

Article

GolfMate: Enhanced Golf Swing Analysis Tool through Pose Refinement Network and Explainable Golf Swing Embedding for Self-Training

Chan-Yang Ju , Jong-Hyeon Kim  and Dong-Ho Lee * 

Department of Applied Artificial Intelligence Major in Bio Artificial Intelligence, Hanyang University, Ansan 15588, Republic of Korea; karunogi@hanyang.ac.kr (C.-Y.J.); jonghyeon@hanyang.ac.kr (J.-H.K.)

* Correspondence: dhlee72@hanyang.ac.kr

Abstract: Digital fitness has become a widely used tool for remote exercise guidance, leveraging artificial intelligence to analyze exercise videos and support self-training. This paper introduces a method for self-training in golf, a sport where automated posture analysis can significantly reduce the costs associated with professional coaching. Our system utilizes a pose refinement methodology and an explainable golf swing embedding for analyzing the swing motions of learners and professional golfers. By leveraging sequential coordinate information, we detect biased pose joints and refine the 2D and 3D human pose estimation results. Furthermore, we propose a swing embedding method that considers geometric information extracted from the swing pose. This approach enables not only the comparison of the similarity between two golf swing poses but also the visualization of different points, providing learners with specific and intuitive feedback on areas that require correction. Our experimental results demonstrate the effectiveness of our swing guide system in identifying specific body points that need adjustment to align more closely with a professional golfer's swing. This research contributes to the digital fitness domain by enhancing the accuracy of posture analysis and providing a specialized and interpretable golf swing analysis system. Our proposed system offers a low-cost and time-efficient approach for users who wish to improve their golf swing, paving the way for broader applications of digital fitness technologies in self-training contexts.



Citation: Ju, C.-Y.; Kim, J.-H.; Lee, D.-H. GolfMate: Enhanced Golf Swing Analysis Tool through Pose Refinement Network and Explainable Golf Swing Embedding for Self-Training. *Appl. Sci.* **2023**, *13*, 11227. <https://doi.org/10.3390/app132011227>

Academic Editor: Fan Zhang

Received: 30 August 2023

Revised: 10 October 2023

Accepted: 11 October 2023

Published: 12 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: digital fitness; self-training; human pose estimation; explainable representation; golf swing analysis

1. Introduction

Recently, the development of human pose estimation (HPE) technology has stimulated research in the field of digital fitness. Digital fitness is garnering attention because it allows for exercise guidance remotely through automated systems, even when face-to-face interactions between instructors and learners are challenging. In particular, there has been an active pursuit of research [1–7] in which artificial intelligence analyzes exercise videos and supports self-training for exercise without communication with a sports instructor.

Self-training is a learning method where individuals observe professional athletes' movements and mimic those actions to improve their own sports abilities. Learners observe actual sports players' movements, apply those movements to their own exercise routines, and analyze areas where they fall short. Through this process, they can self-correct and ultimately achieve movements that are similar to those of professional athletes. Systems that support this process automatically compare the movements of professional athletes and learners, and, by highlighting differences in movement, enable the learners to self-correct. Self-training is attracting attention, especially in high-cost sports fields, because it offers a more affordable way to learn. Particularly in the golf lesson market, where high costs were previously required, automated golf pose analysis technology has been introduced through image processing and human pose estimation research. This technology allows learners to

self-study the movements of professional golfers, thus making it possible to receive golf lessons at a lower cost. The introduction of such technology has the effect of reducing the cost burden for those starting golf.

In automated golf lesson systems, the key feature of self-training is utilizing human pose estimation technology to extract the swing postures of professional golfers and learners and identifying the differences in their movements to provide feedback to the learner. In this process, the swing movements of professional golfers are transformed into embedding vectors, and algorithms are used to detect discrepancies between movements.

Conventional golf posture correction studies have extracted the joint positions of golfers through HPE technology and presented the differences in movements to learners by comparing similarities between the embedding vectors of golfer swings represented through a CNN (Convolutional Neural Network). However, there are two problems with this approach.

Firstly, as golf swing movements tend to be performed at high speeds, there is a high likelihood of obtaining blurry or ambiguous images. As a result, the accuracy of the extracted coordinates in the pose estimation process may be compromised. Therefore, it is necessary to improve the accuracy of the extracted coordinates through a pose refinement process.

Secondly, the conventional image embedding representation method does not contain enough information to compare golf swing movements. The golf swing should consider geometric elements, such as shoulder rotation angle, the angle between the shoulder and elbow, and pelvic angle, as well as the physical differences of the golfer performing the swing. In prior research [2], CNNs have been utilized to embed images of golf swings, facilitating the identification of discrepancies through the computation of similarity against reference embeddings. Nonetheless, this methodology presents problems. The derived similarity might be influenced by extraneous variables, such as background variations in images or disparities in the physical stature of the subjects under comparison. Consequently, the embedding vector's similarity can be affected by factors beyond the mere golf swing dynamics. Furthermore, while the system can detect a frame-wise deviation between two motions, it lacks the precision to identify the specific joints responsible for the discrepancy. This leads to the problem of being able to tell the user in which frame the mismatch occurs but not being able to give them direction for improvement.

In this study, we introduced a method that applies a coordinate correction network to enhance the accuracy of joint coordinates obtained through pose estimation. Our approach focuses on proposing an embedding method to accurately represent a golfer's swing, utilizing features such as geometric analysis, physical characteristics, and swing style. We aim to offer a comprehensive golf analysis system that suggests improved postures to learners using template-based image and video generation. The detailed benefits and unique aspects of our approach, including specialized information representation and the provision of interpretable insights, will be expanded upon in the Method section. The key contributions of our research are as follows:

- We introduce a coordinate correction network to improve the performance of the joint coordinates extracted through human pose estimation technology, thereby enhancing the accuracy of posture analysis.
- We propose a golf swing embedding technique that allows for more accurate representation of golf swing movements, enabling specialized golf swing analysis.
- Unlike previous learning methods, we can provide specific advice for user movement correction through interpretable embedding analysis.

2. Related Works

2.1. Pose Estimation Research

In [8], the field of two-dimensional (2D) human pose estimation research was categorized into Single Person Pose Estimation models and Multi-Person Pose Estimation models. Additionally, each was further divided into regression and heatmap methods,

as well as bottom-up and top-down detection methods. Single Person Pose Estimation using the regression method detects joints by regressing joint coordinates directly from the feature map of the image. This method is fast, direct, and trained in an end-to-end fashion. Because of this, it can be applied to three-dimensional (3D) joint estimation without any change. However, it is difficult to train joint positions, and it is not applicable to multiple person pose estimation. BlazePose [9] used the regression method and employed tracking to utilize previous coordinate information for predicting the next coordinates. It is an extended model that can estimate 3D coordinates, obtaining x , y , and z coordinates. However, when using motion as input, the lower the image quality, the more non-detection issues occur in all coordinates. Single Person Pose Estimation using the heatmap method infers joint coordinates through heatmap prediction of the expected joint positions. It is easy to visualize and applicable to complex cases. However, it requires much memory to obtain heatmaps and is difficult to extend to 3D coordinate estimation. HRNet (High-Resolution Network) [10] uses the heatmap method and applies multiple resolutions in parallel during training to learn models that capture both global and local contexts. This model extracted human joint coordinates with higher performance than other models used in the experiments, but issues of false detection and coordinate inversion still existed. Bottom-up coordinate detection for multi-person estimation is a method that estimates the joint coordinates of people in a video and then distinguishes individuals. It detects coordinates quickly by first finding coordinates and then detecting individuals, but it has the disadvantage of lower performance. OpenPose [11], a bottom-up coordinate detection model, distinguishes individuals through features of body parts, thus improving accuracy. However, non-detection issues occurred when using golf swing motion data. Top-down coordinate detection for multiple person estimation is a method that first detects a person and then iteratively estimates the joint coordinates of the single person. It detects more accurate coordinates but is slower since it detects the person first and then iteratively estimates their pose. To improve coordinate estimation performance, both the human detection and detected human pose estimation parts need refinement. Geometric and spatial transformation processes using STN (Spatial Transformation Network) and SDTN (Spatial Detransformer Network) were suggested in [12]. This network [12] extracts high-quality human candidate frames and shows features that improve recognition performance. Moreover, ref. [13], currently a state-of-art model, significantly improved pose estimation performance through the Vision transformer. However, the current model size ranges from 1 million to 1 billion, which requires high computational costs.

2.2. Pose Refinement Research

Pose refinement research aimed at improving coordinate accuracy in human pose estimation models can be categorized into end-to-end methods and pose-processing methods [14]. This research [14] has focused on enhancing the accuracy of the coordinates estimated by human pose estimation models, which is one of the main topics of this paper. Models in [15–21] are studied using an end-to-end learning approach. Although the implementation methods differ among models, they share the characteristic that pose estimation and refinement occur together within the model. In the case of [19], a method was used that employs iterative error feedback, transferring errors step-by-step within the model and incrementally improving the pose by correcting the model's estimation results. Similar to other models, the step-by-step pose refinement processes are implemented together with the pose estimation process within the model. In [20], a PRM (Pose Refine Machine) is used to improve the estimated pose within the model at the last step of the pose estimation process. In this study [20], PRM makes more precise pose estimation possible by using high-level discriminative semantic information and low-level spatial information. In [21], a structure was used to estimate human pose through a two-step process. A GPR (Graph Pose Refinement) module in the second step was used to obtain an improved pose. The GPR module was designed as a refinement module with a graph structure that considers the relationship between joints. These kinds of end-to-end refinement modules rely on the

estimation results of the pose estimation model for their output values, which does not guarantee that the refinement module will work successfully within the model.

The models that use a post-processing method in performance refinement research are in [14,22,23]. These models have a network structure that corrects the coordinates by receiving the coordinates output by the human pose estimation model. Similar implementation methods are shown in [14,22], but the method proposed in [14], which showed higher performance, used a method that obtains corrected coordinates by passing the input image and coordinates through a CNN (Convolutional Neural Network) backbone + upsampler structure. In [23], research was conducted to recognize actions using extracted joints, correcting coordinates through a PRM (pose refinement module), and then utilizing them. The PRM used in [23] was designed as a structure that obtains corrected poses by passing through a GCL (Graph Convolutional Layer) and TCL (Temporal Convolutional Layer). Unlike the end-to-end methods that operate within the model, these post-processing methods are applied after the model estimates the coordinates, which reduces model dependency. However, they also present the problem of increased computational requirements due to the additional coordinate correction process.

2.3. Self-Training Research

Digital fitness can be classified in three ways depending on the participation of the user and the instructor (Figure 1). First, there is the passive approach [24], which occurs through an online assessment by a human instructor. This method involves evaluating the student's movements through real-time video calls or video submissions and providing feedback, following the model of traditional offline fitness coaching. The advantage of this approach is that it allows for expert guidance. However, it has the drawback of being dependent on human resources, as it requires the assistance of a human instructor.

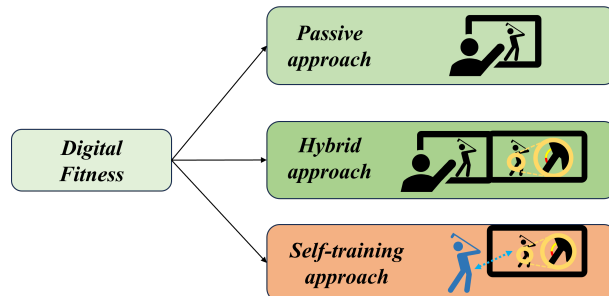


Figure 1. The branches of digital fitness research. Each approach is depicted in a unique color for differentiation: passive in green, hybrid in teal, and self-training in orange. Digital fitness research is divided into passive approaches, hybrid approaches, and self-training approaches. Passive methods involve learners learning exercises through live or recorded videos. Hybrid methods involve a combination of human instructor intervention and automated system analysis. Self-training methods involve learners watching and learning from experts without human intervention.

Second, the hybrid approach [25] allows users to receive help from both a human instructor and an automated system. This approach offers the advantage of unrestricted learning through an automated system and the ability to receive expert assessments. This method allows for more precise and varied perspectives on information, but it has the limitation of still requiring human expert participation in terms of cost savings.

Lastly, the self-training approach [1–7] is based on learners following expert movements or standardized movements on their own without the intervention of a human instructor. In this approach, learners can learn on their own through expert workout videos, and there is an approach that uses automated systems to compare the postures of experts and learners, analyze the differences, and provide guidance. The advantage of this approach is that it does not require human resources, making it cost-effective and allowing for training without time constraints. However, this method heavily relies on the learner's

will and the system’s performance, so the effectiveness of the exercise can vary depending on the individual.

3. Method

In this section, we detail our methodology designed to compare a user’s golf swing with that of a professional golfer. All of the process is illustrated in Figure 2. Our approach consists of two stages. The first stage revolves around pose estimation and its pose-processing refinement for accurate pose guidance illustrated in the HPE and PRN of Figure 2a. To achieve this, we produced pose error data by simulating incorrect poses based on real-world data. These data then inform a sequential model, which detects errors in pose joints extracted from our primary pose estimation model. In the next stage, which is illustrated in the Norm and Vector Representation of Figure 2a, we translate the swing motion into a vector form that consists of explainable feature values, such as gender, ratio, angle, etc., allowing for a direct comparison between two distinct golf swing motions. We compare the user’s swing embedding vector with the pro’s embedding to be aware of most similar pro golfers and to detect the discrepancy joint for self-learning. By visualizing the differences between these motions, we aim to facilitate the user’s self-guided learning.

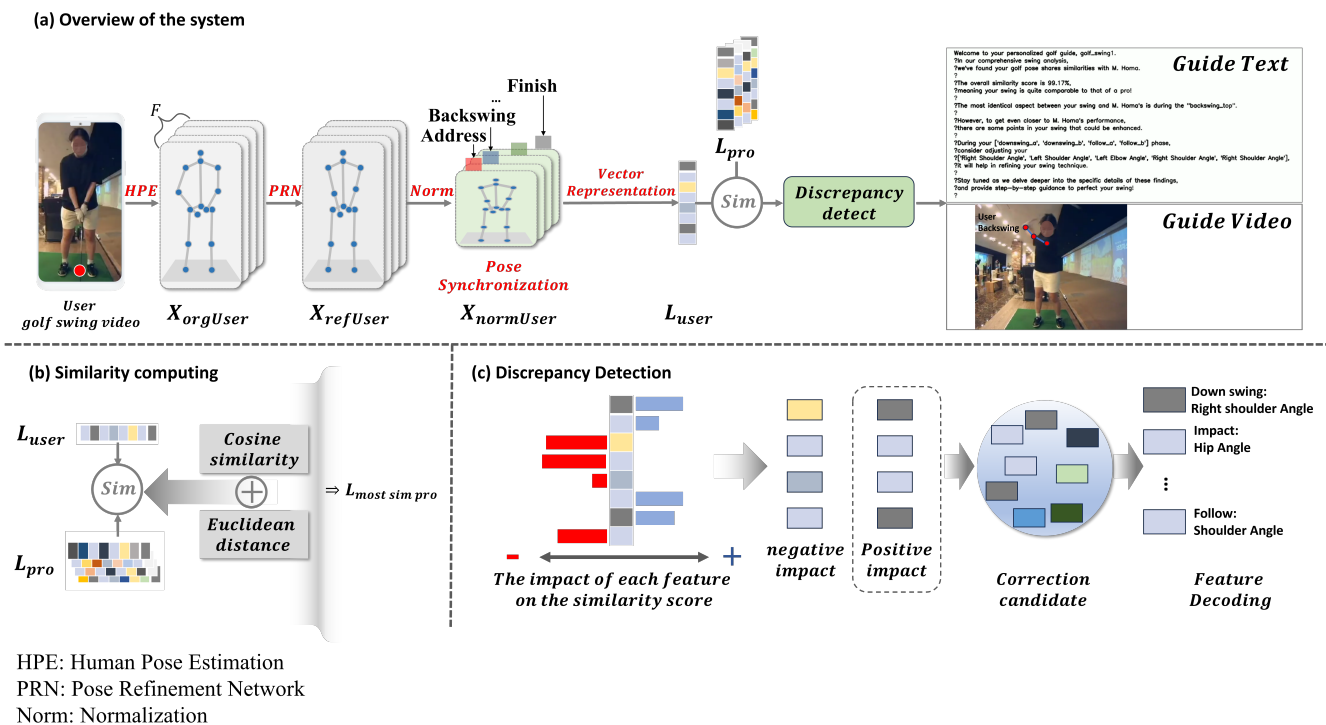


Figure 2. Overall process of proposed methods. (a) is the overview of the system that illustrates the input video to output result. $X_{orgUser}$ is a pose coordinate extracted from the HPE, and $X_{refUser}$ is a refined pose where $X_{normUser}$ is normalized and synchronized pose information for comparison to the professional golf swing. L_{user} is a representation vector including swing information, such as physical state, golf swing style, etc., and L_{pro} is a matrix that consists of swing representations of a pro golfer. (b) is an illustration of how to calculate the similarity between a professional golfer and a learner. (c) shows the process of calculating the importance of each joint feature to extract where the mismatch occurs when the most similar golfer is found. In (c) the impact of each feature, the red bars indicate that removing the feature had a negative impact on the overall similarity, while the blue bars indicate a positive impact.

3.1. Pose Refinement Process

In the pose estimation process for a golf swing video, there are some problems in detecting joint coordinates because of the fast motion. For instance, as shown in Figure 3,

when the swing movement is fast, coordinate estimation errors occur more frequently. These coordinate errors can generate inaccurate guidance during posture analysis and error calculation. To address this problem, we propose a method to effectively remove outliers with a rule-based outlier detection method and low computational cost by training a network using a single-layer Bi-LSTM (Bidirectional Long Short-Term Memory). The Bi-LSTM model for removing outliers is trained on data mimicking outlier coordinates that occur during the pose estimation process of a pose estimation model and outputs whether outliers occur in each joint's frame. We suggest an outlier detection algorithm that can be used in some body parts that cause several pose errors in golf swing motion by utilizing features of the golf swing posture. By removing the joint coordinates of the frames with outliers, we interpolate the missing coordinates through interpolation methods. Figure 4 shows the overall structure of the pose refinement process.



Figure 3. The cases of pose error in the pose estimation process. A white circle indicates that there is no problem with joint coordinate extracted by pose estimation model, and a red symbol means that there is a problem with joint coordinate. A yellow cross sign indicates where the problem occurs. The first image displays a correct or exemplary pose. The second image depicts an incorrect pose due to occlusion. The third image highlights a pose error in the right hand, resulting from image blurring due to rapid motion.

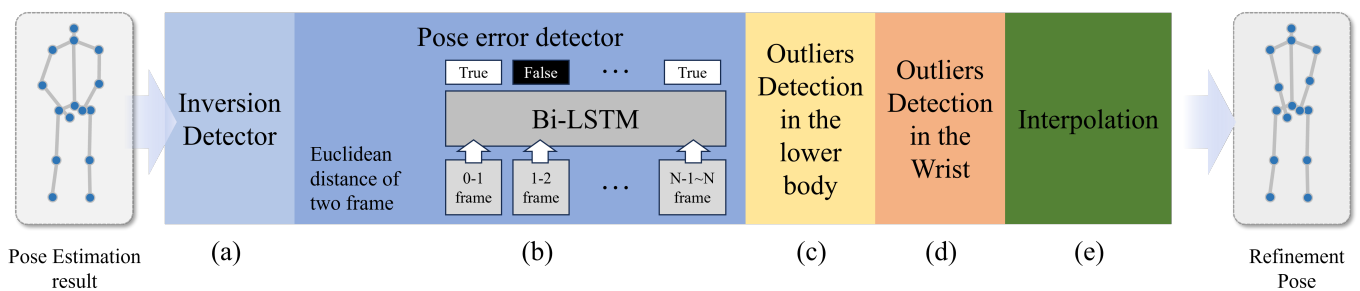


Figure 4. Pose refinement process. (a,c,d) are the algorithms for detecting outliers in a few specific joints using the feature of golf swing motion. (b) is the trainable model for detecting outliers in all joints. (e) is an algorithm for interpolation of the missing coordinates that are deleted by an outlier detecting model and algorithms.

3.1.1. Data Generation Technique for Learning Pose Errors

The proposed algorithm includes steps to remove outliers from estimated coordinates and interpolate missing coordinates. For outlier data collection, we used a data generation technique that mimics outliers based on data containing the correct joint coordinate labels. As illustrated in Figure 3, outliers in existing pose estimation models show a tendency for joint coordinates to be estimated to be a certain arbitrary distance, x , y , away from the actual correct coordinates. To mimic this, we applied amplification or attenuation to arbitrary x - and y -values from the actual correct joint coordinates to generate data mimicking actual outliers. The amplification and attenuation of the coordinates are applied to randomly selected frames so that the model can capture changes in consecutive frames.

3.1.2. Outlier Detection Model and Correction Algorithm

The outlier detection model uses a single-layer Bi-LSTM (Bidirectional Long Short-Term Memory) network and, when an outlier is detected, it is indicated in the output value of the corresponding frame. This model can be used for all joints of the body to detect outliers. Figure 5 shows the changes in coordinates by frame and visualizes the spike point on the graph. This graph confirmed that when outliers occur, the coordinate changes are abnormally large compared to previous frames. As seen in Figure 3, poses can be estimated at a certain level of accuracy in actions with slower speeds, but the proportion of inaccurate coordinate estimations increases as the action speed increases. The data input into the Bi-LSTM network is the coordinate change amount for each frame, calculated as joint coordinate distance changes between frames using the Euclidean distance method. Each piece of data contains a change amount for one joint. As the change amount value range differs depending on the video size, it is divided by the maximum value of each piece of data for normalization between 0 and 1. The network's maximum input length is also set, with padding values of 0 provided for shorter cases. The proposed algorithm includes steps to remove outliers from estimated coordinates and interpolate missing coordinates. The (b) network is a frame-by-frame outlier detection model using the Bi-LSTM model, which outputs the occurrence of outlier data in the coordinates estimated by the pose estimation model. For outlier data collection, we used a data generation technique that mimics outliers based on data containing the correct joint coordinate labels. Outliers in existing pose estimation models occur when the joint coordinates are estimated to be a certain arbitrary distance x , y away from the actual correct coordinates. To mimic this, we used the generated data in the previous step. The amplification and attenuation of the coordinates was applied to randomly selected frames so that the model could capture changes in consecutive frames.

3.1.3. Rule-Based Outlier Detection

The algorithms in Figure 4 are designed for rule-based outlier detection, specifically tailored to the unique features of a golf swing. The coordinate used in this paper is $(0, 0)$ at the top left, with the y -value increasing as you go down, and the x -value increasing as you go to the right. At impact, the ball flies towards the upper right. Algorithm (a), termed the 'Inversion Detector', identifies instances where the left and right pose coordinates are swapped. Throughout the entire swing motion, the x -coordinate of the right ankle should never exceed that of the left ankle. Based on this observation, the Inversion Detector flags any instance where the right ankle's x -coordinate surpasses the left ankle's x -coordinate. Similarly, algorithm (c) identifies outliers in the lower body by examining the distance between the two ankles. If this distance becomes less than half of the expected separation, the swing is flagged as having an anomaly. Lastly, algorithm (d) focuses on wrist coordinates. A golf swing video shows a motion of holding a golf club with both hands from the beginning to the end of the swing, raising it clockwise, and then swinging it counterclockwise towards the ball. During the golf swing, both hands are always holding the golf club. Since the distance between the two wrists remains constant throughout the swing, any deviation in this distance range is considered an error, and the swing is flagged

accordingly by this detector. The flags generated by algorithm (a) are addressed by directly converting the left and right joint coordinates, and the flags generated by algorithms (c) and (d) are deleted.

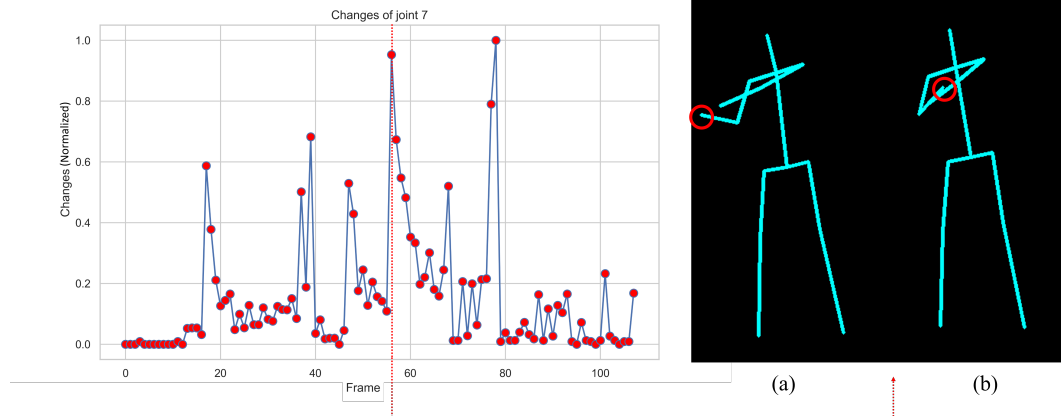


Figure 5. A graph depicting the coordinate changes in 7 (Right Wrist), with a highlighted visualization of the spike point on the graph. In the graph, each red dot records a frame-by-frame change in coordinates, and the red dashed line is a visualization of spike point. In the visualization, the problematic joints are circled in red. The x -axis is the frame, and the y -axis is the change in coordinates per frame, normalized to a value between 0 and 1. The adjacent frames, the previous frame (a), and the next frame (b) display the variations between two consecutive frames.

3.1.4. Interpolating Missing Coordinates

The algorithm for interpolating missing coordinates after outliers are removed through the outlier detection model is represented in Figure 4e. There are interpolation methods, such as Linear, Next, Previous, and Nearest, and, through performance comparisons of each method, it was found that the Linear method shows high performance. The Next and Previous methods fill the missing space by duplicating the values from the subsequent and preceding data points, respectively. The Nearest method fills the missing space by duplicating the value from the data point that has the closest x -value. However, these approaches simply duplicate the existing value, so these do not reflect the changes between two data points. Linear Interpolation is a method used to estimate the value of $f(x)$ for an x -value between two given points. This method allows for more accurate interpolation of values because it takes into account the movement path and changes over time, reflecting the variations in each frame.

3.2. Golf Swing Analysis Algorithm

This section discusses the process of comparing a professional's swing with a learner's swing to identify differences and create a guide based on these insights. To compare two different swing actions, both temporal and spatial alignment are required.

Firstly, spatial alignment involves applying pose normalization to standardize the coordinate sizes of both the expert's swing videos and the learner's swing videos, which have been shot at various resolutions. Secondly, temporal alignment entails dividing the swing actions according to the categorization of golf actions, assigning corresponding actions to each frame, and synchronizing the swings of the expert and learner, which may proceed at different speeds.

Next, to find the most similar expert, an embedding vector is constructed using information such as the angles of each joint, shoulder information, gender, and body proportions. By comparing the expert's embedding and the learner's embedding, the most similar expert is selected. Subsequently, feature importance analysis is conducted to identify the features needing correction, and these features are converted into human-understandable concepts, such as left elbow angle, shoulder angle, etc. Finally, the information requiring correction is

used to create guide text and videos through a template-based generation module. Detailed descriptions of each step are provided below.

3.2.1. Spatial and Temporal Alignment

In this section, we introduce an algorithm for spatially and temporally aligning coordinates to compare golf swing poses that were gathered in various environments. Spatial alignment refers to adjusting coordinates estimated at different resolutions to the same size and location. To do this process, we define the standard coordinate $s(x^s, y^s)$ and height h^s for normalizing the size and location of the pose. We first calculate the ratio r by dividing the standard height h^s by the target height h^t , which is calculated as the distance between the head and foot coordinates of the target pose. Then, we calculate the distance between the standard coordinates $s(x^s, y^s)$ and the resized root joint $t(x^{root}r, y^{root}r)$ of the target pose to derive the required coordinate shift distances $d^x = x^s - x^{root}r$ and $d^y = y^s - y^{root}r$ for each axis. Finally, we can acquire the normalized X of each target joint i by:

$$X_{SpatialNorm}^i = ((x^i r) + d^x, (y^i r) + d^y) \quad (1)$$

Temporal alignment involves the use of a posture segmentation algorithm to divide the swings of both learners and experts into stages based on the eight distinct actions in golf. The primary goal of this algorithm is to adjust temporal differences between two videos by recognizing each segment of the swing action through action segmentation. In this study, golf positions were categorized into Address, Takeback, Backswing, Backswing-top, Downswing, Impact, Follow, and Finish. We annotated the joint coordinates manually using standard swing posture images representing each posture and calculated the similarity of the user's entire swing and each action using Euclidean distances. However, due to the nature of golf swing actions, the Address, Takeback, and Backswing positions display similar motions to the Impact and Downswing positions. This similarity causes irregular posture segmentation and prevents the adjustment of the temporal differences between positions by reversing the temporal flow of the swing. In this study, we proposed an algorithm that divides golf swing positions into two segments based on the Backswing-top and Finish positions to distinguish each action. The proposed algorithm is detailed in Algorithm 1, and the flow chart is illustrated in Figure A1 in Appendix A.

Algorithm 1 analyzes changes in the y-coordinates of both wrists to detect the Backswing-top and Finish actions in a golf swing, where the wrist has the highest value. When the y-value increases from the preparatory action, the action is considered to have started, and the search for the highest height begins. The algorithm tracks changes, updates the highest y-value, and determines that the Backswing-top action has been found when the value no longer increases and begins to decrease. Afterward, the y-value is updated again until the Finish action appears. Through this process, the golf action is divided into two sections based on the Backswing-top, and the Euclidean distance measurement is used to determine the reference frames for the Address, Takeback, and Backswing actions in the first section and the remaining actions in the second section. In this way, the posture data containing the action segmentation markers for the temporal alignment of golf swing video frames can be obtained. Finally, we are able to obtain the normalized data's X_{Norm} through the spatial and temporal alignment algorithms.

Algorithm 1 Pose Division Algorithm

```

1: Input: Pose keypoints  $k$ 
2: Output: The two dividing point list  $div$ , The height of two top point  $h$ 
3: function POSEDIVISION( $k$ )
4:   Initialize:  $lm\_stack, stck\_distance \leftarrow 0$ 
5:    $h \leftarrow [\infty, \infty]; div \leftarrow [0, 0]; hist\_list \leftarrow [False, False]$ 
6:   for each frame,  $user\_joint$  in enumerate( $k$ ) do
7:      $wrist\_height \leftarrow user\_joint[7][1] + user\_joint[8][1]$ 
8:     if length( $lm\_stack$ ) > 0,  $stck\_distance \geq t$ , and  $hist\_list[1] == False$  then
9:        $hist\_list[0] \leftarrow True$ 
10:    end if
11:    if  $h[0] \neq \infty$  and  $stck\_distance \leq -t$  then
12:       $hist\_list[0] \leftarrow False$ 
13:       $hist\_list[1] \leftarrow True$ 
14:    end if
15:    if  $hist\_list[0] == True$  and  $h[0] > wrist\_height$  then
16:       $h[0] \leftarrow wrist\_height$ 
17:       $div[0] \leftarrow frame$ 
18:    end if
19:    if  $hist\_list[1]$  is True and  $h[1] > wrist\_height$  then
20:       $h[1] \leftarrow wrist\_height$ 
21:       $div[1] \leftarrow frame$ 
22:    end if
23:     $lm\_stack, stck\_distance \leftarrow limited\_queue(lm\_stack, wrist\_height)$ 
24:  end for
25:  return  $div, h$ 
26: end function

```

3.2.2. Swing Similarity Calculation

We detail a method for comparing the actions of professional golfers and learners using golf swing embedding to find the most similar pro golfer. To accurately represent the golf swing, we employ swing embedding, using the swing style and physical features of the golfer. The swing style is inspired by golf styles classified as hitter and swinger and is characterized by elements that can distinguish each style. We utilize four features of golfer and swing motion: gender, body proportion, the ratio of y-values of two top positions (Backswing-top, Finish), and the body angle at the impact position. To separate two genders in a large difference of similarity, we assign gender values of 0 and 10^3 in the embedding. The body proportions are calculated to divide all 15 edges of the joints by the height of the golfer. To acquire the relative speed of swing, we calculate the ratio of five positions to the total frame. Because the Impact and Finish positions are fixed motions, Takeback, Backswing, Backswing-top, Downswing, Impact, and Follow are used in the representation of the swing speed. Furthermore, we represent the swing style by calculating the ratio of the y-values of the two top positions and the right knee and right elbow angle at the impact. The generated 24-dimensional embedding is used to compare the swing actions of professional golfers and a learner by their cosine similarity scores and select the most similar pro golfer. This allows learners to refer to the swing of a pro golfer similar to their own swing and proceed with the necessary joint correction.

3.2.3. Golf Swing Pose Correction

After finding a similar golfer, the system identifies joints that need correction to improve the swing action of learners using explainable golf swing embedding. The posture correction algorithm measures core elements of the golf swing, such as shoulder and hip rotation angles, for each action and creates a joint angle embedding. The angle embedding is represented as a 10-dimensional vector, including the angles of both shoulders, hips, elbows, and knees, and the rotation angles of shoulders and hips. Based on this embedding

that represents the geometric information of the pose, a cosine similarity score with the most similar pro golfer is calculated for each swing position. To identify the specific points of body parts that show the discrepancy, the impact of each feature of embedding is observed by sequentially removing each embedding element. If the similarity increases when an element is removed, that element has a negative impact on similarity and needs improvement. Conversely, if the similarity decreases, that element has a positive impact on the similarity and needs to be maintained. We can easily convert each feature to a location of the body because the elements of the feature have the semantic information of body parts. By detecting and presenting specific angles that need improvement in the user's actions in this way, the algorithm supports self-training by users.

3.2.4. Generation of Swing Guide for Self-Training

We created template-based guide texts and videos to help users clearly analyze their swings. The text generation process begins with a brief greeting and presents a summary of the pro player most similar to the user, the similarity scores, the most similar posture, and the actions that need improvement. In addition, for actions that need correction, the guide video sequentially shows the footage of a pro golfer and the user for each segmented action and visualizes the skeleton of the joints that need correction, making it easy to understand at a glance. Figure 6 presents an example of the generated video. To simplify the differentiation of poses, the pose information is displayed at the top-left corner of the video. Additionally, the video highlights specific joints that require adjustments to align with the swing motion of a professional golfer.

Welcome to your personalized golf guide, user3.
 ?In our comprehensive swing analysis,
 ?we've found your golf pose shares similarities with T. Kim.
 ?
 ?The overall similarity score is 99.59%,
 ?meaning your swing is quite comparable to that of a pro!
 ?
 ?The most identical aspect between your swing and T. Kim's is during the "address".
 ?
 ?However, to get even closer to T. Kim's performance,
 ?there are some points in your swing that could be enhanced.
 ?
 ?During your ['address', 'takeback', 'backswing', 'backswing_top', 'downswing_a', 'downswing_b', 'impact', 'follow_a', 'follow_b', 'follow_c', 'follow_d', 'follow_e', 'follow_f', 'follow_g', 'follow_h', 'follow_i', 'follow_j', 'follow_k', 'follow_l', 'follow_m', 'follow_n', 'follow_o', 'follow_p', 'follow_q', 'follow_r', 'follow_s', 'follow_t', 'follow_u', 'follow_v', 'follow_w', 'follow_x', 'follow_y', 'follow_z'],
 ?consider adjusting your
 ?['Left Shoulder Angle', 'Right Elbow Angle', 'Right Shoulder Angle', 'Left Shoulder Angle', 'Right
 ?It will help in refining your swing technique.
 ?
 ?Stay tuned as we delve deeper into the specific details of these findings,
 ?and provide step-by-step guidance to perfect your swing!
 ?

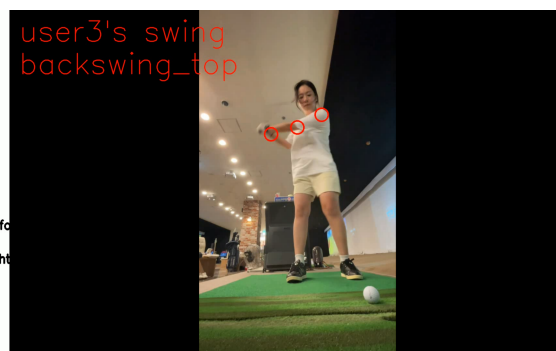


Figure 6. An example of a generated golf guide video. The image on the left is the generated text that will appear as the intro to your guide video. The text includes a brief greeting and provides a brief summary of the overall similarity score, the most similar joints, and the joints that need correction. The image on the right is a visual representation of the pro's movement and the learner's movement, showing where improvement is needed. The red circles in the image are automatically generated by the guide system to indicate areas that need to be corrected.

4. Prototype Implementation

We implemented the prototype of the proposed system utilizing the Python Qt5 library version 5.15.9. The implemented system includes simple login functionality, video upload, and guide video creation processes. Through this system, users can upload their golf swing videos and visually compare which parts differ from a professional's motion. This allows users to self-learn to achieve a swing more similar to that of a professional. Figure 7 illustrates the prototype where the proposed method is applied.

Users can create an account using basic information (account details, gender, etc.). Upon logging in with the created account, the main page for golf swing diagnosis is displayed. On the main page, users can diagnose their golf swing, view past records, and set options. Options allow users to decide whether they want to strictly view the swing differences between the pro and the learner, view them in a balanced manner, or view them conventionally. Additionally, settings related to the pose estimation model and coordinate interpolation can be adjusted.

After setting the options and pressing the swing diagnosis button, users are directed to the video upload page. Upon setting the file path for the user’s swing video, swing analysis is initiated. In this process, the proposed method identifies the swing of a professional golfer that is most similar to the user’s swing and demonstrates, through video and text generation, which parts need correction to achieve a more similar swing. The text generation offers a template-based presentation of an overall evaluation of the swing, which actions were most similar to those of the pro golfer, and which actions need improvement. The generated video aids the user by visually marking which joints differ, helping users easily understand the required corrections. The created guide videos are accessible through the history page, allowing users to easily access and review past records in the future.

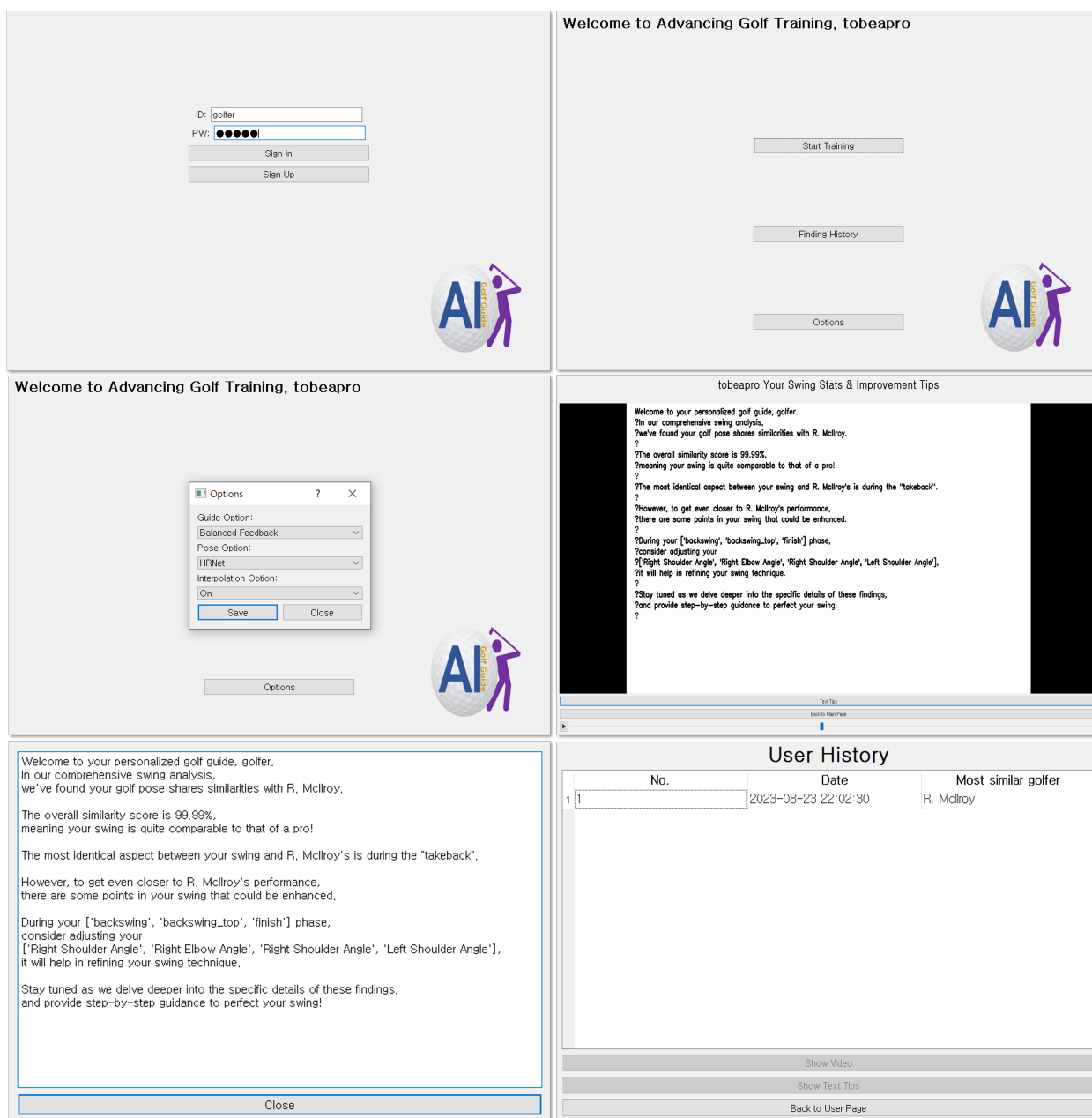


Figure 7. Prototype of proposed system. This figure begins with the top-left image as the first and concludes with the bottom-right image as the last. The first and second figures show the sign-in and main page view. The third figure shows the option of swing guide system. The fourth and fifth figures indicate the swing guide video and generated text. The last figure is the user history view that shows the user’s entire swing guide history.

5. Experiments

In this section, we present detailed evaluation results to assess the efficacy of our proposed approach. Firstly, we analyze the quantitative improvement in performance during the pose refinement process, accompanied by various visual aids supporting our findings. Moreover, we demonstrate that our suggested enhancement method is effective not only for 2D coordinate refinement but also for 3D coordinate improvement. Through a qualitative evaluation, we describe the intuitive characteristics of the enhanced pose. Finally, we provide examples of text and videos generated using the proposed self-training posture comparison algorithm, illustrating the effectiveness of our approach. The joint coordinates and their indices used in this paper are depicted in Figure 8.

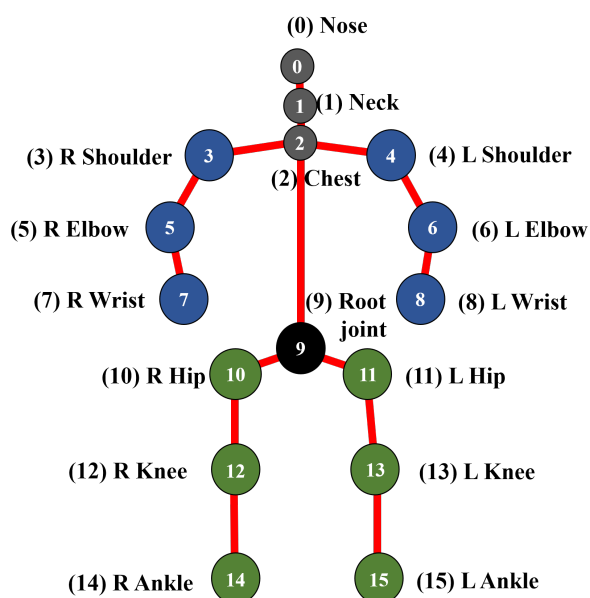


Figure 8. Joint index for swing analysis. The numbers in the figure represent the index of each joint, with grey representing the area near the face, purple the arms, green the legs, and black the root joint.

5.1. Dataset

To assess performance, we gathered golf swing motion data, manually annotating joint labels on golf swing videos. The collected data encompasses a total of ten distinct swing motions, of which five were used for training and the remaining five for evaluation. For measuring the enhancement performance of 3D coordinates, we utilized the publicly available 3DPW [26] dataset. Out of the data in the 3DPW dataset featuring individual subjects, ten instances were used for training, and four instances were deployed for testing. To build a database of professional golfers, we collected swing data from 16 players listed on the World Ranking provided by the PGA TOUR [27]. These data came without joint annotations. Therefore, joint labels were automatically generated using a pose estimation model.

5.2. Metric

To verify performance improvements, we employed the mAP (mean Average Precision) score as a metric. The mAP score is a commonly used evaluation metric in joint detection tasks, assessing the accuracy of the estimated joint coordinates. The AP score utilizes OKS (Object Keypoint Similarity), a normalized distance measurement criterion between the predicted and actual key points. In OKS measurements, threshold values exist; values closer to 1 entail a stricter evaluation, whereas values closer to 0 offer a more lenient assessment. We adopt a standardized approach, averaging the results tested at threshold values of 0.5, 0.05, and 0.95. Additionally, for evaluating 3D coordinates, we also measure the MPJPE score. With the mAP accuracy measurement in 3D coordinates,

there is a tendency for the accuracy to significantly drop as the number of prediction axes increases. To counter this, our performance measurement calculates the accuracy using the error between the estimated pose's root joint distance to the target joint and the ground truth pose's root joint distance to the target joint.

5.3. The Result of 2D Golf Pose Refinement

In this section, we evaluated our pose refinement method using 2D golf swing data. For our baseline model, we utilized BlazePose [9]. Through a total of five golf swing motion datasets, we assessed the effectiveness of our proposed method. Table 1 displays the evaluation results. The higher performance values are bolded. Our experimental results indicate that adding our proposed pose refinement module to the baseline model enhances performance. The notations (a), (c), and (d) in the table represent the rule-based outlier detection methods that utilize the characteristics of golf swings. When we removed the rule-based method and experimented, there were instances in which the performance improved. However, the overall performance declined. This shows that, while our proposed method effectively improves the actual performance, relying on a rule-based approach can result in performance degradation when the scenario deviates from the proposed conditions.

Table 1. Experimental result of 2D golf pose refinement module. This experiment compares the results of applying our pose refinement method to the baseline model, BlazePose. The evaluation score is based on mAP(%), which represents the average precision across different recall levels. (a), (b), and (c) are the algorithms in Figure 4. The “w/o” represents “without”.

| Model | Swing 2 | Swing 4 | Swing 5 | Swing 6 | Swing 7 | Mean |
|--------------------------------------|---------|---------|---------|---------|---------|-------|
| BlazePose | 67.11 | 59.49 | 57.24 | 66.66 | 40.09 | 58.12 |
| BlazePose + Our | 67.11 | 66.66 | 66.30 | 66.66 | 41.54 | 61.66 |
| BlazePose + Our w/o (a), (c), (d) | 67.11 | 58.64 | 66.66 | 66.66 | 41.54 | 60.12 |

5.4. The Result of 3D Pose Refinement

We evaluated the applicability of our proposed model in 3D coordinates using images from the 3DPW dataset where only a single person appears. Table 2 presents our experimental results. The data used for evaluation include “courtyard_bodyScannerMotions_00”, “courtyard_jumpBench_01”, “courtyard_relaxOnBench_00”, and “outdoors_freestyle_01”, represented as Pose 3, 6, 9, and 12, respectively. Due to the length constraints of the trained model, each dataset was truncated to 180 frames for evaluation. In the 3D coordinate improvement evaluation, we did not apply the rule-based outlier removal technique that is only applicable to golf motions. In the MPJPE scores, we observed performance improvements in most evaluation results when using our post-process coordinate refinement module. Notably, for Pose 9, we observed a roughly 9% reduction in the MPJPE score. Previous studies faced limitations in the refinement of 3D coordinates because of using an image process approach. On the other hand, our proposed method demonstrates that improvements in three-dimensional coordinates are achievable by leveraging coordinate sequence information.

Table 2. Experimental result of 3D pose refinement module. This experiment compares the results of applying our pose refinement method to the baseline model, BlazePose. The evaluation score is based on MPJPE, which measures the average Euclidean distance between predicted and ground truth 3D joint positions in human pose estimation tasks. (a), (b), and (c) are the algorithms in Figure 4. The “w/o” represents “without”.

| Model | Pose 3 | Pose 6 | Pose 9 | Pose 12 | Mean |
|-----------------------------------|--------|--------|--------|---------|--------|
| MPJPE | | | | | |
| BlazePose | 135.98 | 112.93 | 200.44 | 157.04 | 151.59 |
| BlazePose + Our w/o (a), (c), (d) | 136.97 | 112.59 | 182.79 | 150.61 | 145.74 |
| mAP | | | | | |
| BlazePose | 2.96 | 11.29 | 15.74 | 0.37 | 7.59 |
| BlazePose + Our w/o (a), (c), (d) | 3.51 | 10.92 | 16.11 | 0.18 | 7.68 |

5.5. The Impact of Interpolation Method

The interpolation methods compared in this experiment are Linear, Cubic, Nearest, Previous, and Next. We conducted the experiment based on the model with the best performance in Table 2. Table 3 shows the average interpolation performance comparison for each method with MPJPE score. The Cubic method uses a polynomial to interpolate values. It is sensitive to outliers, meaning that the inclusion of even a single outlier has the potential to significantly decrease the overall performance. In our performance experiments, the interpolation performance was lower than those of the other methods. The other methods performed similarly, but Previous and Next simply copied values, which is not the purpose of this study, and Nearest performed well, but not as well as Linear Interpolation. The Linear Interpolation method is stable in the sense that outliers do not bring down the overall performance, and it has the highest interpolation performance.

Table 3. The comparison result of the interpolation method in 3D pose. The method contains the experiments “w/o interpolation” (without interpolation) and the interpolation methods used in the comparison experiments. The evaluation was conducted with the MPJPE score, and the average score was used to measure the overall interpolation performance.

| Method | Mean of MPJPE |
|-------------------|---------------|
| w/o Interpolation | 151.59 |
| Linear | 145.74 |
| Cubic | 149.31 |
| Nearest | 146.61 |
| Previous | 146.53 |
| Next | 146.97 |

5.6. Impact of Coordinate Change Rate Adjustment

The pose refinement method proposed in this paper identifies abnormal changes, not typical joint trajectories, in the frame-by-frame coordinate change rate and considers them as outliers for removal. To understand the impact of this approach on performance improvement, we compared the coordinate change rate graphs before and after refinement. Figure 9 displays the change rate graphs of each joint before and after refinement. The red graph represents the change rate before refinement, and the blue graph shows the change rate after refinement. Each graph starts with joint 0 at the top and ends with joint 15 at the bottom. We compared the change rates of Pose 3, which showed an increase in the MPJPE score, and Pose 9, which demonstrated the most significant reduction in the MPJPE score. The proposed results reveal that Pose 3 still had many spikes in the change rate after refinement, indicating missed

detection in regions of abrupt change. In contrast, for Pose 9, the post-refinement change rate showed significant stabilization. Consistently, Pose 9 achieved more substantial pose accuracy improvement than Pose 3. Through these results, we confirmed that stabilizing the change rate can contribute to enhancing the accuracy of coordinates.

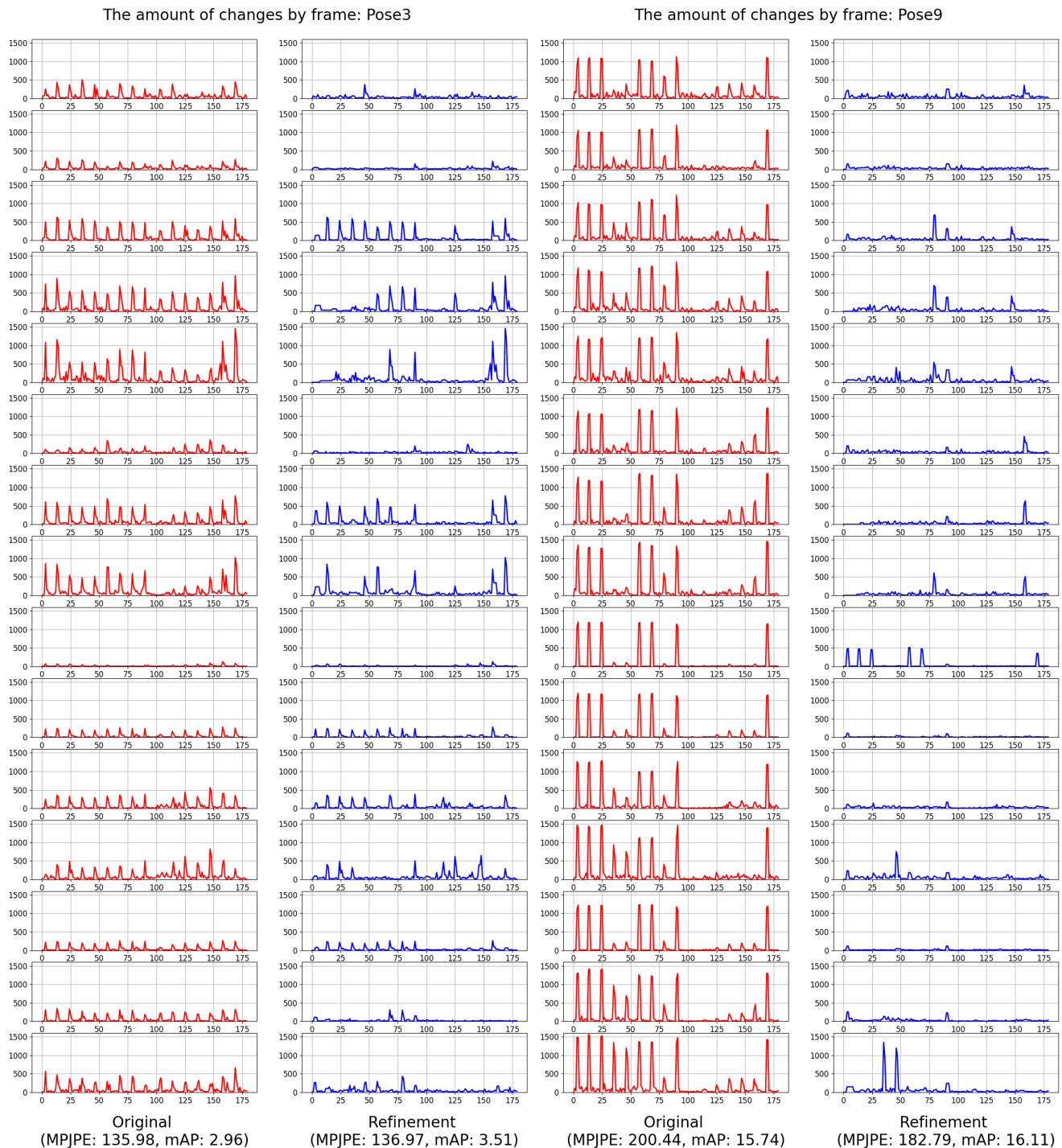


Figure 9. Graph of coordinate changes for each frame interval. The red graph represents the change rate of data estimated by the pose estimation model, while the blue graph shows the change rate after refinement. The x -axis is the frame, and the y -axis is the amount of change in coordinates. The top graph is at joint 0, and the bottom graph is at the last joint. The MPJPE score measures the average Euclidean distance between the predicted and ground truth 3D joint positions in human pose estimation tasks, and the mAP score represents the average precision across different recall levels.

5.7. Swing Pose Analysis Results

Figure 10 illustrates the results of comparing the similarity between the swings of a professional golfer and a user through a similarity comparison method (b) and measuring the influence of each feature on the pose with the lowest similarity (c) with its visualization shown in (a). From the results in (b), we can see that during the backswing phase, the user's swing motion has the least similarity to that of the professional. This similarity measure can help determine which movements should be corrected first and which ones are most similar. The influence of each joint feature during the backswing phase with the lowest similarity is presented in (c). By removing each feature and measuring its impact, if a feature has a negative value, it implies that it has a significant influence on pose similarity and can be interpreted as being similar to the professional's motion. Conversely, if a feature has a positive value, it suggests that removing this feature angle improves the similarity, indicating that corrections are necessary. The left shoulder angle with the highest positive value is visualized in (a) with a yellow circular marker, showing a noticeable difference from the actual motion of the professional. We can also observe positive values in the shoulder and hip angles, signifying a difference in the shoulder and hip angles compared to the professional. From these results, we can conclude that our Swing Embedding method is intuitive and effective in identifying real swing differences.

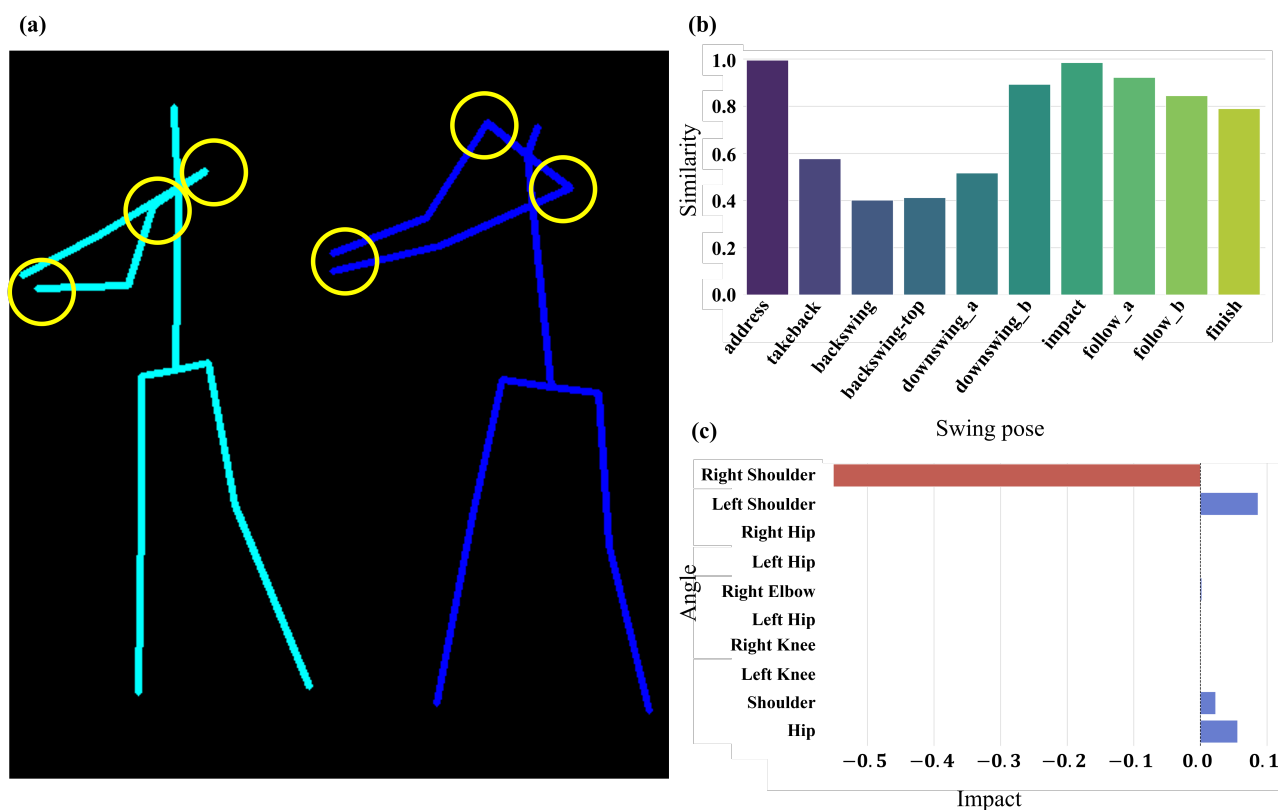


Figure 10. Comparison and visualization of similarity for each swing pose through swing motion embedding. (a) On the left is the user's swing, and on the right is the pro's swing. Visualization of the user and pro's action during Backswing. Yellow markers indicate the areas with the most differences. (b) Similarity calculation results between the pro and user for each swing action. (c) Impact analysis results of each embedding feature in Backswing, highlighting the least similar action. A red bar indicates that removing the feature had a negative impact, while a blue bar indicates a positive impact.

6. Limitations

In our study, we have proposed a method to enhance golf swing analysis. However, there are inherent limitations that need to be addressed. One primary concern is the distinct separation between our outlier detection and removal process and the interpolation process. This separation means that if one module does not function optimally, it could potentially compromise the efficacy of the entire system. For instance, accurate outlier detection, when followed by a sub-optimal interpolation, might lead to results that are not as reliable as when no interpolation is used. An integrated approach that combines both outlier removal and interpolation in an end-to-end framework may be more aligned with the desired objectives.

Additionally, our system's effectiveness is heavily reliant on the performance of the pose estimation model. If this model does not produce accurate results, it could significantly affect the quality of the guidance provided. Challenging conditions, such as low light or situations where the subject does not contrast well with the background, can decrease the precision of the joint coordinate detection. Such limitations can affect the user experience with the swing guide. Future work should focus on refining the system to capture swing motions effectively in diverse environments.

7. Conclusions

In this paper, we introduced pose refinement methodology and a golf swing analysis system based on explainable swing embedding for self-training. Our approach for pose refinement utilizes the changes in coordinates per frame for detecting biased pose joints. Because this method uses sequential coordinate information of the coordinates, we can apply it not only to 2D poses but also to 3D poses. Additionally, we demonstrated through these findings that we can refine the human pose estimation result by reducing the sharp changes in coordinates. Furthermore, we proposed a swing embedding method using the geometric information extracted from the swing pose. Our embedding method not only can compare the similarity of two golf swing poses but also can visualize the different points because the features of the embedding vector consist of intuitive information, such as the angle of the shoulder. Consequently, the case study showed that our swing guide system for self-training can appropriately suggest the specific body point that needs to be fixed to become more similar to the pro golfer's swing. Our proposed system can be utilized in an application service for a user who wants to study golf swing with a low-cost and time-efficient approach.

Author Contributions: Conceptualization, C.-Y.J.; methodology, C.-Y.J.; software, J.-H.K.; writing—original draft preparation, C.-Y.J.; writing—review and editing, D.-H.L.; supervision, D.-H.L.; funding acquisition, D.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by a grant from the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korea government (MSIT) (No.RS-2022-00155885, Artificial Intelligence Convergence Innovation Human Resources Development (Hanyang University ERICA)), and a grant from the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (No.NRF-2022R1F1A1073208).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HPE Human Pose Estimation
CNN Convolutional Neural Network

GPR Graph Pose Refinement
 PRM Pose Refine Machine, Pose Refinement Module
 GCL Graph Convolutional Layer
 TCL Temporal Convolutional Layer

Appendix A

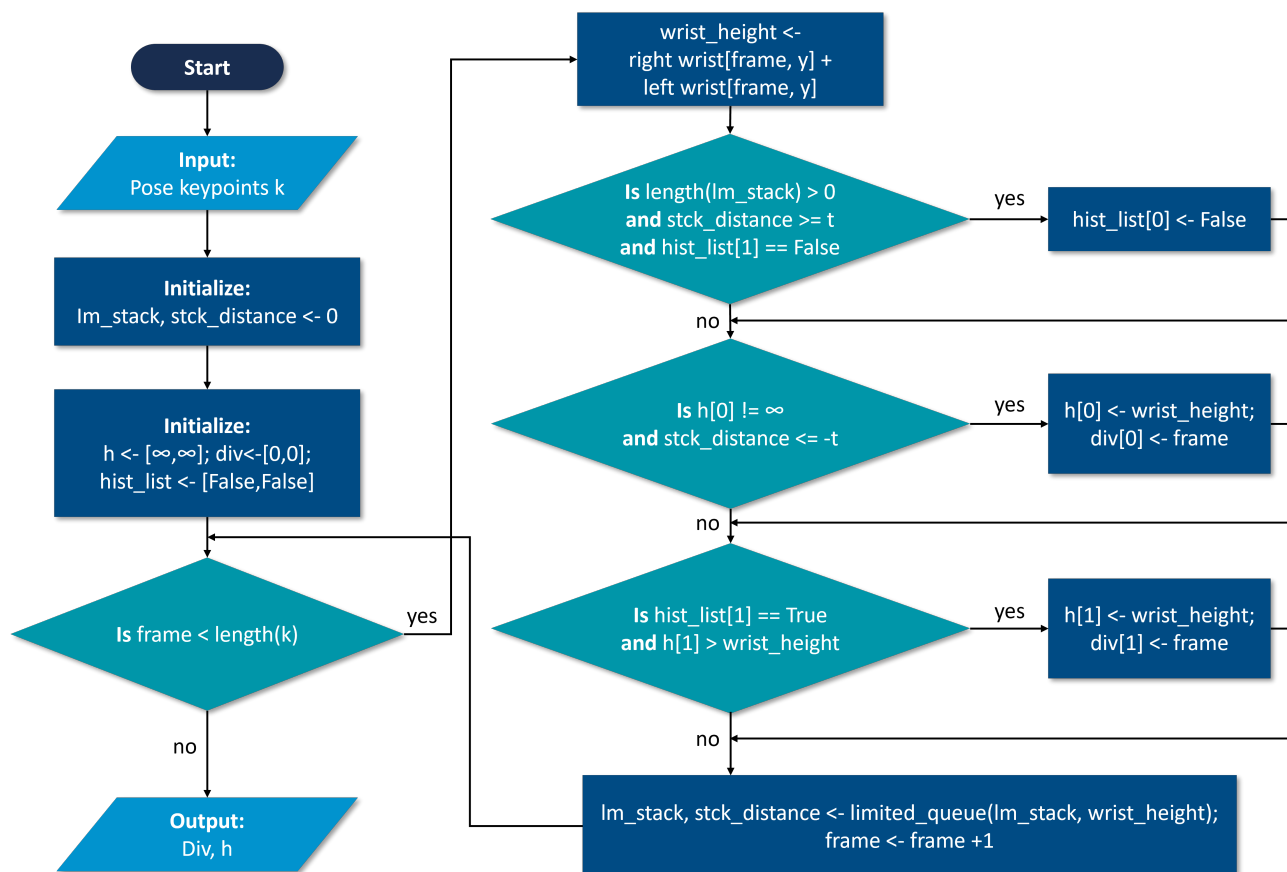


Figure A1. Flow chart of algorithm 1 for golf swing pose division.

References

1. Hoang, T.N.; Reinoso, M.; Vetere, F.; Tanin, E. Onebody: Remote posture guidance system using first person view in virtual environment. In Proceedings of the 9th Nordic Conference on Human-Computer Interaction, Gothenburg, Sweden, 23–27 October 2016; pp. 1–10. [\[CrossRef\]](#)
2. Liao, C.-C.; Hwang, D.-H.; Koike, H. How can i swing like pro? Golf swing analysis tool for self training. In Proceedings of the SA'21: SIGGRAPH Asia 2021, Tokyo, Japan, 14–17 December 2021; pp. 1–2. [\[CrossRef\]](#)
3. Han, P.-H.; Chen, Y.-S.; Zhong, Y.; Wang, H.-L.; Hung, Y.-P. My Tai-Chi coaches: An augmented-learning tool for practicing Tai-Chi Chuan. In Proceedings of the 8th Augmented Human International Conference, Silicon Valley, CA, USA, 16–18 March 2017; pp. 1–4. [\[CrossRef\]](#)
4. Kuramoto, I.; Nishimura, Y.; Yamamoto, K.; Shibuya, Y.; Tsujino, Y. Visualizing velocity and acceleration on augmented practice mirror self-learning support system of physical motion. In Proceedings of the 2013 Second IIAI International Conference on Advanced Applied Informatics, Los Alamitos, CA, USA, 31 August–4 September 2013; pp. 365–368. [\[CrossRef\]](#)
5. Kyan, M.; Sun, G.; Li, H.; Zhong, L.; Muneesawang, P.; Dong, N.; Elder, B.; Guan, L. An approach to ballet dance training through MS Kinect and visualization in a CAVE virtual reality environment. *ACM Trans. Intell. Syst. Technol.* **2015**, *6*, 1–37. [\[CrossRef\]](#)
6. Liao, C.-C.; Hwang, D.-H.; Koike, H. AI Golf: Golf Swing Analysis Tool for Self-Training. *IEEE Access* **2022**, *10*, 106286–106295. [\[CrossRef\]](#)
7. Liao, C.-C.; Hwang, D.-H.; Wu, E.; Koike, H. AI Coach: A Motor Skill Training System using Motion Discrepancy Detection. In Proceedings of the Augmented Humans International Conference, Glasgow, UK, 12–14 March 2023; pp. 179–189. [\[CrossRef\]](#)

8. Dang, Q.; Yin, J.; Wang, B.; Zheng, W. Deep learning based 2d human pose estimation: A survey. *Tsinghua Sci. Technol.* **2019**, *24*, 663–676. [[CrossRef](#)]
9. Bazarevsky, V.; Grishchenko, L.; Raveendran, K.; Zhu, T.; Zhang, F.; Grundmann, M. BlazePose: On-device real-time body pose tracking. *arXiv* **2020**, arXiv:2006.10204. [[CrossRef](#)]
10. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703. [[CrossRef](#)]
11. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 172–186. [[CrossRef](#)] [[PubMed](#)]
12. Li, Y.; Yang, D.; Chen, Y.; Peng, C.; Sun, Z.; Jiao, L. A lightweight top-down multi-person pose estimation method based on symmetric transformation and global matching. *IEEE Access* **2022**, *10*, 22112–22122. [[CrossRef](#)]
13. Xu, Y.; Zhang, J.; Zhang, Q.; Tao, D. Vitpose: Simple vision transformer baselines for human pose estimation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 38571–38584.
14. Moon, G.; Chang, J.Y.; Lee, K.M. Posefix: Model-agnostic general human pose refinement network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7773–7781. [[CrossRef](#)]
15. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the European Conference on Computer Vision 2016, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 483–499. [[CrossRef](#)]
16. Bulat, A.; Tzimiropoulos, G. Human pose estimation via convolutional part heatmap regression. In Proceedings of the European Conference on Computer Vision 2016, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 717–732. [[CrossRef](#)]
17. Wei, S.E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional pose machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4724–4732. [[CrossRef](#)]
18. Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; Sun, J. Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7103–7112. [[CrossRef](#)]
19. Carreira, J.; Agrawal, P.; Fragkiadaki, K.; Malik, J. Human pose estimation with iterative error feedback. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4733–4742. [[CrossRef](#)]
20. Cai, Y.; Wang, Z.; Luo, Z.; Yin, B.; Du, A.; Wang, H.; Zhang, X.; Zhou, X.; Zhou, E.; Sun, J.; et al. Learning delicate local representations for multi-person pose estimation. In Proceedings of the European Conference on Computer Vision 2020, Glasgow, UK, 23–28 August 2020; pp. 455–472. [[CrossRef](#)]
21. Wang, J.; Long, X.; Gao, Y.; Ding, E.; Wen, S. Graph-pcnn: Two stage human pose estimation with graph pose refinement. In Proceedings of the European Conference on Computer Vision 2020, Glasgow, UK, 23–28 August 2020; pp. 492–508. [[CrossRef](#)]
22. Fieraru, M.; Khoreva, A.; Pishchulin, L.; Schiele, B. Learning to refine human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 16 December 2018; pp. 205–214. [[CrossRef](#)]
23. Li, S.; Yi, J.; Farha, Y.A.; Gall, J. Pose refinement graph convolutional network for skeleton-based action recognition. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1028–1035. [[CrossRef](#)]
24. Bhagat, M.; Mandlekar, A.; Verma, R.; Lathia, T.; Tanna, S.; Saraf, A.; Bandukwala, S.; Patange, S.; Thakkar, P.B.; Singal, A. Video Call-based Fitness Assessment shows Poor Fitness in People with Type II Diabetes: Findings from Diabefly Digital Therapeutics Program. *J. Assoc. Physicians India* **2022**, *70*, 11–12. [[CrossRef](#)] [[PubMed](#)]
25. Kang, J.; Kang, C.; Yoon, J.; Ji, H.; Li, T.; Moon, H.; Ko, M.; Han, J. Dancing on the inside: A qualitative study on online dance learning with teacher-AI cooperation. *Educ. Inf. Technol.* **2023**, *28*, 12111–12141. [[CrossRef](#)] [[PubMed](#)]
26. Marcard, T.V.; Henschel, R.; Black, M.J.; Rosenhahn, B.; Pons-Moll, G. Recovering accurate 3D human pose in the wild using imus and a moving camera. In Proceedings of the European Conference on Computer Vision (ECCV) 2018, Munich, Germany, 8–14 September 2018; pp. 601–617. [[CrossRef](#)]
27. PGA TOUR. Official World Golf Ranking. Available online: <https://www.pgatour.com/tournaments/2023/tour-championship/R2023060/leaderboard> (accessed on 12 July 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.