complex ML algorithms to resource-limited environments such as smartwatches and smart clothing. For example, the best pose estimation algorithms tend to use plenty of computational resources, making them hard to be implemented in the hardware of average users.

This thesis will describe the research methodology we followed in designing and developing a Digital Twin Coach tailored for Edge Computing. For this, we use Deep Learning 2D pose estimation models to track the participants' body parts in real-time and give them immediate feedback on their performance compared to a professional. Our research is focused on physical exercises such as stretching and squatting but can be easily applied to many other full movement activities like martial arts stances or yoga. We also present several methodologies that help synchronize the trainee-coach poses and a light neural network trained with both the trainee and the coach's data to fix troublesome estimations.

1.1. Motivations

We cannot ignore the current unprecedented times we live in: the global pandemic caused by COVID19. Since it is our moral and social obligation to maintain safe distancing, Digital Twin Coaching presents an opportunity in the present day to support a quality coaching experience even in quarantine. Having a Digital Twin technology system focused on Edge Computing can bring significant benefits for contactless coaching and any other field that may need real-time feedback from a trusted professional, such as education, health, and well-being.

Let's create a scenario where one father and son want to remain physically active while in quarantine. The father has an underlying issue with his column, and he needs to perform standing-up stretches several times a day to improve his health. Usually, he would go to an expert since a wrongly executed stretch could worsen his condition, but now he needs to maintain social distancing. The son spends most of his time in front of the PC taking classes, so he would like to avoid staying sedentary most of the day and reactive his legs.

In both cases, they were forced to keep social distance, which affected their ability to do physical activities in controlled environments. Using e-Coaching, we can fill the gap left by this issue as it is tailored to work remotely on any modern device. For example, we can follow the persons' pose in real-time and give instant feedback to avoid any injuries using pose estimation algorithms. Pose estimation also allows us to measure the activity's performance. We can compare the trainee's posture to an expert and gamify the experience, making it more attractive and incentivizing the user to keep using the system.

We cannot expect the trainees to have complex camera setups to follow their posture at every angle, but today it is not hard to have a smartphone or a laptop. Both of these devices have three things in common: they were designed to be portable, they have a camera, and they usually have limited computational resources. Therefore, we need to consider that our system's requirements are adequate for the average users' devices.

There is also the issue of security and privacy. When working with personal records, it is essential to keep the data safe for the users. Therefore, Edge Computing provides an excellent solution to these problems. It presents an environment where the user has total control over their data as it does not need to leave their own hands into could servers. We can have information processing and retention in edge devices like laptops or smartphones. Finally, we need to remember we are working with two totally different profiles, one of an individual with some physical limitations and the other of a person who would be comfortable having a more demanding routine. A Digital Twin of each of these users would grant researchers the ability to create different routines that adapt to each user's needs.

1.2. Contributions

The main thesis contributions are the following:

- Design and development of an architecture for Digital Twin Coaching focused on Edge Computing. We also list several characteristics a DTC must fill as a guideline for future researchers interested in the topic.
- Design and development of a Trainee-Coach pre-processing synchronization technique to compensate for the action lag while training using a video routine and a fast post-

processing procedure for pose estimation merging using light Pose Inference Neural Network (PINN).

- Design and development of a methodology to use an Angular Pose Representation (APR) instead of using joint keypoints for feedback generation to solve issues related to different body proportions and the body's location inside the frame.

We implemented a DTC system called DTCoach based on the pose estimation PINN model. This proof-of-concept application is aimed to be run on any modern devices such as PCs and smartphones. We also performed a usability test analyzing several vital factors that people look for in a virtual coaching environment.

1.3. Scholarly Output

- R. Gámez Díaz, Q. Yu, Y. Ding, F. Laamarti, and A. El Saddik, 'Digital Twin Coaching for Physical Activities: A Survey', *Sensors*, vol. 20, no. 20, Art. no. 20, Jan. 2020, doi: 10.3390/s20205936.
- R. Gámez Díaz, F. Laamarti, and A. El Saddik, 'DTCoach: Your Digital Twin Coach on the Edge during COVID-19 and beyond'. Submitted to the IEEE Instrumentation & Measurement Magazine.

1.4. Outline

The chapters in this thesis are structured as follows:

- **1. Introduction:** In this chapter, we discuss the reasoning behind our research, our motivations, and contributions made throughout the thesis.

- **2. Related Work:** We discuss past work related to the technologies and paradigms we will be working on: virtual coaching, deep learning, pose estimation, and edge computing.
- **3. Proposed System:** This chapter describes the proposed system architecture and the prototype's development to support our research, including the data pre-processing and post-processing.
- **4. Implementation:** In this chapter, we present an implementation of our proposed system with an application called DTCoach. We describe the objective of this proof of concept along with the requirements, coding architecture, etc.
- **5. Usability Studies:** Using the DTCoach application, we performed an anonymous usability test. In this chapter, we discuss the questionnaire and the results obtained from this study.
- **6. Conclusions, Limitations, and Future Work:** Finally, we present all this research project's conclusions and future perspectives that would be interesting to work on.

2. Related Work

Our research topic juggles three main concepts: Digital Twin, Virtual Coaching, and Pose Estimation. In this chapter, we will present a description of the related work on these technologies.

2.1. Smart Coaching

Coaching is a two-way conversation between the coach and the trainee. Even if the trainee is the principal beneficiary of this interaction, we cannot ignore the coach's role. A coach's presence enhances the trainee’s response to stress and improves their performance while also providing motivation [12], [13], a situation that is not an exception in smart coaching.

Smart Coaching, also called Virtual Coaching, can be defined as “a set of smart devices able to work independently to help people improve in a specific field” [14]. It can be seen as a subset of e-learning, which is “learning supported by digital electronic tools and media” [15]. We cannot ignore that the current global pandemic has increased the necessity of remote learning, so e-learning has become more of a reality in several countries in the world. Another concept related to Smart coaching is the Recommender Systems, sets of tools that help curate items, movies, products, etc., from a database based on the users' past preferences and behaviors [16]. In fact, Smart Coaching can use Recommender System techniques in its analysis of the user data for future training plans.

Smart Coaching is not only useful in sports, fields such as wellbeing and rehabilitation can also benefit from this technology [14]. The difference between these three fields is the trainee's current state and what they want to achieve through the coaching. In sports, the trainee is usually a healthy individual whose objective is to be more competitive. In wellbeing, the trainee has good or below average health and wants to maintain a healthy lifestyle. While in rehabilitation, the trainee usually has poor health that they want to recover. In Table 1, we can see the difference between these three fields.

Table 1. Difference in fields that benefit from Smart Coaching.

Field	Current state of the trainee	Objective
Sports	Good health	Be competitive
Wellbeing	Good/bad health	Maintain good health
Rehabilitation	Bad health	Recover health

2.1.1. Smart Coaching for Sports

People looking to excel at any athletic activity to stay competitive, such as sports, will need to rely on third parties' help: nutritionists, massage therapists, medics, and coaches. Athletes are generally in good health, but it is usually not enough; they want to take their bodies to the next level. The problem is that these types of resources are only affordable to a few crowds, like affluent people or elite athletes. Another issue is the limited pool of trained professionals, becoming a bottleneck in tailored personal coaching. Smart coaching solves these problems using technology as a tool that coaches can use to provide better service for the trainees, and in some cases, even being able to serve as a substitute. Next, we will describe how researchers use computer science paradigms such as ML, and AR/VR, to enhance the smart coaching experience in several sports.

Starting with Tai Chi, Kamel et al. [17] try to solve the coach's attention scarcity in a one-to-many class environment using a CNN pose estimation model. They tracked the trainee pose while doing Tai Chi stances and compared them to the teacher’s poses to provide instant feedback. Their results showed that people performed better and felt more engaged while using their system. Hülsmann et al. [18] also focused on tracking the trainee pose for squats and Tai Chi stances, but their system involved Random Forest feature selection and SVM pose classification. For feedback, they designed a Cave Automatic Virtual Environment (CAVE) environment. Here the trainee can see their virtualized model perform the same pose as them while following a VR coach.

Weightlifting is another sport where pose evaluation is critical to avoid wrong postures due to the possibility of severe injury. Yasser et al. [19] used the Kinect to track athletes while performing weightlifting exercises such as squats, deadlifts, and shoulder presses. They compared KNN, SVM, Naïve Bayes, and Fast Dynamic Time Warping[a] (DTW) to determine whether the exercise is being done correctly or not, being the latter the best performing model. Jian et al. [20] took a different approach to analyze the routines; instead of inspecting every frame, they focused only on frames containing key poses for the exercise. They used a Fully Convolutional Network (FCN) to extract the region of interest of every frame and then a CNN to estimate the frame's probability to be a key frame.

There is also decent interest in using smart coaching techniques for tennis and table tennis. Bačić et al. [21] collected joint data on novice tennis players and had professionals comment on their technique. With this information, they trained a polynomial regression model to classify the swing technique as good or bad.

Researchers also had to develop novel ideas for data collection and analysis in water-related environments for sports such as swimming. Jensen et al. [22] used a gyroscope/accelerometer sensor placed on the back of the head of 12 professional swimmers while they performed a training routine comprised of four different swimming styles. The sensor data was used to train a linear regression classifier for these resting, turning, and these four swimming styles. Kobayashi et al. [23] had a similar approach, but they placed the sensors in the torso area, and they used a decision tree classifier.

---

[a] For more visit https://en.wikipedia.org/wiki/Dynamic_time_warping.

Different other sports can benefit from smart coaching using ML, for example, golf [24], martial arts [25], and cycling [26], among others.

2.1.2. Smart Coaching for Wellbeing

The objective of the field of wellbeing is to maintain a healthy lifestyle. Therefore, research focused on specific activities such as stretching and yoga. The target population in this field is broader compared to the field of sports. For example, seniors looking to engage in healthy ageing or middle-aged sedentary adults trying to stay physically active.

As mention above, yoga has been the focus of many studies for smart coaching. Pullen et al. [27] used Microsoft Kinect to collect pose data from six yoga instructors and trained an AdaBoost classifier for these yoga poses. Then, they tested their approach with 20 participants throughout 10-weeks, reaching more than 90% accuracy for all poses. Trejo et al. [28] followed the same research line using Kinect and an AdaBoost classifier for six yoga poses. They took it one step further into the coaching perspective by giving audiovisual feedback to the trainees. Amrani et al. [29] designed a complete serious game based on full-body tracking for yoga using Kinect and an Artificial Neural Network (ANN). Their application's advantage is that it gives relevant feedback on improving posture while performing yoga exercises. Gochoo et al. [30] used three infrared cameras to record the trainees instead of using the Kinect, mentioning that their approach respects the users' privacy and has better long-term support. They also trained a CNN to classify 26 yoga postures using data collected from 18 participants.

Similar to yoga, stretching has caught the attention of several researchers. Kumar et al. [31] used the Kinect to collect pose data to classify ten different stretch postures using a Random Forest Classifier. They also provided a 3D representation of the posture, which can be followed on a screen.

Delgado Bellamy et al. [32] addressed the issue of the growing ageing population worldwide, which researchers forecast will reach 8.6 billion in 2030 [33]. They proposed a robot assistance coaching system to support this population's needs and help the limited workforce. Using a KNN model, they were able to classify the range of motion in each user's stretching routine to personalize the feedback given to them via robot-human interaction or a tablet.

Smart coaching for wellbeing does not stop at physical exercise; Erdeniz et al. [34] proposed a system called Quantified Self that focuses on tracking everyday behavior to improve habits such as sleep time. They took advantage of IoT and smart wearables to collect the needed data using devices with sensors like oximeter, blood pressure sensor, heart rate monitor, and glucometer, along with personal data such as height, age, etc. A dashboard was designed to present the processed information of the participants' daily routine, steps count, sleep time, calories burned, etc. and offer several recommendations about how to improve.

2.1.3. Smart Coaching for Rehabilitation

As mentioned in [32], one of the main issues in all types of coaching is the limited resources and workforce available for the general population. The truth is, there are not enough healthcare workers to provide a complete customized rehabilitation service for all the people that needs it [35]. The person may also suffer from motion limitations or any other condition that could affect their ability to reach professional rehabilitation services. Another problem is the cost of rehabilitation, usually out of possibilities for a large part of the vulnerable population. Having virtual/smart coaches available in devices such as smartphones or personal computers can help alleviate some of these issues.

In this chapter, we will discuss how researchers approach smart coaching issues for rehabilitation. While doing the literature review, we found three central target populations in research: stroke patients, cerebral palsy, and limb disorder.

Stroke patients were the main target of the literature reviewed. The most common sensors to collect data in this line of research are the Electroencephalogram[b] (EEG), recorded using electrodes measuring brain activity, and the Electromyography[c] (EMG), which measures the electrical muscle activity.

As advanced as sensors such as the EEG can be, it also faces a problem in stroke patients as their brain waves do not behave the same as healthy patients. This is why Liang et al. [36] proposed F-boost[d] to improve a set of weak SVM classifiers for motor imagery classification. Another approach was taken by Zhang et al. [37]; they trained an ensemble of two models using an iterative method to classify hand movements. Using electromyography, Yang et al. [38] fused VR and EMG in a smart coaching system for hand movement rehabilitation using SVM for gesture classification.

The Kinect is also used in rehabilitation for stroke patients. Lee et al. [39] used this device to perform motor analysis in stroke patients while doing daily life exercises such as bringing a cup to the mouth or flipping a light switch and provided a dialogue interface for feedback. Chen et al. [42] used EMG sensors to train a KNN model to classify the therapist's hand movements. These movements are then simulated by a robot hand exoskeleton allowing the possibility of upper limb rehabilitation. Guo et al. [40] also used a robot hand for self-rehabilitation. They classified EMG

[b] For more: https://en.wikipedia.org/wiki/Electroencephalography.
[c] For more: https://en.wikipedia.org/wiki/Electromyography.
[d] For more: https://en.wikipedia.org/wiki/Boosting_(machine_learning).

signals using a Wavelet Neural Network (WNN) to pilot the robot hand, which in turn serves as a guide for the patient.

Wang et al. [41] also used EMG sensors and an RGB camera, but now for children with cerebral palsy. They used OpenPose, a pose estimation model, to track the patients while doing treadmill exercises wearing EMG sensors and compared their evaluation with the Gross Motor Function Measure[e] (GMFM) standard.

2.2. Pose Estimation

Computer Vision aims to build computational models and autonomous systems capable of performing visual tasks similar to humans [42]. One of the most exciting tasks in this field is Human Pose Estimation, which refers to "computer vision techniques that detect human figures in images and videos" [43]. For example, we can determine the location of different body parts, like a person's joints, in a photo. There are two types of pose estimation: 2D and 3D, with the difference being the number of axes used in the prediction. It is not necessary to use a photo captured using an RGB-Depth (RBG-D) camera to estimate the joints in 3D; there are ways to calculate 3D points using a 2D image that we will discuss later in this chapter.

Recently, most of the research in pose estimation for both 2D and 3D is done using DL models. One of the first approaches using DL for pose estimation is made by Toshev et al. [44]. They used a CNN in a cascading architecture to achieve state-of-the-art results with the added benefit of estimating hidden joints and valid poses in challenging photos. Tompson et al. [45] also used CNNs with a twist. Instead of using pooling and subsampling for joint regression, they introduced heatmaps into pose estimation, mapping the joint's location's probability in the image.

[e] For more: https://canchild.ca/en/resources/44-gross-motor-function-measure-gmfm.

They proposed a two-step architecture. First, they had a coarse heatmap CNN model. The output of this model is then passed to a fine heatmap model for final estimations. The use of probability heatmaps became a standard in future pose estimation research.

In 2016, Newell et al. [46] introduced the Stacked Hourglass Networks DL model for pose estimation. This model consists of repeated pooling and upsampling layers giving this “stacked hourglass” shape (Figure 1). This architecture's objective is to provide a coherent understanding of the body at every scale: orientation, limb position, joint relationships, etc. In their words, “the hourglass is a simple, minimal design that has the capacity to capture all these features and bring them together to output pixel-wise predictions.”

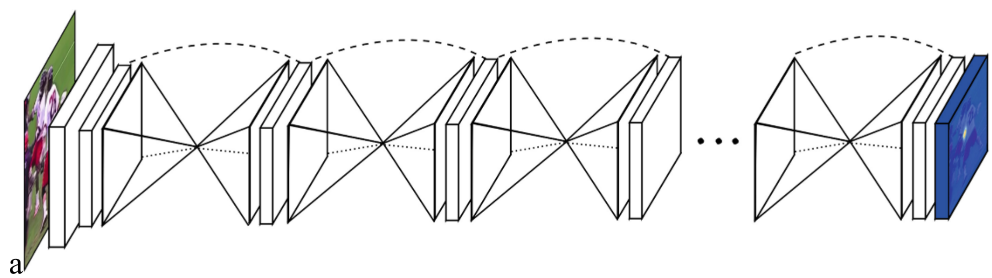

Figure 1. Stacked Hourglass Network [46].

One of the most exciting tasks in pose estimation is 3D estimation from 2D images. Typical approaches for 3D estimation focus on end-to-end methodologies; this means they go from pixels to joint predictions [47]. Martinez et al. [48] proposed a different approach: first, they estimated the 2D joints, then they used a simple deep feed forward network to predict their 3D positions. Their methodology aims to understand the errors made while trying to estimate 3D joints. For example, mistakes generated from limited 2D pose understanding or from lifting those estimations into a 3D space.

TensorFlow (TF) is an ML python framework proposed by Google with 150k stars and 83.4k forks on GitHub, making it the 5th most starred and the 4th most forked project on the site as of December of 2020[f]. It has been slowly reducing ML and engineers/scientists' gap, especially when libraries such as Keras[g], built on top of TF, allow complex models written in just a few lines. The advantage of using these libraries is that the companies that maintain them, Google in this case, along with the open-source community, are always updating it, improving it, and adding more features. For example, some pre-trained models are implemented directly in the library. One of these models is PoseNet [52], for 2D pose estimation. PoseNet is based on the work by Papandreou et al. [49], [50]. They proposed two multi-person detections, pose estimation and segmentation approaches. The first is the top-down approach, where they first focused on detecting each person's bounding box and then extracting the corresponding keypoints. The second one is the bottom-up approach that first localizes identity-free entities and then groups them into person instances. The advantage of using PoseNet is that it is already optimized to run on Tensorflow.js, the web version of the standard TF library.

There is plenty of research done using DL-based pose estimation models for sports and coaching. One example is the work done by Kurose et al. [51] called "Player pose analysis in tennis video based on pose estimation." Their objective was to provide objective feedback to the player without a qualitative opinion of experts. They estimated the players' joint positions from tennis match videos using Part Affinity Fields [52] and fed them to an SVM to predict the racket shots' success.

---

[f] From https://github.com/EvanLi/Github-Ranking.
[g] Found at https://keras.io/.

Kamel et al. [17] proposed a system called iTai-Chi using CNNs for pose estimation. They used a Kinect to collect RGB-D data from the participants to train a 4-layer CNN for 3D pose estimation. These estimations are then post-processed to calculate the Motion Parameters, which are a collection of angles and distances of the joints estimated. The parameters are used to compare the participants' motion to a Tai-Chi professional, provide feedback, and correct mistakes.

Talking about another field, Wang et al. [53] designed the AI Coach system for skiing and other fast movement sports. The main problem in high-speed sports such as skiing and soccer is the motion blur. Their system focuses on tracking and taking advantage of spatial and temporal relation of the body keypoints in videos to solve this issue. They first used a CNN model to obtain the keypoints from each frame, which are then passed to a different module to extract the spatial and temporal relations between frames. The output of this novel pose estimation architecture is used to train an SVM with a radial kernel to classify the poses as good or bad.

2.3. Deep Learning and Edge Computing

Deep Learning does not need an extensive introduction. Being a subset of ML, DL focuses on Artificial Neural Networks (ANN), structures inspired by how neurons communicate in the brain. DL has been steadily rising in popularity since the groundbreaking paper by Krizhevsky et al. [34] focusing on image classification using CNNs. Since then, researchers have used DL's flexibility for tasks such as speech recognition, Natural Language Processing (NLP), self-driving cars, and many others.

To improve results, researchers often choose to add complexity to the Neural Networks ending with resource-heavy models. For example, top NLP models for the SuperGLUE tasks[h]

[h] Found at https://super.gluebenchmark.com/leaderboard.

were designed by leading tech companies such as Google and Huawei, who have access to *virtually unlimited* computational resources for their research.

Edge computing, defined by Shi et al. [54], “refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services.” Edge Computing's main advantage is the opportunity to delegate most of the processing bulk to the edge, in other words, to the clients or endpoints of the ecosystem. These endpoints can be embedded devices, smart wearables, personal computers, etc. Edge Computing implementation of DL models is a difficult task. Most of the devices mentioned are designed to be portable and as independent as possible, so the priority is to consume as few resources as possible [55].

One example of an “edge” device is the smartphone, a piece of hardware that has become a necessity in current times. For this reason, it is crucial to consider Edge Computing as a valid research path for DL that can be applied to many fields. For example, Martinez et al. [56] presented a system called Cardio Twin. They used a CNN to detect Ischemic Heart Disease in ECG signals reaching 85.77% accuracy with a classification time of 4.8 seconds. The most interesting part of this paper is that the architecture was built with Edge Computing in mind, using TensorFlow Lite to run their model in middle-range smartphones.

Giammatteo et al. [57] designed two CNNs based on the VGG16 architecture [58] for age and gender classification. Their models weighted less than 600MB and could run on edge devices like the NVIDIA Jetson Nano with 4GB of RAM. Chen et al. [59] also used CNN for their pedestrian detection system, but they based their model on the YOLO architecture [60]. To improve the performance of the YOLOv3-tiny model, they extracted only relevant features using different image sizes and modified the feature layers accordingly, reaching 31fps in the NVIDIA

TX2. CNNs are not only useful for humans analysis, Liu et al. [61] proposed using them to classify vegetables in mobile devices using a combination of MobileNets, Transfer Learning, and Model Quantization. They implemented the model using TensorFlow Lite.

One of the most critical developments in DL for Edge Computing are “MobileNets” by Howard et al. [62]. Their architecture uses depth-wise separable convolutions, which factorizes the filtering and combining of standard convolutions into two separate layers, reducing computation and model size. They also proposed two model shrinking hyperparameters to reduce the model's size and parameters at a reasonable cost of accuracy: width multiplier and resolution multiplier. The first hyperparameter is used to thin the network uniformly at each layer, while the latter reduces the input image resolution affecting the internal representation of every layer. As a follow-up of this work, Sandler et al. [63] proposed “MobileNetsV2”, which reduced the runtime memory usage with an inverted residual with linear bottleneck layer. This layer inserts linearity into convolutions to combat information loss while at the same time skipping expansion layers reducing the overall complexity of the model. The PoseNet models designed by Google, described in the last subchapter, are an implementation of the work of Papandreou et al. [49], [50] using MobileNet, MobileNetV2, and ResNet [64].

Finally, it is worth mentioning the recent developments for training and running ML models using JavaScript. This allows any device with access to modern web browsers to use ML and, consequently, DL. There are currently three main open-source libraries for ML/DL using JavaScript: Tensorflow.js[i] maintained by Google, ONNX.js[j] maintained by Amazon, Microsoft, Facebook, among others, and WebDNN[k] maintained by the University of Tokyo. Working with

[i] https://www.tensorflow.org/js
[j] https://github.com/microsoft/onnxjs
[k] https://github.com/mil-tokyo/webdnn

these frameworks, engineers can use their models designed with other libraries such as Keras and PyTorch.

2.4. Remarks

When designing a Digital Twin Coaching system, we will encounter several issues that should be solved to be considered a complete solution. For example, the number of resources, computational or hardware related, needed to research in this field. In a survey done in 2020 [14], researchers showed that even if half of the articles surveyed used a user-friendly device such as Kinect, much of the work done used hardware not accessible for the average user. This issue becomes a barrier of entry that manufacturing companies can overcome if they wish to use DT technology but not possible for anybody else.

Smartphone adoption is increasing, and so is the popularity of IoT and wearable technology [65]. We can take advantage of this situation to design and implement Digital Twins for Edge Computing (EC). One of the issues regarding EC is the device processing limitations; we will not always be able to perform heavy computations in real-time, so we need to come up with alternatives to compensate for this shortcoming. DTC gives us the unique opportunity of working within an environment where we know what the trainee is trying to perform beforehand. Therefore, we can fine-tune generalist models, such as pose estimation neural networks, for the coach-planned routines. Within the DTC ecosystem, we can develop ML models that are more optimized in resource consumption and performance for the task at hand.

Coaching is a “two-way conversation” between the Coach and the Trainee. Consequently, having real-time feedback is one of the priorities that a Digital Twin Coach (DTC) should fulfill. One of the advantages of having instant and precise feedback for the trainee is injury prevention, as injuries can happen in a fraction of a second, and a DTC can quickly correct any bad posture that may lead to such tragedies. Real-time feedback can also help engage the trainee with the system from a usability perspective, having an overall better user experience.

In 2016, Schwab [66] coined the concept of Industry 4.0 to describe the trends of automation and advanced communication, and complex data analysis in manufacturing industries. Industry 4.0 is a convergence of many technologies such as Cyber-Physical Systems (CPS), Internet of Things (IoT), Could Computing, and AI. This gave rise to Health 4.0, where they applied the concepts used in Industry 4.0 into the Healthcare and Wellbeing industries [8]. Both Industry 4.0 and Health 4.0 share many characteristics with DT technology. Therefore, a DT focused on wellbeing, sports, or physical exercise coaching, is an excellent point to start.

3. Proposed System

We performed our own data collection for training (chapter 3.4.1) and a usability test (chapter 5). The research experiments have been approved by the University of Ottawa Research Ethics Board. File Number: H-08-20-5859.

3.1. DTC Ecosystem

As discussed previously, the Digital Twins concept originated from the manufacturing industry, but it has made its way to the Health sector recently. El Saddik proposed in 2019 an ecosystem for the Digital Twin for health and wellbeing [68]. This ecosystem, shown in Figure 2, has five main components: Data Source, AI-inference Engine, Multimodal Interaction, Communication, Privacy/Security, and quality of Experience-based Feedback. Since this architecture is focused on individuals, the way it is described makes it hard to understand how a DT would interact with external entities. Therefore, if we want to apply it to virtual coaching, we need to consider aspects such as the interaction with the coaching staff and other team members.

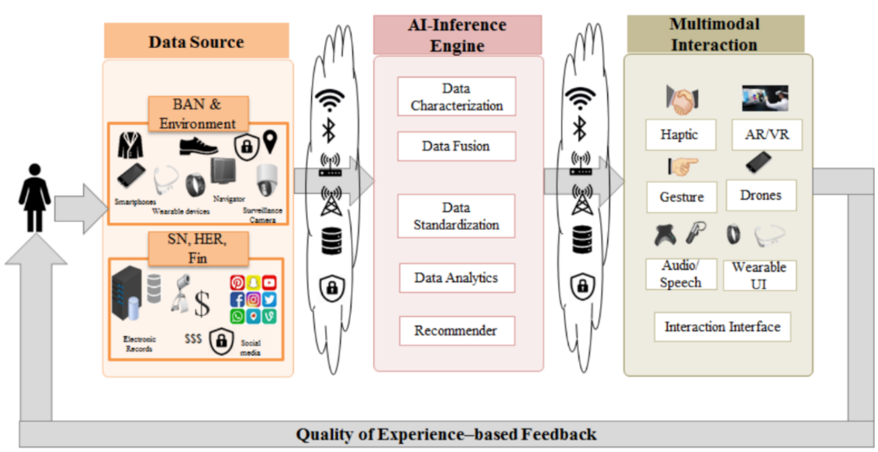

Figure 2. Dtwins ecosystem: DT for health and wellbeing [68].

In Figure 3, we present the Digital Twin Ecosystem for Coaching (DTC ecosystem) based on the work of El Saddik. The main difference is that we have two Real Twin entities: the trainee and the coach. The trainees will still be the system's focal point since they are the party that benefits the most from DTC. The role of the Coach is to provide guidance, expertise, and feedback for the system to improve the training routines. There is also external historical information collected from other team members that the DTC can use to its advantage.

The two actors of the DTC ecosystem are:

- **Coach.** They are the starting point of virtual coaching, even if they are not directly benefited. The idea behind having a coach in the ecosystem is to serve as the experts. Coaches have the knowledge to design a complete training routine for the trainees even without the DTC existing in their planning. The aim of DTC, in this case, is not to replace them but serve as another tool in their arsenal. Coaches can use DTC to have a better knowledge of the trainees and improve the overall coaching efficiency. Both the DTC ecosystem and the coach should benefit from each other.
- **Trainee.** The central part of the whole system. They are the ones that benefit the most from being part of the DTC ecosystem. Their primary purpose is to follow the DTC and the Real Twin Coach (RTC) instructions and aim to provide accurate data to the system. The more data is collected from this party, the better the system will improve the analysis. Consequently, their collected data will enhance future training plans and help the DTC ecosystem and RTC make better decisions.

Meanwhile, the DTC ecosystem is mainly divided into the following components:

- **Data Source.** The first module inside the DTC system. This module is where all the pertinent data will be collected to feed the rest of the ecosystem, mainly the Smart Module. There are two types of sources: hard sensors and soft sensors. Hard sensors are the hardware that can be used to collect data from the real world. They can be audio/visual such as cameras, any type of wearable tech like smartwatches or fit trackers, or specialized medical equipment in a laboratory setting. Soft sensors allude to data collection services. They may be personal records such as medical history, feedback records (questionnaires or comments), or social media behavior, which is not needed for this case of virtual coaching. There is one crucial difference between the coach and the trainee's data collection. The coach's personal information is not needed since only the trainee's performance is analyzed because the DTC ecosystem objective is to adapt to their physical limits and requirements. Therefore, the coach's required information is instructional (for the routine planning) and corrective (feedback on how the trainees can improve or enhance their training). There is also a third module involved in the data source: historical team records. Neither the trainee nor the coach exists in an isolated environment. The trainee may be part of a sports team, and the coach can have several people working with them, even in different unrelated groups. From the data collected of these two individuals, pertinent anonymized data is saved into historical records to improve future models in the Smart Module.
- **Smart Module.** The component where the *intelligence* of the ecosystem comes from. Different techniques to analyze data are applied here. They can range from Machine Learning, Deep Learning, Recommender Models, Computer Vision, Data Mining, etc. This component's main objective is to process the raw data collected from the Data

Source module. Then, the Smart Module analyzes the data before passing it to the Multimodal Interaction Interface to produce useful feedback. An example of these techniques is to use the biometric data collected from the trainee during training to adapt future routine intensity. It can also provide future predictions to the coaches about the trainee's current performance to make any decisions regarding the training plan. This component can also provide real-time analysis for activities such as posture correction in gym settings and injury tracking to avoid worsening the condition. This module takes information from three sources: the trainee, the coach, and the historical team data. Finally, even though being anonymized to protect other individuals' identities, the historical team records can be used to improve different AI models and enhance all individuals inside the ecosystem.

- **Multimodal Interaction Interface (MII).** After the data is analyzed, it needs to be passed through the Multimodal Interaction Interface (MII) to provide real value. The MII is a set of techniques and hardware in which the data is transformed into digestible media for the RT to consume. After being processed by the Smart Module, the information is not useful to make decisions yet. The MII oversees the post-processing of the data into media that could be auditive, visual, text, haptic, etc., or any combination of them (multimodal). The hardware used here ranges from complex systems such as VR/AR headsets or CAVE [69] to simple devices like webcams. In this ecosystem's case, it is essential to be unintrusive as possible since the trainee will usually need space and flexibility to make a wide arrangement of movements. Therefore, it is preferable to use audiovisual and haptics modalities to interfere as little as possible with the training routine while keeping real-time feedback.

- **Communication Channels.** Tying all together are the communication channels in charge of information flow between all the other components. The communication channels are physical and virtual mediums in which the information flow: Bluetooth, 4G, 5G, wired, etc. They also include any protocols and interfaces such as IEEE standards like X73[70].
- **Cloud and Storage Services.** Cloud storage saves the collected information in a secure environment that can be accessed as needed. It can be stored online in remote servers or saved locally to be accessed through low-distance channels such as Bluetooth. Storage is not the only use of cloud services as they can also include data collection assistance, such as weather or nutrition reports. There is also the possibility of leasing processing power from services like Azure or Amazon Web Services (AWS) on demand.
- **Privacy and Security.** Finally, one of the most overlooked components in current research is the issue of privacy and security. For the whole ecosystem to work correctly, the trainee and the coach need to be sure their data is safe, whether on a server or locally. It encompasses all the components from the moment information is read from the RTs to the moment the data is displayed to them.

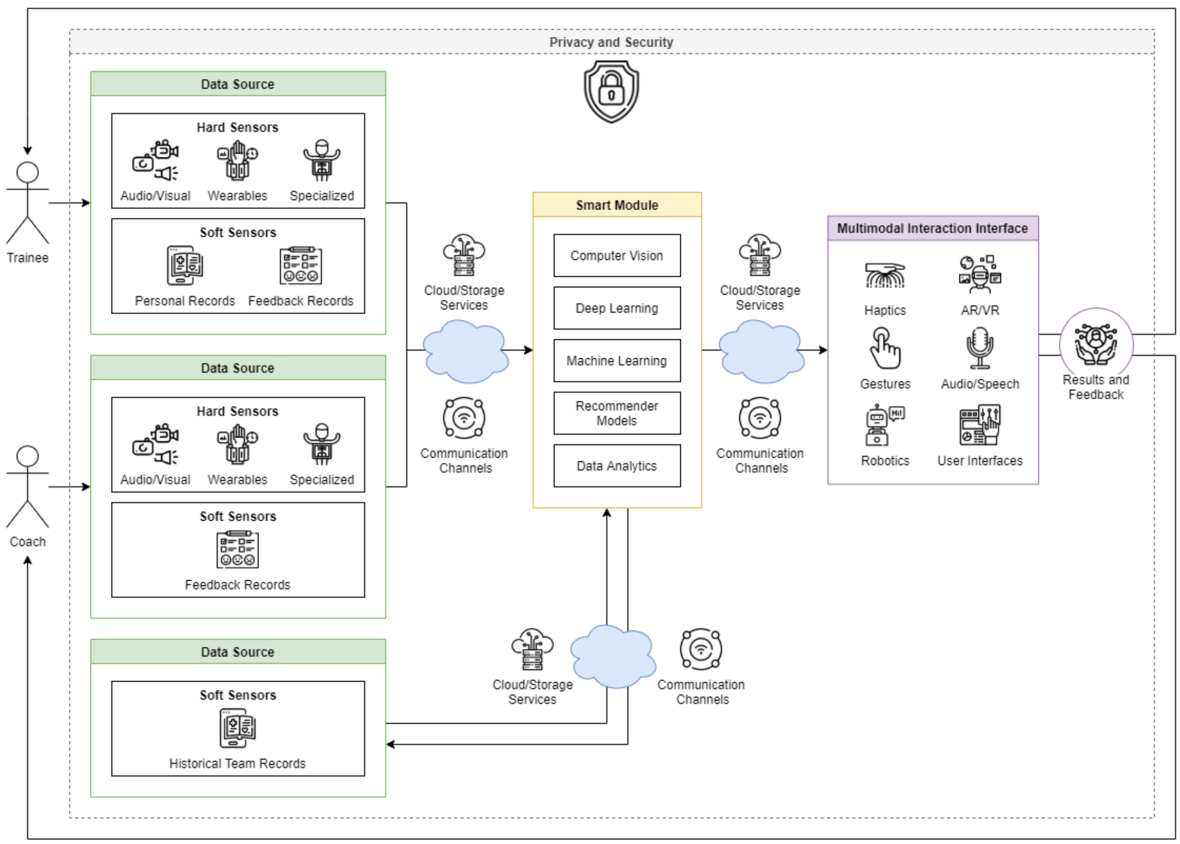

Figure 3. DTC ecosystem.

3.2. Characteristics of DTC

Based on the ecosystem presented in the last chapter and the literature reviewed, our proposed DTC system will be based on five different characteristics. These points are not exhaustive but should serve as the baseline for future development in the field of Smart Coaching. These characteristics (Figure 4), ordered alphabetically, are:

- **Adaptability:** People have different heights, weights, body proportions, etc. Therefore, the system should adapt to the body of the person that is currently using it. It does not stop in the trainee's physical characteristics. For example, stretching exercises may be performed by both a high-level athlete and a 5-month pregnant woman, so the DTC system should adapt to different capabilities and needs. Adaptable routine planning and feedback types should also be considered. We can leverage the power of technologies such as Recommender Systems, ML, DL to adapt our coaching routine, feedback, and algorithms. Adaptability also includes the system's ability to evolve as the trainee's performance changes over time, for better or for worse.

- **Compatibility:** Our DTC system should be as accessible as possible, especially when working in edge computing environments where we may encounter limited hardware resources. We should be focusing on getting the DTC in as many devices as possible, translating the system into different languages, using a development platform that can deploy to many environments, etc. Compatibility also requires researchers to think about computation resources. Current DL models usually focus on accuracy over resource consumption, which becomes an issue in the DTCoach's target environments. In that case, we should be looking for alternatives and optimization techniques such as model compression (e.g., quantization and pruning) and model distillation (e.g., transfer learning).
- **Flexibility:** Following the steps of Adaptability, Flexibility is all about the trainee and coach preferences. The DTC system can adapt to the trainee performance and physical measurements, and it needs to use this knowledge for any routine that the coach may plan. DTC needs to be flexible enough to be applied to light routines such as stretching as well as heavy performance routines like weightlifting. The smart module also comes into play here as it allows the DTC to run different types of routines using technologies like ML and DL.

- **Portability:** Current smart coaching systems focuses on providing a full coaching experience despite the coach and trainee's hardware available resources, often using complex equipment such as CAVE or RGB-D cameras. This may work for exceptional cases, but the objective of DTC is to be available anywhere, and anytime the trainee needs to perform the routine. There is always a tradeoff researchers need to investigate so that the DTC can be run in edge devices such as smartphones without compromising the overall coaching experience.
- **Privacy and Security:** Finally, tying all together, we have privacy and security, allowing the individuals using the DTC the peace of mind that their data is safe. Whether the data is shared to the server or stored within their edge device, it should always be encrypted so only the owner or any trusted party can get access to it.

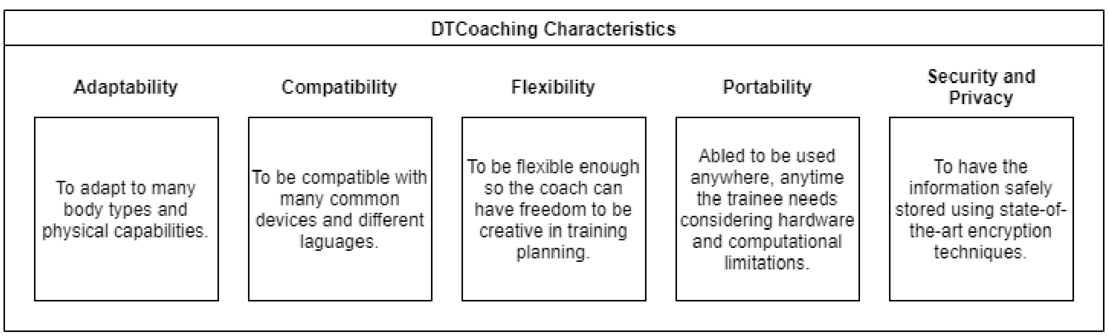

Figure 4. Summary of DTC characteristics.

3.3. DTC System Design

Considering the characteristics described in the last chapter, we can apply the DTC ecosystem of Chapter 3.1 to a specific field within physical coaching. For example, DTC for physical activities based on pose estimation DL models for Edge Computing.

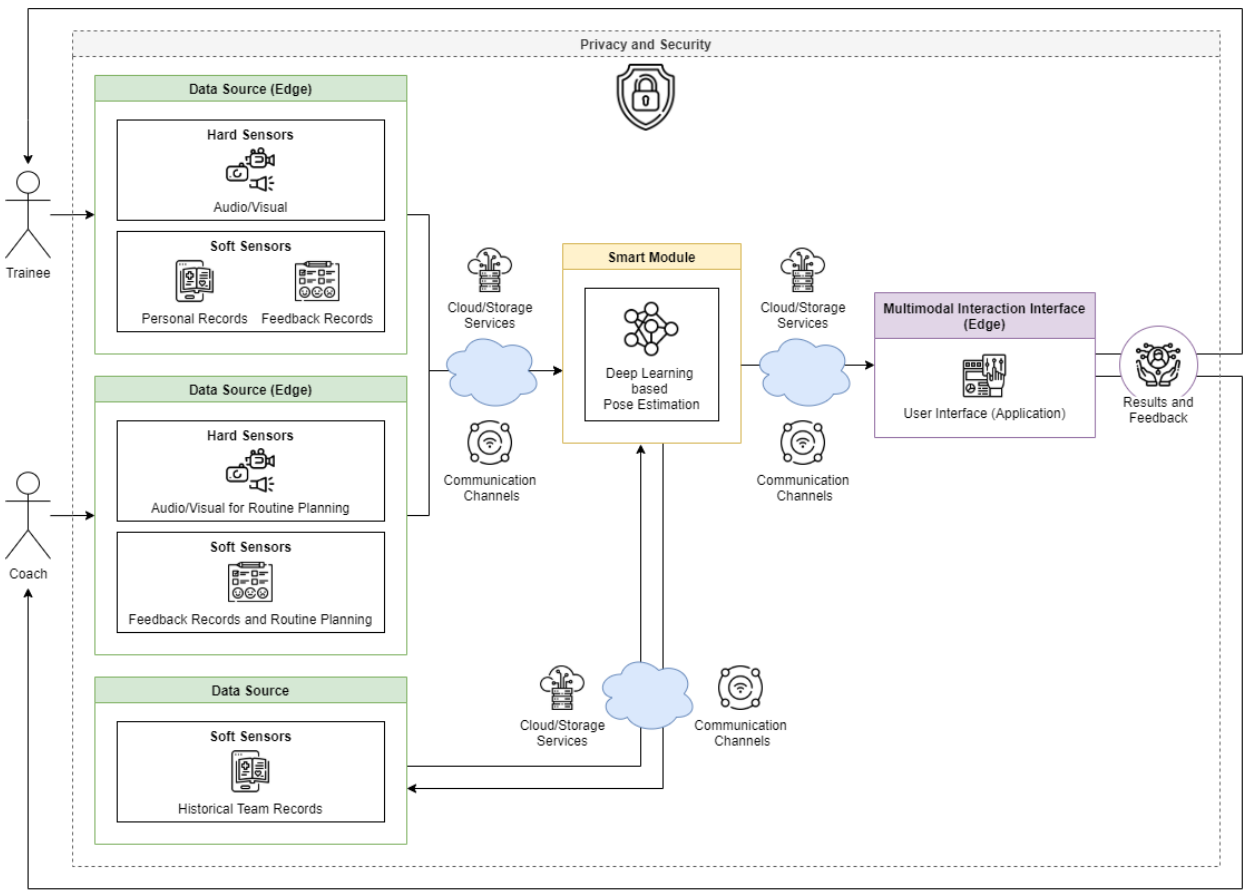

Figure 5. Applied DTC Ecosystem.

In Figure 5, we can see the applied DTC Ecosystem for this case. Starting from the trainee sensors, only audio/visual data needs to be collected from the hard sensors. Personal records are still required because information such as age, weight, height, etc., comes in handy when fine-tuning the DL algorithm for each person. Satisfaction questionnaires and comments should also be considered to provide useful information for the system's overall improvement. The data collected from the coach sensors is for routine planning. The audio/visual information will serve as the basis of the routine that will be presented. Simultaneously, the coach's feedback records will be related to the trainees' performance to improve later sessions. Historical Team Records in this ecosystem serve for improving the Smart Module of the DTC system using anonymized data from other trainees.

The smart module in this ecosystem uses the hard sensors' data to train and fine-tune a pose estimation DL model, which will be the heart of the analysis. The smart module receives visual information from both the trainee and the coach and, using a DL model, estimates the pose from both in real-time and gives relevant feedback. This feedback depends on the recipient. The trainee will receive their performance analysis while doing the routine so they can correct any mistakes in real-time. Simultaneously, the coach can see the training history to check if they need to change the routine in any meaningful way to adapt to the current trainee's needs. The feedback is presented using a graphical interface in the users' device since this system is designed to work on the edge.

In the next three chapters, we will thoroughly discuss the steps to make this DTC System a reality. In chapter 3.4, we present the shape of the data along with our approach used to collect it to train and fine-tune the pose estimation DL model. Chapter 3.5 presents the architecture, optimization, training methodology, and pose estimation neural network results. In chapter 3.6. we discuss the approach we took to solve common issues in pose estimation such as occlusion and differences in human proportions to provide relevant feedback to the trainee.

3.4. Data Collection

In this chapter, we will first discuss the data collection process. Second, we will present the shape of the data used to train the NN and generate feedback in our proof-of-concept implementation (Chapter 4).

3.4.1. Data Collection

There are currently no datasets available of coach-athlete synced videos for our task in hand. For that reason, we chose to collect data for our experiments. We performed three different data collection procedures: pose estimation routine collection, video routine collection, and usability test collection.

First, we asked ten people to perform a 5-minute routine in their free time over one month using our DTCoach application for the pose estimation routine collection. This 5-minute video routine is composed of various 30 seconds stretching and squatting exercises with a rest time of 10 seconds between them. The DTCoach app then collects the pose estimation of the subjects' movement while they are performing the routine. We ended up with 38 routines collected, which add up to more than 110,00 single frames that can be used for training since each routine is composed of approximately 2900 frames. The participants' profiles were seven males and three females, ages 22 to 58.

Then, we recorded five people in a controlled environment for video routine collection using a laptop webcam. The video was trimmed to 302 seconds. This dataset was used to test our NN architecture. Therefore, we needed to make sure the participants' full bodies were always in the frame. The participants were four males and one female, ages 27 to 58.

Finally, we did a social media campaign to test our DTCoach application for a usability test. We provided instructions on how to use the application and let the participants use it freely, including user creation, routine performing, and usability test answering. We did not collect any information aside from the usability questionnaire that pops up at the end of every video routine. In total, 43 people tested our application, 19 males, 16 females, and 8 who chose not to disclose. The ages range from 17 to 60 years, with the range of 21 to 30 being the most prominent.

3.4.2. Data Shape

Each frame processed by PoseNet produces a set of 17 2D-keypoints (as shown in Figure 6) along with one confidence score for each keypoint. Each keypoint is made of the *x* and y pixel coordinate of a body part inside the image frame. The confidence score is a number between 0 and 1 that indicates the probability of the keypoint being in the position that PoseNet estimates. Therefore, each frame produces a total of 51 possible features: 34 from the 17 2D keypoints and 17 from each confidence score.

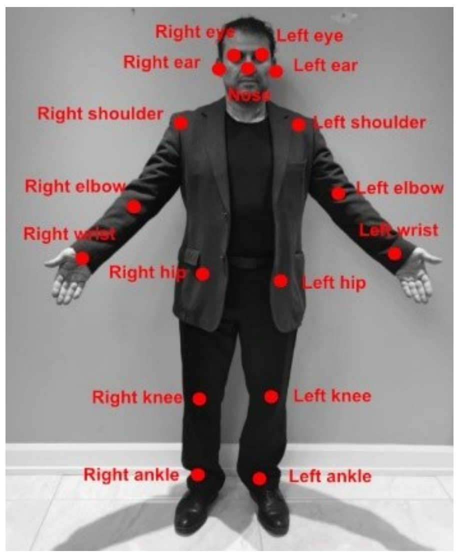

Figure 6. Keypoints produced by PoseNet.

We did not use head-related keypoints for the NN training since we are only interested in the extremities for pose estimation. We also used a middle point between the shoulders and another middle point between the hips to connect the top and bottom keypoints for easier visualization and to serve as anchor joints in the Angular Pose Representation (more on this in chapter 3.6). In the end, we have 14 keypoints in total, as shown in Figure 7, along with their confidence scores.

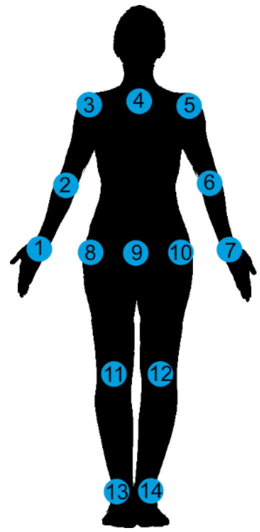

Figure 7. 14 keypoints for training and feedback generation.

3.5. Pose-Inference Neural Network (PINN)

One of the objectives of this research is to take advantage of the DTCoach ecosystem to improve our machine learning algorithms. In this case, we use the trainee and the coach pose estimation results to train a NN. This NN learns the trainee's intentions and infers which pose they are trying to do. Then, we perform a correction on the original pose estimation for low confidence keypoints. In this chapter, we detail this process.

3.5.1. PINN Architecture Optimization

To select the best architecture for the PINN we used Random Search Optimization (RSO), which is a methodology consisting of running several trials with randomly chosen parameters and select the configuration that performed the best. Since we want to keep our NN light to run it in real-time with decent framerate, we decided to test for 1, 2, 3, 4, 5, and 10 layers. Each layer was tested using a node range from 1 to 100. For each layer configuration, we performed 50 trials using the Keras Tuner[1] library.

In Table 2, we present the results of the RSO. From the 50 trials made, we show the average MSE of the 50 trials and the MSE of the best result we had. We also present the parameter count for the best-performing configuration. Finally, the last column shows the time it took to process each sample for the top configuration. It is important to note that these tests use the pre-processed data described in chapter 3.5.2.

After analyzing these results, we decided to choose the 2-Layer Neural Network (2-LNN) for two reasons. The 2-LNN was the fastest configuration, with an average time per sample of 179.89 µs; when working in real-time for edge computing devices, every fraction of a second count. The second reason is the tradeoff of parameter count and performance. Even though the 5-LNN achieved better performance, it was 10 µs slower (5% slower) than the 2-LNN. It was triple in size (%293 bigger) while only being 28% better than the 2-LNN. These are the reasons we chose to go with the 2-LNN, which has the following node configuration: Input-42, Hidden-88, Hidden-26, Output-28.

[1] Found at https://www.tensorflow.org/tutorials/keras/keras_tuner.

Table 2. Random Search Optimization results over 50 trials.

Number of layers	Avg MSE	Std Dev MSE	Best MSE	Best Param #	Avg time per sample (μs/sample)
1	0.0181	0.0008	0.0176	7585	183.58
2	0.0171	0.0015	0.0150	8660	179.86
3	0.0158	0.0018	0.0136	21038	195.40
4	0.0151	0.0018	0.0126	19399	214.56
5	0.0152	0.0019	0.0123	25429	189.70
10	0.0188	0.0058	0.0149	29614	213.34

3.5.2. Data Pre-processing

Before training the PINN, we must pre-process the collected data. The PoseNet model produces a set of 17 keypoints with their confidence score, but we cannot use the data as-is for several reasons. First, each keypoint is a 2D coordinate that marks the body part's location inside the processed image frame. We cannot control where the subject is in the frame, so even if the trainee is following the coach perfectly, it may differ significatively because he is not standing in the same place inside the frame of the picture or video taken. Second, the trainee’s distance to the camera will differ for every participant and the coach, so we will have conflicting lengths between body parts. Third, the coordinates of images in JavaScript start counting from the top left corner. This is a problem as the coordinates' difference in values between left and right body extremities can insert bias into the NN initial weights even if the pose being analyzed is symmetrical.

The data pre-processing is divided into five steps (Figure 8):

1. **Centralizing:** Coordinates (pixels) in images are counted from the top left corner, and we cannot control where the person is in the frame of the picture. Therefore, the first step is to centralize to 0, 0 using the centroid of the keypoints as reference.

2. **Mirroring:** While doing experiments, for understandability purposes, we rotate horizontally and vertically the coordinates so that they are not flipped upside down.
3. **Normalization:** With the objective of decreasing weight bias, we normalize each keypoint coordinate in the range of -1 to 1.
4. **Pose Synchronization:** While the trainees are following the routine, they will not perform the same pose in the video routine simultaneously. Synchronization is the process in which we try to compensate for this action lag. When processing the trainee pose, we look up in a 41-frame window for the most similar pose in the routine video using the R^2 score. We chose a 41 frame because having a bigger window did not improve the R^2 score much while the time it took to process one frame constantly grew, as seen in Table 3. This match is then used for training. In Figure 9, we see an example where the middle image is selected as it is the one with the best R^2 score.
5. **Flatten.** Finally, we flatten all the keypoint coordinates and confidence scores into a 1D array to feed them into the NN for training and prediction.

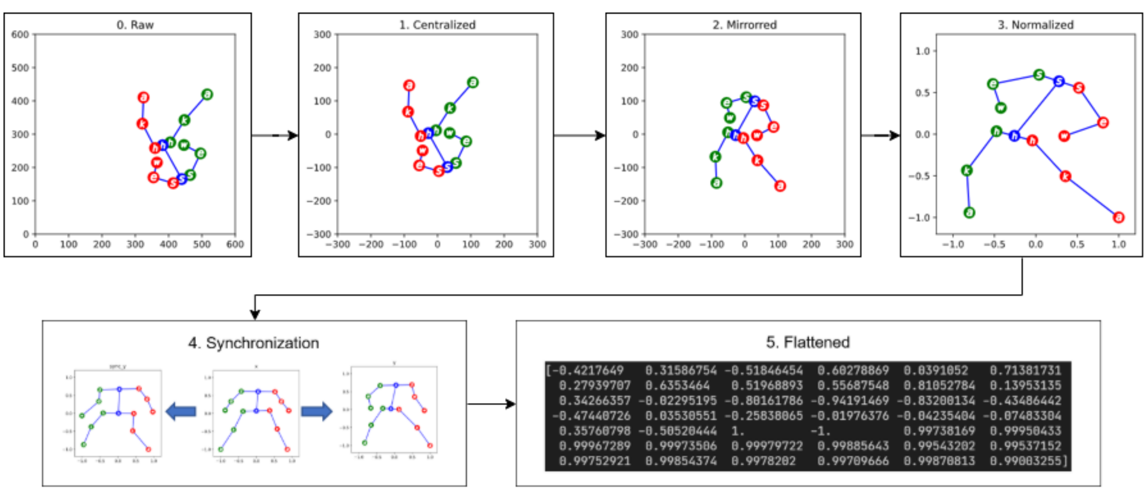

Figure 8. Preprocessing steps.

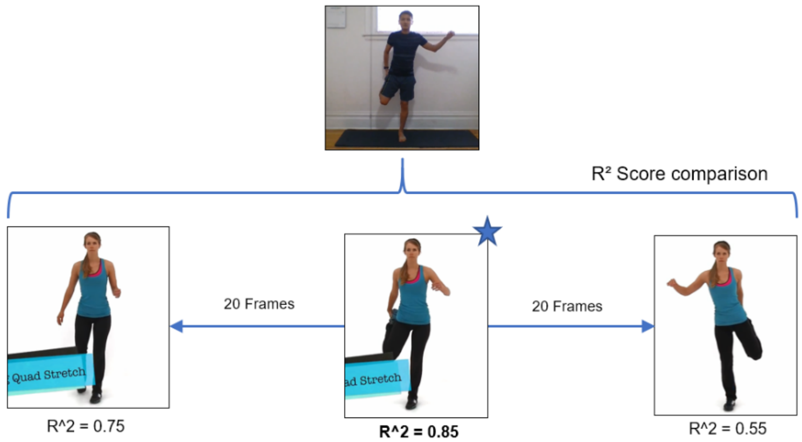

Figure 9. Synchronization example.

Table 3. Comparison between different synchronization windows for five randomly selected routines.

Window size in frames	**Processing time per frame in ms**	**R^2 score**
0	0	-0.2162
11	2.0046	-0.0863
21	3.7201	-0.0154
31	5.4241	-0.0083
41	7.1552	0.0075
51	9.7931	0.0081
61	12.4349	0.0101
71	14.1609	0.0111
81	15.8891	0.0115

The R^2 score[m], used in the pose synchronization method, is a statistical measure of how well the regression predictions approximate the real data points. In other words, it provides an indication of the goodness of fit of our model. An R^2 of 1 indicates that the regression predictions

[m] Coefficient of determination, denoted as R^2 https://en.wikipedia.org/wiki/Coefficient_of_determination.

perfectly fit the data. Still, it can also be negative if the model is worse than using the values' average as the prediction. Citing from the scikit-learn documentation[n], the R^2 equation is given by:

"If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value for total n samples, the estimated R^2 is defined as:

$$R^2(\mathrm{y}, \hat{\mathrm{y}}) = 1 - \frac{\sum_{i=1}^{n}(\mathrm{y_i} - \hat{\mathrm{y}}_i)^2}{\sum_{i=1}^{n}(\mathrm{y_i} - \bar{y})^2} \qquad (2)$$

where $\hat{\mathrm{y}} = \sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}(\mathrm{y_i} - \hat{\mathrm{y}}_i)^2 = \sum_{i=1}^{n} \epsilon^2$."

3.5.3. Training

We used the trainee features as the input (X) and the coach features as output (Y) to train the PINN. We ended up with 42 features for the trainee from the pre-processing steps: 28 from the coordinates of the fourteen 2D keypoints plus fourteen confidence scores, one from each keypoint. The coach has only 28 features from the fourteen 2D keypoints coordinates as we didn't need the confidence scores.

We used the Adam optimizer with a learning rate of 0.003 and Mean Squared Error (MSE) as the loss measurement to decrease. From the RSO analysis, we selected the 2-LNN configuration. In Figure 10, we can see a diagram of the training process.

---

[n] https://scikit-learn.org/stable/modules/model_evaluation.html#r2-score

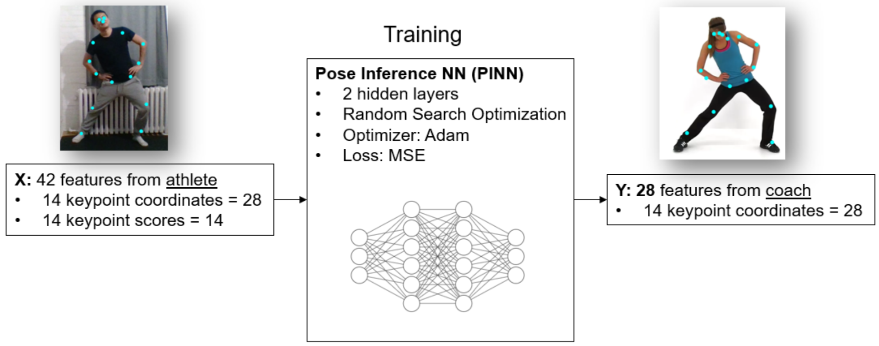

Figure 10. Training diagram.

It is essential to discuss the effect of synchronization. In Figure 11, we see the train and test loss without using synchronization. In Figure 12, we see the training results after synchronization. As expected, the loss diminished considerably, going from 0.0249 to 0.0141 after synchronization (76% improvement). In Table 3, we can also see how using synchronization improved on the R^2 for all of the window frames tested. After these tests, we can conclude that synchronization had a positive effect on the overall training results.

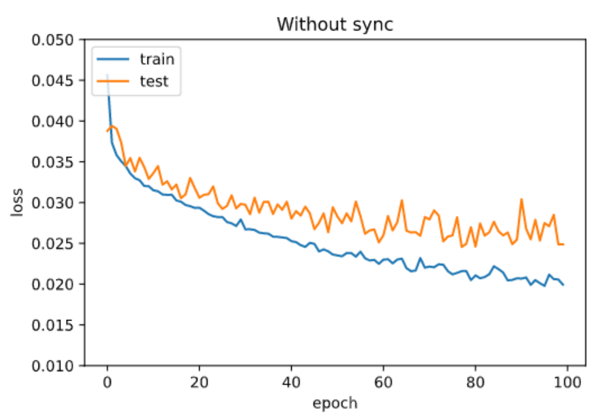

Figure 11. Train/test loss without synchronization. Final training loss: 0.0199 – Final test loss: 0.0249

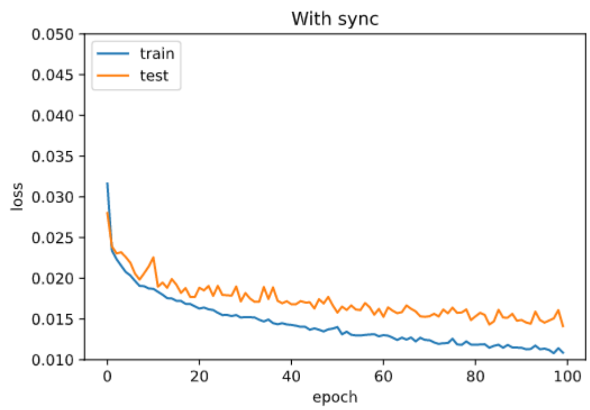

Figure 12. Train/test loss with synchronization. Final training loss: 0.0109 – Final testing loss: 0.0141

3.5.4. Data Post-processing

Post-processing's main idea is to use the PINN estimation to improve the pre-trained PoseNet model's results. To do this, we use the confidence score as a *coordinate merge factor* in a function we call the Weighted Coordinate Function (WCF). Given the pre-trained PoseNet model original coordinate c_{orig}, the original coordinate confidence score s_{orig}, and the PINN predicted coordinate c_{pred}, the fixed coordinate c_{fix} is given by:

$$c_{fix} = c_{orig}\left(s_{orig}\right) + c_{pred}\left(1 - s_{orig}\right) \tag{2}$$

The idea is to use the original confidence score of the coordinate from the pre-trained model as the deciding factor in merging the PINN and the PoseNet coordinates. In Figure 13, we can see a visualization of this process. When the original confidence score approaches 1, the fixed coordinate will approach the original pre-trained model estimation. If it approaches 0, it will resemble more the estimation from the PINN model. Either way, the fixed coordinate will always be a combination of the estimation of both models. The advantage of using this approach is that the process will only affect low confidence coordinates and will leave reliable estimations practically intact.

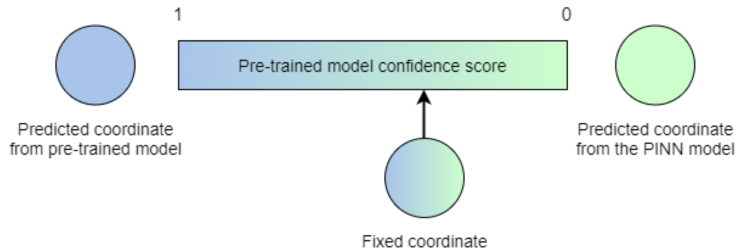

Figure 13. WCF visualization.

3.5.5. PoseNet Model Comparisons

The PoseNet library uses five different configuration parameters to select different pre-trained models:

- **Architecture:** MobileNetV1 or ResNet50. MobileNet is the lighter of the two. While ResNet is more robust, it demands more computational resources. For DTCoach, we used MobileNet for the trainee estimations, while ResNet is used for the coach estimations.
- **Input Resolution:** Size of the input image to be processed. The larger the picture, the better but slower the estimation. The default value is 257 pixels for width (and variable height). We used 320x240 pixels for all the videos to maintain uniformity for the input data.
- **Stride:** The downsampling methodology involves sliding a filter through the image, convolving the values using different types of functions. In this library, the stride can take values of 8, 16, 32. The lower the value, the slower the estimation since the model would need to process a larger image. We used a stride of 16 to reach a middle ground between performance and speed.
- **Depth Multiplier:** The depth of the convolutional layers and can be 0.5, 0.75, or 1.00, being the last value the more accurate and heavier model. This parameter does not apply to the ResNet architecture.

- **Quantization Bytes:** Quantization is an optimization process in which we convert floating-point values into integers to make the model faster and lighter at the cost of accuracy and information loss. This parameter can take three values, 1, 2, and 4, which are the number of bytes per float used, 1 being the lightest model and 4 being the heaviest.

Since we had to collect our own dataset, there is no ground truth in the data we are using for training. Therefore, we used the pose estimations from the ResNet model (the most robust configuration of PoseNet) as the guideline. Thus, our approach aims to approximate the ResNet model results using the PINN explained in chapter 3.5.

To test our approach, we used three different configurations of the MobileNet architecture with varying complexity degrees. We used model A for the DTCoach trainee application since it is the lightest of them. Model D, the heaviest, was used to process the coach video. Model B and C were only used for testing purposes. The three MobileNet models, along with the ResNet model, can be seen in Table 4.

Table 4. Different models used for testing.

Label	Model	Stride	Input Resolution	Quantization Bytes	Depth Multiplier	Used for
A	MobileNetV1	16	320x240	2	0.75	DTCoach
B	MobileNetV1	16	320x240	4	1.00	Testing
C	MobileNetV1	8	320x240	4	1.00	Testing
D	ResNet50	16	320x240	1	-	Ground Truth

We had five people record their routine in a controlled environment for testing purposes in the data collection stage. We trimmed each video to 302 seconds (3020 frames per video) and ran them through the four models to measure how much time they take to process. Results are shown in Table 5. As expected, model D took the longest to process each video, while model A was the fastest. If we want to use pose estimation in real-time, we can only use model A or B since the other models do not even reach 1 fps. This table also includes how our approach of using model A (PoseNet) and the PINN only added 4 seconds (2% increase) to the total processing time and almost no average FPS difference.

Table 5. Seconds for each video to process for the four models used.

Video	Model A	Model B	Model C	Model D	Model A + PINN
1	147	220	539	931	151
2	145	221	492	925	149
3	159	231	508	930	163
4	147	216	488	893	152
5	145	216	493	886	149
Average	149	221	504	913	153
Average FPS	2.0	1.4	0.6	0.3	2.0

We then tested our approach of using the PINN to improve the pose estimation of the models A, B, and C since D will be our ground truth. On average, our estimation improvement process took four extra seconds for the whole video with varying degrees of success. The R^2 score was used to test whether the pose estimation is close to the ResNet prediction (model D) or not. R^2 scores can be seen in Table 6.

The first thing to notice in Table 6 is that model C, the slowest of the MobileNet configurations, was the worst-performing of them all. Compared to the other models, the big difference of this model is that we used a stride of 8, which could negatively affect the pose estimation task's generalization.

Table 6. R^2 score of each video per model compared to the ResNet model (model D).

Video	Model A	PINN + Model A	Model B	PINN + Model B	Model C	PINN + Model C
1	0.152	0.306	0.225	0.355	-0.138	0.020
2	0.269	0.317	0.312	0.339	-0.260	-0.104
3	0.157	0.238	0.347	0.316	-0.077	-0.005
4	0.159	0.374	0.367	0.452	-0.179	0.016
5	0.461	0.533	0.442	0.483	-0.219	0.011
Average	0.240	0.354	0.339	0.389	-0.175	-0.012
Improvement over the model alone		48%		15%		93%

We can also see that our approach has improved the R^2 score of almost all the five videos tested, with the average being better for all three models. Another relevant point is that the improvement in pose estimation is inversely proportional to the original model's performance. In other words, the worse the original model, the better the improvement is. For the worst model of the three (A, B, and C), the improvement is 93%, then 48% to the second-best, while the more accurate model only improved 15% on average.

Overall, using an extra NN (in this case, the PINN) to improve the pose estimation results using past training data has been successful. Our approach has improved the models at least 15% and at most 93% for the cost of 4 seconds per 5 minutes video. Even if there are cases where the enhanced pose estimation is worse than the prediction of the original video (only in video 3 model B), it was a 9% decrease in accuracy. This situation can be mitigated by using the Angular Pose Representation (APR) while generating feedback.

3.6. Feedback Generation Using Angular Pose Representation (APR)

For feedback generation, we compared the trainee and the coach's pose in the same time frame. We have several issues discussed in chapter 3.5.2, like differences in body part lengths, the subject's position in the camera frame, and synchronization. We also have problems with weak estimations from the light pre-trained models. Therefore, we proceed with the coordinates generated in the data post-processing discussed in chapter 3.5.4.

The difference between the PINN model training and the feedback generation is that we cannot use the R^2 score to compare the trainee and coach poses for two reasons. The R^2 score is a measurement of how data fit a model; it calculates correlation, not similarity. Also, the R^2 score penalizes data outliers harshly; this means that even if just one keypoint is totally different between the trainee and the coach, the overall score will be affected negatively.

To solve these issues, we propose the use of the Angular Pose Representation (APR) instead of the raw keypoints, even after data-preprocessing. The advantages of using the APR to compare poses are that APR is agnostic to body proportions and body position in the frame. It is inspired by the work of Kamel et. al. [20] and involves simple calculations to avoid hogging resources from the primary pose estimation process.

Feedback generation using the APR involves three steps:

1. **Confidence Level Filtering:** Even after improving the pose estimation using the PINN model, we will have keypoints whose confidence score is low enough (approaching 0). These keypoints usually have another underlying problem, for example, keypoints of body parts not in the frame. Therefore, we filter out keypoints that are below a certain threshold of confidence. For the DTCoach application, this threshold is set to 0.8; in other words, we only use those coordinates with 80% or higher confidence.
2. **Angular Parametrization:** The idea behind the APR is to use the angles between keypoints instead of distance to avoid proportions and position issues. In this process we used the body extremities with a wide range of movement (Figure 7): wrists, elbows,

ankles, and knees. Then we calculated the angle between these extremities and several other points: the parent joint called "local parameter" and two origin points called "global parameters" (Figure 14). The local parameters' joint-parent relations are the following: wrist-elbow, elbow-shoulder, ankle-knee, and knee-hip. The two origin points for global parameters we used are the middle points between hips and shoulders. The APR is formed by all the angles obtained from the four extremity types and the local and global parameters. Therefore, we have three different angles for each extremity. This redundancy will help in case some keypoints are filtered out in the first step. For easier understandability while doing experiments, we used degrees instead of radians.

3. **Scoring:** After we obtain the Angular Pose Representation for both coach and trainee, we continue to the scoring stage. For each angle parameter, we calculate the difference between the coach and the trainee. We take a 180-degree difference as a 0% similarity and a 0-degree difference as 100% similarity. We considered the average of these similarity measurements for the 12 parameters as the frame's overall score. Finally, this score is shown to the user in real-time.

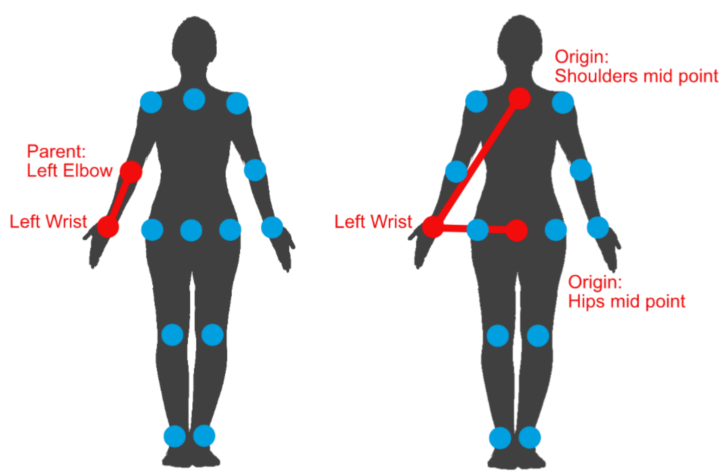

Figure 14. Left: local parameter example. Right: global parameters example.

4. Implementation

We developed a Proof of Concept (PoC) application called DTCoach to test our proposed approach, as discussed in chapter 3. This application leverages the Digital Twin ecosystem's advantages to provide a virtual coaching experience on the edge using DL pose estimation. The idea behind this PoC app is to have a real coach record a video routine for the trainee to follow. Then, on their phone or computer, the trainee can use the device's camera to record themselves following the training and get real-time feedback on their performance.

4.1. Design Overview

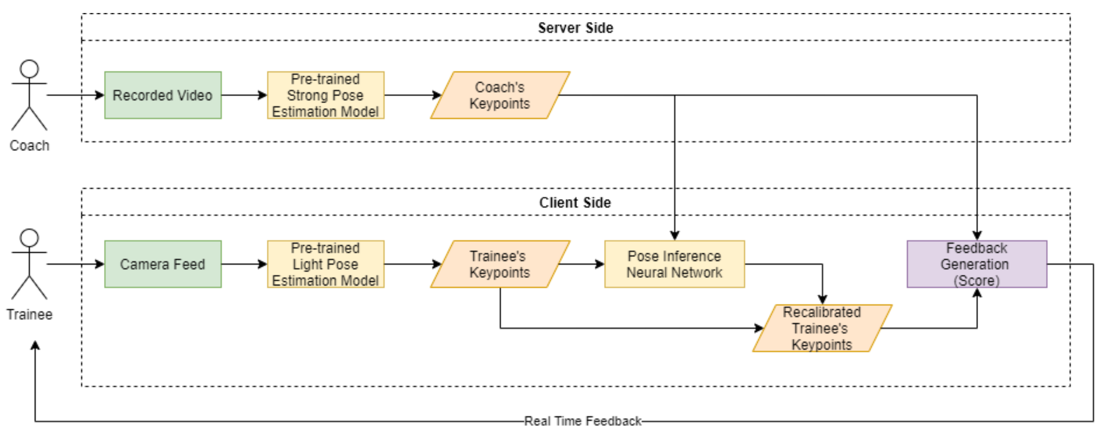

Figure 15. DTCoach workflow.

In Figure 15, we can observe the workflow of this application. We have two primary environments: server and client. In this case, the client-side is the real-time component of the workflow as it interacts with the trainee, while any server-side process can be done beforehand (offline).

Let us start with the server-side where the coach analysis takes place. Before anything else, the coach records a routine video that will be processed using a pre-trained robust pose estimation model (ResNet). The resulting keypoints will then be saved into a file that will be accessed on the client-side.

On the client-side, the trainee uses the DTCoach app on their phone or computer. First, the trainee selects the routine to perform. Then, the trainee will stand up in front of the camera while performing the exercise. The camera feed will be passed to a pre-trained light pose estimation model to obtain the current pose keypoints. We use a light model (PoseNet) to run this process in real-time, so we need to keep it as fast as possible.

The trainee and the coach's keypoints will be used to train the PINN. The idea behind the PINN is to learn which coach pose the trainee is trying to imitate; more on this in chapter 3.5. The predictions of the PINN are used to recalibrate low confidence trainee keypoints (chapter 3.5.4). Finally, using both the coach and the recalibrated trainee keypoints, we generate a real-time score for the trainee as feedback on their performance. At the end of the workout, the trainees can see their overall performance in a graph, select another routine, or choose to answer a quick usability test. Figure 16 presents the DTCoach proof of concept application screens.

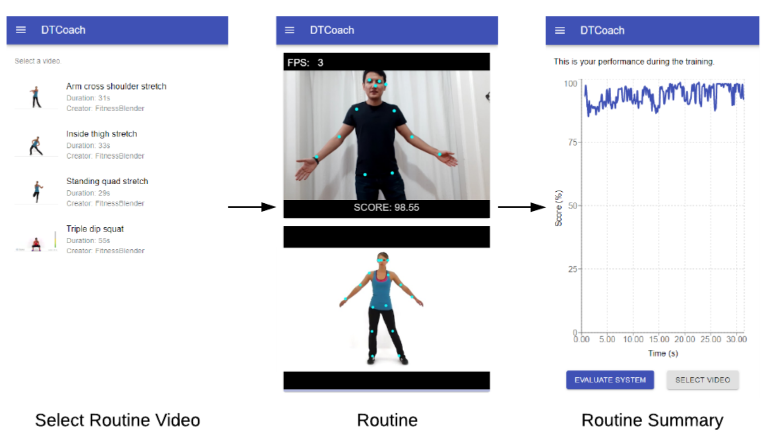

Figure 16. DTCoach proof of concept application screens.

4.2. Technical Implementation

Portability, or the ability to be run in most modern smart devices (smartphones and PCs) with access to a camera, is one of the DTCoach app's pillars. We also will be taking advantage of these devices' ability to run complex JavaScript web apps from within common browsers such as Chrome. For these two reasons, we chose React.js as the frontend framework for DTCoach.

To make the app more accessible for as many people as possible, we translated the app, excluding the coaching videos' audio, to four different languages: English, French, Chinese, and Spanish. The library we used for this, called i18next, allows JSON files as a dictionary for the web app's labels.

With the objective to maintain uniformity within the complete project, we chose to use Node.js as the back-end framework to keep using JavaScript server-side. The express library was the one selected for API management. Completing the MERN stack[o], MongoDB was the database chosen for its superb compatibility in Node.js projects as well as innate support for JSON/JavaScript objects.

For privacy and security reasons, the DTCoach app implements an account module using email and password as credentials. This password is encrypted in the database using the Node bcrypt[p] library. All personal information collected by the application (name, age, sex, and body proportions) is also encrypted using the same library. Access to view and edit this data is only possible to the user using their password. In case the user chooses to register with their Google account, a feature available in this app, the password will also be asked to be provided so we can

---

[o] For more: https://www.mongodb.com/mern-stack
[p] For more: https://en.wikipedia.org/wiki/Bcrypt

keep the same security level for their information. In the case the user forgets their password, a mailing module was developed to send a link with the instructions needed to reset it.

In Figure 17, we can see the hosting environment of the DTCoach project. For the backend, we used the free tier of Heroku[q] for its facility in deployment using Git. The database is currently being hosted in MongoDB Atlas[r], which allows a 512MB free tier, perfect for proof-of-concept apps like our case. The front-end web application is being hosted in an MCRLab owned server provided by Digital Ocean[s].

For the DTCoach environment's smart module, we have the pose estimation model architecture of PoseNet + PINN. The latter being trained initially in Python 3 to speed up the testing process described in chapter 3. The PINN model was designed using TensorFlow 2.x and Keras, later exported to TensorFlow.js for the React.js frontend.

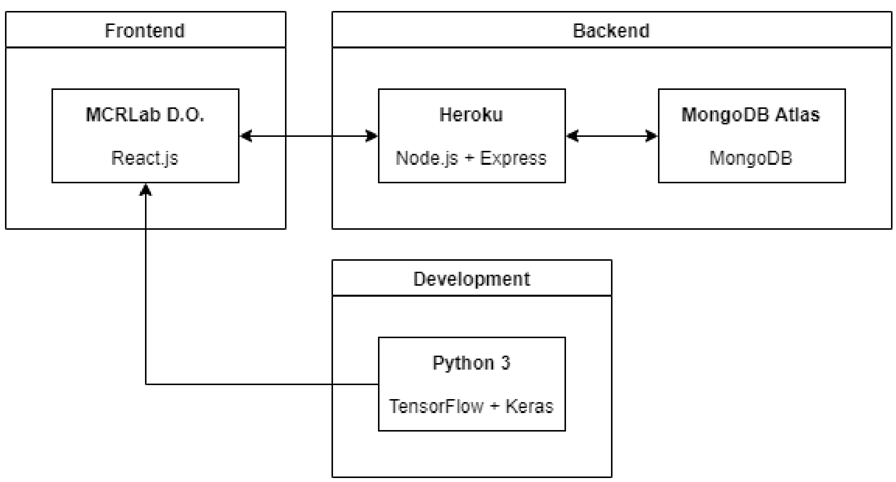

Figure 17. DTCoach hosting environment.

---

[q] https://www.heroku.com/
[r] https://www.mongodb.com/cloud/atlas
[s] https://www.digitalocean.com/

5. Usability Studies

To have a tangible result on how our POC app would behave in the real world, we ran an anonymous System Usability Scale (SUS) test throughout one week. The SUS is a 10-item Likert Scale questionnaire designed by John Brooke with the objective to evaluate different attributes of a system such as effectiveness, efficiency, and satisfaction [71]. Each question in the SUS test counts towards the overall score ranging from 0 to 100, having 80 points or more considered positive.

We chose to use this predesigned set of questions since it has proved useful for many other systems since its inception in 1996. We also asked few optional personal questions for anonymous profiling. Questions regarding home training while in quarantine were also added to the questionnaire to test if our system has helped the participants in these difficult times.

5.1. Questionnaire

The test is composed of the following 17 questions:

A. Optional Profiling Questions
    1. Email
    2. Age range
    3. Sex
    4. How many times a week do you perform physical exercise?

B. Usability Questions (5-Likert Scale)
    1. I think that I would like to use the DTCoach system frequently.
    2. I found the DTCoach system unnecessarily complex.
    3. I thought the DTCoach system was easy to use.

4. I think that I would need assistance to be able to use the DTCoach system.
5. I found the various functions in the DTCoach system were well integrated.
6. I thought there was too much inconsistency in the DTCoach system.
7. I would imagine that most people would learn to use the DTCoach system very quickly.
8. I found the DTCoach system very cumbersome/awkward to use.
9. I felt very confident using the DTCoach system.
10. I needed to learn a lot of things before I could get going with the DTCoach system.

C. Quarantine Questions

1. Do you think the DTCoach app is flexible enough to be used while you are at home?
2. Do you think a system like this motivates people to exercise in their homes during confinement?

D. Provide any comments about the DTCoach system if you wish to do so.

We deployed our POC app to a public URL and launched a social media campaign looking for participants through Facebook, Twitter, and Instagram. Emails to other MCRLab members were also sent asking for help in testing the DTCoach app.

5.2. Participants Profiles

After one week of the social media campaign, we ended up with 43 total participants. Table 7, Table 8, and Table 9 present the general anonymous profiling information of the people who answered the test.

Table 7. Usability test age distribution.

Age Range	Quantity	Representation Percentage
11 ~ 20	3	7%
21 ~ 30	16	37%
31 ~ 40	11	26%
41 ~ 50	2	5%
51 ~ 60	3	7%
NA	8	19%

Table 8. Usability test sex distribution.

Gender	Quantity	Representation Percentage
Female	16	37%
Male	19	44%
Other	8	19%

Table 9. Usability test weekly exercise frequency distribution.

Weekly Exercise Frequency	Quantity	Representation Percentage
1 ~ 2	10	23%
3 ~ 4	13	30%
5+	5	12%
NA	15	35%

We can see that the bulk of the participants were adults between the ages of 21 to 40, but we still had three young people and, surprisingly, five mature adults. It is not surprising that we did not have anyone older than 61 years by the nature of the routine's squatting exercises. It is also good to know that we reached a good enough balance in terms of sex representation, having 44% male, 37% female, and 19% who chose not to answer.

One interesting result is found in Table 9, where 65% of participants answered that they perform a physical exercise at least once per week, even when there is a social lockdown in most countries. Even though we had an option for 0 weekly exercises, none of the participants seem to have selected it, which is not surprising since people should at least be already interested in light exercises such as stretching and squatting.

It is important to note that we had at least eight people, 19% of the test population, not interested in answering personal questions such as age and sex, with even more choosing not to disclose their activity-related habits. It is understandable as people are generally wary of revealing such private information with third parties even if their name is not attached to the data. This enforces our points discussed in chapter 3 about having privacy and security as a pillar in the DT and DTC ecosystem.

5.3. System Usability Scale Results

As we can see in Figure 18, the overall impression of the app was positive. The highlights of the scores obtained are the following:

- We had an average score of 82.91. The minimum being 47.2, and the maximum being 100.
- From the 43 people surveyed, 29 scored the app's usability as 80 points or more (67% of the population).

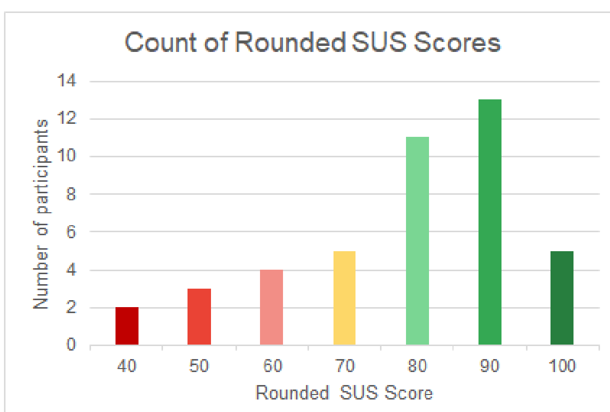

Figure 18. Rounded SUS Scores.

Most of the comments we got from participants who scored above 80 were about how easy the app was to use and that they found interesting the use of AI/ML in this field. Examples:

- "Very easy to use and make me feel confident" – Participant with score 97.5.
- "I find interesting the Machine Learning integration." – Participant with score 87.5.

We had some valid feedback comments from some participants, even if they valued the app usability as high. The issue boils down to the device they were using to test the app. Even if we never mentioned that it was preferable to use a smartphone to test the app, most of the feedback comments were related to using a small screen. They noted that it was difficult to follow the video while standing a few meters away. Examples:

- "Include the optimal distance I should have from the camera to be tracked" – Participant with score 97.5.
- "The distance may be kind of far that I cannot see the video clearly. And short distance makes the camera not catch the whole person." – Participant with score 62.5.

Regarding this issue, the only possible solution without changing the approach of showing both the coach video and the trainee camera is to give the option of hiding the latter. This way, the coach video can take the whole screen so the trainee can watch it better from afar if using a small device. We can also draw the trainee keypoints skeleton over the coach video to still compare their pose. There are also more options that may not be favorable for modern users, such as using a bigger device (laptop or tablet), an external camera, or a second monitor.

The other things mentioned were simple quality of life improvements but did not follow a pattern like the comments mentioned above. They were related to specific app features, like modifying the video screen selection or adding more information about the performance graph we show at the end. We can include many other features, like adding 3D automatic modeling, AR/VR, etc.

In Figure 19 and Table 10, we breakdown the results for each of the usability chapter questions. The SUS is designed to have one positive question (where 5 is the best answer) and then one negative question (where 1 is the best answer). Two questions are relevant to highlight, the first being: “1. I think that I would like to use the DTCoach system frequently”. This question, on average, contributed fewer points to the overall score. The best-scored question was “7. I would imagine that most people would learn to use the DTCoach system very quickly.”

Question 1 main measures the value of the system for the users' daily lives. Having this question scored the lowest on average (7.5 points, which is by no means a bad score) implies that people thought that the current version of the DTCoach application might not provide enough value for them to use frequently. This can be solved by working on two factors: adding more features and solving the app's issue with small screens, which was discussed above.

Question 7 is about how the participants perceive the DTCoach app's usability for other people. Having this question scored the best means that even if the participants feel that they still consider the app, it is easy to use for people who may not have the same knowledge level.

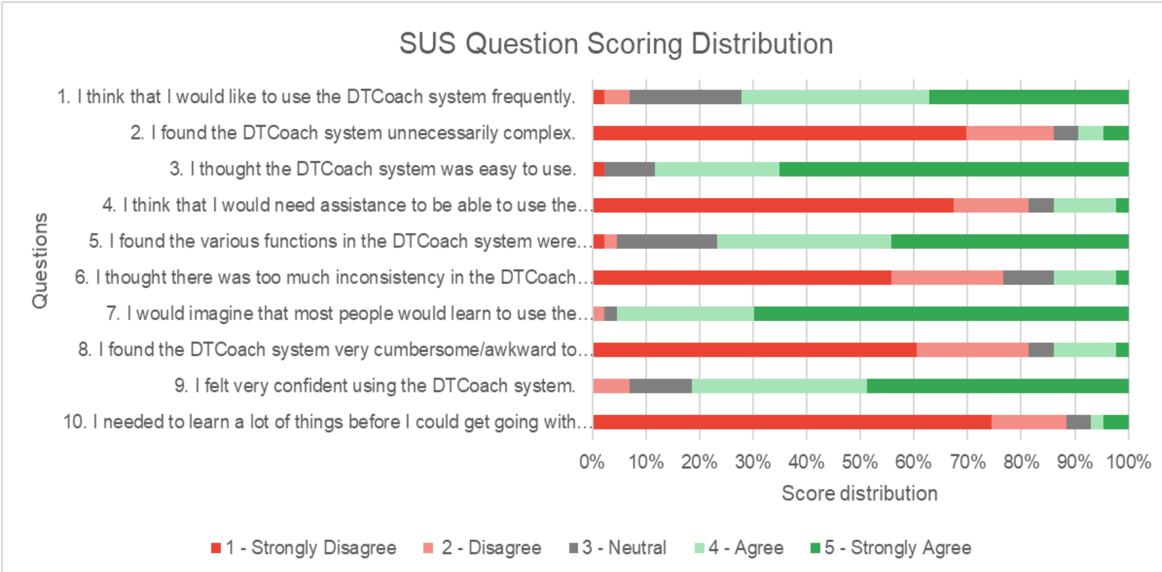

Figure 19. SUS Single Question Score Distribution.

Table 10. Count for each selected option per question.

Question number	1 - Strongly Disagree	2 - Disagree	3 - Neutral	4 - Agree	5 - Strongly Agree
1	1	2	9	15	16
2	30	7	2	2	2
3	1	0	4	10	28
4	29	6	2	5	1
5	1	1	8	14	19
6	24	9	4	5	1
7	0	1	1	11	30
8	26	9	2	5	1
9	0	3	5	14	21
10	32	6	2	1	2

We should also discuss the issue of selection bias. This usability study was announced on social media (Facebook, Instagram, and Twitter), and interested parties were directed to the web application URL. Therefore, we may have a selection bias situation, where the participants may be already physically active younger people. This group is naturally more attracted and comfortable with novel applications or well versed in sports/coaching-related applications.

5.4. Digital Twin Coaching for a Quarantine

We also included two questions about using the application while in quarantine caused by COVID19. The first question inquired about the app usage at home. This question got an average rating of 4.60 out of 5. This positive score means that participants found the app convenient enough to be used in closed spaces like rooms. It also confirms that DTCoach can be used in everyday devices like smartphones or personal computers.

The second question got a lower score of 4.28 out of 5. This question is about the adoptability of the DTCoach application while in quarantine. It’s not surprising that we got a lower score this time as this question is linked to point no 7 of the SUS questionnaire, both talking about the perceived value of the application to the participants. Adding more features and improving the routine video training visibility is already in the future roadmap.

6. Conclusions, Limitations, and Future Work

In this thesis, we proposed a Smart Coaching system from the perspective of Edge Computing. There are hard limitations on the hardware and computational resources of edge devices (smartphones). Therefore, we considered the tradeoff between real-life performance and accuracy.

We proposed a Digital Twin Coaching architecture specialized in Edge Computing Environments. To achieve this, we designed a novel DL approach using a pre-trained light pose estimation model along with a relatively shallow NN (PINN). The objective was to fine-tune the model to physical exercise routines in this case. The video routine processing is automatic, letting the coach plan according to their trainees' needs. The coach can adapt the training to include standing up positions such as stretching, yoga, weightlifting, squatting, etc.

The system also considers that every person has different body proportions. We proposed an approach comparing the joints' angles instead of the joints' position to compare the trainee and the coach pose while training. This approach, called Angular Pose Representation, has many benefits as it does not deal with distances but with angles within the joints. It is agnostic to body proportions like height or extremity length and ignores the position of the body within the camera frame. Since it is a set of simple angle calculations, it affects the system's time complexity very little.

We also developed a proof-of-concept web application called DTCoach following the architecture described. Using this application, we performed a usability study for one week using social media to recruit participants. There are two main takeaways from this usability study. While the participants found the application easy to use and engaging, they also expressed, by their answers to the SUS questionnaire, that the app should provide more value for them to consider using it on a daily basis.

From the development of the DTCoach application and the DTC architecture design, we encountered several limitations that are important to discuss. The first limitation is related to using the combination of two light NN for pose estimation: PoseNet + PINN. Our PINN model takes the output of PoseNet and tries to improve upon it using the context of the physical exercise being analyzed. Therefore, if the base pose estimation model performs poorly, it will negatively affect the coaching process. The second limitation is that the PINN model is trained using scenarios where the person uses the app in real life. We depend on the participants correctly since the wrong usage can affect the PINN model's estimation. Third, we tested our system with ten subjects; this number needs to be enhanced to reduce potential selection bias and improve the PINN model.

Considering these limitations, there is a lot of work to be done if we want to improve our system. From the usability study, we noticed participants wanted the DTCoach app to provide more value. The first step would be working with HR sensors so we can measure the participant biometrics. More sensors should give the ability to provide more accurate feedback to modify future routines if necessary. Including haptics can also enhance the coaching experience as the trainee can focus while their sight is away from the screen without losing the routine grasp. It can also be easily included using smart wearables like a smartwatch.

Another interesting path we can take is designing a framework for trainee-coach data collection as we could not find any useful dataset to train Smart Coaching models. The only datasets found were sports-related without any professional input. Regarding ML and DL models, we can also try different architectures of NN and different ML approaches that could provide good and quick results since we are working with hardware limitations.

References

[1] A. El Saddik, ‘Digital Twins: The Convergence of Multimedia Technologies’, *IEEE Multimed.*, vol. 25, no. 2, pp. 87–92, Apr. 2018, doi: 10.1109/MMUL.2018.023121167.

[2] M. Grieves, ‘Virtually Intelligent Product Systems: Digital and Physical Twins’, in *Complex Systems Engineering: Theory and Practice*, American Institute of Aeronautics and Astronautics, 2019, pp. 175–200.

[3] R. Piascik, J. Vickers, D. Lowry, S. Scotti, J. Stewart, and A. Calomino, ‘Technology area 12: Materials, structures, mechanical systems, and manufacturing road map’, *NASA Off. Chief Technol.*, 2010.

[4] K. Panetta, ‘5 Trends Drive the Gartner Hype Cycle for Emerging Technologies, 2020’, Aug. 18, 2020. //www.gartner.com/smarterwithgartner/5-trends-drive-the-gartner-hype-cycle-for-emerging-technologies-2020/ (accessed Nov. 24, 2020).

[5] ‘Digital Twin Market Size Expected to Grow at a CAGR over 42.54% from 2018 to 2025 | MRFR - News and Updates’. https://www.marketresearchfuture.com/press-release/digital-twin-market (accessed Nov. 24, 2020).

[6] ‘Top 10 Digital Twin Vendors for 2019 | EM360’. https://em360tech.com/business_agility/tech-news/top-10-digital-twin-vendors (accessed Nov. 27, 2020).

[7] N. Bagaria, F. Laamarti, H. F. Badawi, A. Albraikan, R. A. M. Velazquez, and A. El Saddik, ‘Health 4.0: Digital twins for health andwell-being’, in *Connected Health in Smart Cities*, 2019, pp. 143–152.

[8] C. Scheuermann, T. Binderberger, N. von Frankenberg, and A. Werner, ‘Digital twin: a machine learning approach to predict individual stress levels in extreme environments’, in *Adjunct*

Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers, New York, NY, USA, Sep. 2020, pp. 657–664, doi: 10.1145/3410530.3414316.

[9] M. Nikitina and I. Chernukha, ‘Personalized nutrition and “digital twins” of food’, *Potravinarstvo Slovak J. Food Sci.*, vol. 14, pp. 264–270, 2020, doi: 10.5219/1312.

[10] L. V. Calderita, A. Vega, S. Barroso-Ramírez, P. Bustos, and P. Núñez, ‘Designing a cyber-physical system for ambient assisted living: A use-case analysis for social robot navigation in caregiving centers’, *Sens. Switz.*, vol. 20, no. 14, pp. 1–25, 2020, doi: 10.3390/s20144005.

[11] B. R. Barricelli, E. Casiraghi, J. Gliozzo, A. Petrini, and S. Valtolina, ‘Human Digital Twin for Fitness Management’, *IEEE Access*, vol. 8, pp. 26637–26664, 2020, doi: 10.1109/ACCESS.2020.2971576.

[12] L. Davis, R. Appleby, P. Davis, M. Wetherell, and H. Gustafsson, ‘The role of coach-athlete relationship quality in team sport athletes’ psychophysiological exhaustion: implications for physical and cognitive performance’, *J. Sports Sci.*, vol. 36, no. 17, pp. 1985–1992, Sep. 2018, doi: 10.1080/02640414.2018.1429176.

[13] G. A. Mageau and R. J. Vallerand, ‘The coach–athlete relationship: a motivational model’, *J. Sports Sci.*, vol. 21, no. 11, pp. 883–904, Nov. 2003, doi: 10.1080/0264041031000140374.

[14] R. Gámez Díaz, Q. Yu, Y. Ding, F. Laamarti, and A. El Saddik, ‘Digital Twin Coaching for Physical Activities: A Survey’, *Sensors*, vol. 20, no. 20, Art. no. 20, Jan. 2020, doi: 10.3390/s20205936.

[15] S. Kumar Basak, M. Wotto, and P. Bélanger, ‘E-learning, M-learning and D-learning: Conceptual definition and comparative analysis’, *E-Learn. Digit. Media*, vol. 15, no. 4, pp. 191–216, Jul. 2018, doi: 10.1177/2042753018785180.

[16] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, ‘Recommender system application developments: A survey’, *Decis. Support Syst.*, vol. 74, pp. 12–32, Jun. 2015, doi: 10.1016/j.dss.2015.03.008.

[17] A. Kamel, B. Liu, P. Li, and B. Sheng, ‘An Investigation of 3D Human Pose Estimation for Learning Tai Chi: A Human Factor Perspective’, *Int. J. Human–Computer Interact.*, vol. 35, no. 4–5, pp. 427–439, Mar. 2019, doi: 10.1080/10447318.2018.1543081.

[18] F. Hülsmann, J. P. Göpfert, B. Hammer, S. Kopp, and M. Botsch, ‘Classification of Motor Errors to Provide Real-Time Feedback for Sports Coaching in Virtual Reality — A Case Study in Squats and Tai Chi Pushes’, *Comput. Graph.*, vol. 76, pp. 47–59, Nov. 2018, doi: 10.1016/j.cag.2018.08.003.

[19] A. Yasser, D. Tariq, R. Samy, M. Allah, and A. Atia, ‘Smart Coaching: Enhancing Weightlifting and Preventing Injuries’, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, 2019, doi: 10.14569/IJACSA.2019.0100789.

[20] M. Jian, S. Zhang, L. Wu, S. Zhang, X. Wang, and Y. He, ‘Deep key frame extraction for sport training’, *Neurocomputing*, vol. 328, pp. 147–156, Feb. 2019, doi: 10.1016/j.neucom.2018.03.077.

[21] B. Bačić, ‘Towards the next generation of exergames: Flexible and personalised assessment-based identification of tennis swings’, *ArXiv180406948 Cs*, Apr. 2018, Accessed: Jan. 07, 2020. [Online]. Available: http://arxiv.org/abs/1804.06948.

[22] U. Jensen, F. Prade, and B. M. Eskofier, ‘Classification of kinematic swimming data with emphasis on resource consumption’, in *2013 IEEE International Conference on Body Sensor Networks*, May 2013, pp. 1–5, doi: 10.1109/BSN.2013.6575501.

[23] M. Kobayashi *et al.*, ‘Swimming motion classification for coaching system by using a sensor device’, *ICIC Express Lett. Part B Appl.*, vol. 9, no. 3, pp. 209–217, Mar. 2018.

[24] L. Zhang, J.-C. Hsieh, T.-T. Ting, Y.-C. Huang, Y.-C. Ho, and L.-K. Ku, ‘A Kinect based Golf Swing Score and Grade System using GMM and SVM’, in *2012 5th International Congress on Image and Signal Processing*, Oct. 2012, pp. 711–715, doi: 10.1109/CISP.2012.6469827.

[25] E. Wu and H. Koike, ‘FuturePose - Mixed reality martial arts training using real-time 3D human pose forecasting with a RGB camera’, in *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, Mar. 2019, pp. 1384–1392, doi: 10.1109/WACV.2019.00152.

[26] A. Silacci, O. A. Khaled, E. Mugellini, and M. Caon, ‘Designing an e-Coach to Tailor Training Plans for Road Cyclists’, in *Human Systems Engineering and Design II*, Cham, 2020, pp. 671–677, doi: 10.1007/978-3-030-27928-8_102.

[27] P. Pullen and W. Seffens, ‘Machine learning gesture analysis of yoga for exergame development’, *IET Cyber-Phys. Syst. Theory Appl.*, vol. 3, no. 2, pp. 106–110, 2018, doi: 10.1049/iet-cps.2017.0027.

[28] E. W. Trejo and P. Yuan, ‘Recognition of Yoga Poses Through an Interactive System with Kinect Device’, in *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*, Jun. 2018, pp. 1–5, doi: 10.1109/ICRAS.2018.8443267.

[29] M. Z.-E.-A. Amrani and N. Achour, ‘Serious Game based on Skeleton Shape Matching for Functional Rehabilitation Exercises’, in *Proceedings of the 2019 3rd International Conference on Automation, Control and Robots*, Prague, Czech Republic, Oct. 2019, pp. 113–118, doi: 10.1145/3365265.3365279.

[30] M. Gochoo *et al.*, ‘Novel IoT-Based Privacy-Preserving Yoga Posture Recognition System Using Low-Resolution Infrared Sensors and Deep Learning’, *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7192–7200, Aug. 2019, doi: 10.1109/JIOT.2019.2915095.

[31] P. Kumar, R. Saini, M. Yadava, P. P. Roy, D. P. Dogra, and R. Balasubramanian, ‘Virtual trainer with real-time feedback using kinect sensor’, in *2017 IEEE Region 10 Symposium (TENSYMP)*, Jul. 2017, pp. 1–5, doi: 10.1109/TENCONSpring.2017.8070063.

[32] D. D. Bellamy and P. Caleb-Solly, ‘Collaborative HRI and Machine Learning for Constructing Personalised Physical Exercise Databases’, in *Towards Autonomous Robotic Systems*, Cham, 2019, pp. 209–220, doi: 10.1007/978-3-030-23807-0_18.

[33] ‘World Population Prospects: The 2017 Revision | Multimedia Library - United Nations Department of Economic and Social Affairs’, Jun. 21, 2017. https://www.un.org/development/desa/publications/world-population-prospects-the-2017-revision.html (accessed Dec. 14, 2020).

[34] S. P. Erdeniz, A. Menychtas, I. Maglogiannis, A. Felfernig, and T. N. T. Tran, ‘Recommender systems for IoT enabled quantified-self applications’, *Evol. Syst.*, Oct. 2019, doi: 10.1007/s12530-019-09302-8.

[35] K. B. Lasater *et al.*, ‘Chronic hospital nurse understaffing meets COVID-19: an observational study’, *BMJ Qual. Saf.*, 2020, doi: 10.1136/bmjqs-2020-011512.

[36] J. Liang, H. Zhang, Y. Liu, H. Wang, J. Li, and L. Zhang, ‘A Frequency Boosting Method for Motor Imagery EEG Classification in BCI-FES Rehabilitation Training System’, in *Advances in Neural Networks – ISNN 2013*, Berlin, Heidelberg, 2013, pp. 284–291, doi: 10.1007/978-3-642-39068-5_35.

[37] H. Zhang, J. Liang, Y. Liu, H. Wang, and L. Zhang, ‘An Iterative Method for Classifying Stroke Subjects’ Motor Imagery EEG Data in the BCI-FES Rehabilitation Training System’, in *Foundations and Practical Applications of Cognitive Systems and Information Processing*, Berlin, Heidelberg, 2014, pp. 363–373, doi: 10.1007/978-3-642-37835-5_32.

[38] X. Yang, S.-C. Yeh, J. Niu, Y. Gong, and G. Yang, ‘Hand Rehabilitation Using Virtual Reality Electromyography Signals’, in *2017 5th International Conference on Enterprise Systems (ES)*, Sep. 2017, pp. 125–131, doi: 10.1109/ES.2017.27.

[39] M. H. Lee, ‘A Technology for Computer-Assisted Stroke Rehabilitation’, in *Proceedings of the 2018 Conference on Human Information Interaction&Retrieval - IUI 18*, Tokyo, Japan, 2018, pp. 665–666, doi: 10.1145/3172944.3173146.

[40] S. Guo, Z. Wang, and J. Guo, ‘Study on Motion Recognition for a Hand Rehabilitation Robot Based on sEMG Signals’, in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2019, pp. 1061–1066, doi: 10.1109/ICMA.2019.8816497.

[41] Z. Wang and Z. Ding, ‘Rehabilitation system for children with cerebral palsy based on body vector analysis and GMFM-66 standard’, *J. Phys. Conf. Ser.*, vol. 1325, p. 012088, Oct. 2019, doi: 10.1088/1742-6596/1325/1/012088.

[42] T. S. Huang, ‘Computer Vision: Evolution And Promise’, presented at the 19th CERN School of Computing, Egmond aan Zee, The Netherlands, 1996, doi: 10.5170/CERN-1996-008.21.

[43] ‘Pose estimation | TensorFlow Lite’, *TensorFlow*. https://www.tensorflow.org/lite/models/pose_estimation/overview (accessed Mar. 26, 2020).

[44] A. Toshev and C. Szegedy, ‘DeepPose: Human Pose Estimation via Deep Neural Networks’, Jun. 2014, pp. 1653–1660, doi: 10.1109/CVPR.2014.214.

[45] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, ‘Efficient Object Localization Using Convolutional Networks’, *2015 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 648–656, Jun. 2015, doi: 10.1109/CVPR.2015.7298664.

[46] A. Newell, K. Yang, and J. Deng, ‘Stacked Hourglass Networks for Human Pose Estimation’, in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 483–499, doi: 10.1007/978-3-319-46484-8_29.

[47] A.-I. Popa, M. Zanfir, and C. Sminchisescu, ‘Deep Multitask Architecture for Integrated 2D and 3D Human Sensing’, *ArXiv170108985 Cs*, Jan. 2017, Accessed: Dec. 06, 2020. [Online]. Available: http://arxiv.org/abs/1701.08985.

[48] J. Martinez, R. Hossain, J. Romero, and J. J. Little, ‘A simple yet effective baseline for 3d human pose estimation’, *2017 IEEE Int. Conf. Comput. Vis.*, pp. 2659–2668, Aug. 2017, doi: 10.1109/ICCV.2017.288.

[49] G. Papandreou *et al.*, ‘Towards Accurate Multi-person Pose Estimation in the Wild’, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 3711–3719, doi: 10.1109/CVPR.2017.395.

[50] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, ‘PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model’, in *Computer Vision – ECCV 2018*, Cham, 2018, pp. 282–299, doi: 10.1007/978-3-030-01264-9_17.

[51] R. Kurose, M. Hayashi, T. Ishii, and Y. Aoki, ‘Player pose analysis in tennis video based on pose estimation’, in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Jan. 2018, pp. 1–4, doi: 10.1109/IWAIT.2018.8369762.

[52] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, 'Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields', in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1302–1310, doi: 10.1109/CVPR.2017.143.

[53] J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu, 'AI Coach: Deep Human Pose Estimation and Analysis for Personalized Athletic Training Assistance', in *Proceedings of the 27th ACM International Conference on Multimedia*, New York, NY, USA, Oct. 2019, pp. 374–382, doi: 10.1145/3343031.3350910.

[54] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, 'Edge Computing: Vision and Challenges', *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[55] J. Chen and X. Ran, 'Deep Learning With Edge Computing: A Review', *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019, doi: 10.1109/JPROC.2019.2921977.

[56] R. Martinez-Velazquez, R. Gamez, and A. E. Saddik, 'Cardio Twin: A Digital Twin of the Human Heart Running on the Edge', in *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, Jun. 2019, pp. 1–6, doi: 10.1109/MeMeA.2019.8802162.

[57] P. Giammatteo, F. V. Fiordigigli, L. Pomante, T. D. Mascio, and F. Caruso, 'Age Gender Classifier for Edge Computing', in *2019 8th Mediterranean Conference on Embedded Computing (MECO)*, Jun. 2019, pp. 1–4, doi: 10.1109/MECO.2019.8760160.

[58] K. Simonyan and A. Zisserman, 'Very Deep Convolutional Networks for Large-Scale Image Recognition', *ArXiv14091556 Cs*, Apr. 2015, Accessed: Dec. 09, 2020. [Online]. Available: http://arxiv.org/abs/1409.1556.

[59] W.-H. Chen, H.-Y. Kuo, Y.-C. Lin, and C.-H. Tsai, ‘A Lightweight Pedestrian Detection Model for Edge Computing Systems’, in *Distributed Computing and Artificial Intelligence, 17th International Conference*, Cham, 2021, pp. 102–112, doi: 10.1007/978-3-030-53036-5_11.

[60] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘You Only Look Once: Unified, Real-Time Object Detection’, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[61] C. Liu, X. Wang, J. Ni, Y. Cao, and B. Liu, ‘An Edge Computing Visual System for Vegetable Categorization’, in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec. 2019, pp. 625–632, doi: 10.1109/ICMLA.2019.00115.

[62] A. G. Howard *et al.*, ‘MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications’, *ArXiv170404861 Cs*, Apr. 2017, Accessed: Feb. 03, 2020. [Online]. Available: http://arxiv.org/abs/1704.04861.

[63] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, ‘MobileNetV2: Inverted Residuals and Linear Bottlenecks’, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[64] K. He, X. Zhang, S. Ren, and J. Sun, ‘Deep Residual Learning for Image Recognition’, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[65] ‘Gartner Says Global End-User Spending on Wearable Devices to Total $52 Billion in 2020’, *Gartner*. https://www.gartner.com/en/newsroom/press-releases/2019-10-30-gartner-says-global-end-user-spending-on-wearable-dev (accessed Mar. 04, 2020).

[66] K. Schwab, *The fourth industrial revolution*. Currency, 2017.

[67] 'Health 4.0: Challenges for an Orderly and Inclusive Innovation', *IEEE Technology and Society*, Sep. 24, 2019. https://technologyandsociety.org/health-4-0-challenges-for-an-orderly-and-inclusive-innovation/ (accessed Nov. 29, 2020).

[68] A. El Saddik *et al.*, 'Dtwins: A Digital Twins Ecosystem For Health And Well-Being', *IEEE COMSOC MMTC Commun. - Front.*, vol. 14, pp. 39–46, May 2019.

[69] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, 'The CAVE: audio visual experience automatic virtual environment', *Commun. ACM*, vol. 35, no. 6, pp. 64–72, Jun. 1992, doi: 10.1145/129888.129892.

[70] H. Badawi, F. Laamarti, F. Arafsha, and A. El Saddik, 'Standardizing a Shoe Insole Based on ISO/IEEE 11073 Personal Health Device (X73-PHD) Standards', in *Information Technology and Systems*, Cham, 2019, pp. 764–778, doi: 10.1007/978-3-030-11890-7_72.

[71] J. Brooke, 'SUS-A quick and dirty usability scale', *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.