Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 4, 2022, 549-557.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Sports movements modification based on 2D joint position using YOLO to 3D skeletal model adaptation

Anis Rahati and Kambiz Rahbar*

Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.

Article Info

Article History:
Received 06 June 2022
Revised 05 August 2022
Accepted 16 September 2022

DOI:10.22044/jadm.2022.11975.2344

Keywords:
Sports Movements Detection, Sports Movements Correction, 2D Joint Position Extraction, Joint Labeling, YOLO Neural Network, Sparse 3D Skeletal Model.

**Corresponding author: k_rahbar@azad.ac.ir (K. Rahbar).*

Abstract

Doing sports movements correctly is very important in ensuring body health. In this work, an attempt is made to achieve the movements correction through the usage of a different approach based on the 2D position of the joints from the image in 3D space. A person performing in front of the camera with landmarks on his/her joints is the subject of the input image. The coordinates of the joints are then measured in 2D space which is adapted to the extracted 2D skeletons from the reference skeletal sparse model modified movements. The accuracy and precision of this approach are accomplished on the standard Adidas dataset. Its efficiency is also studied under the influence of cumulative Gaussian and impulse noises. Meanwhile, the average error of the model in detecting the wrong exercise in the set of sports movements is reported to be 5.69 pixels.

1. Introduction

Doing sports exercises correctly in order to prevent injury and achieve the desired result is one of the most important factors in ensuring the health of the body, and therefore, is very important [1]. The lack of access to a coach, closure of gyms during the epidemic of diseases, and other factors affect public sports. Machine vision technologies are used to deal with such problems in the present work. Therefore, it is required to train the visual system to monitor the athlete and provide the necessary corrections in performing his/her sports movements. In this regard, extracting the three-dimensional position of the athlete's body and the coordinates of his joints is considered as a challenge in this area. In the following, first, the four groups of the body position extraction approaches are reviewed, and then the proposed solution is presented. These four groups can be organized as: 1) methods based on supervised learning, 2) methods based on unsupervised learning, 3) methods based on semi-supervised learning, and 4) methods based on sparse modeling; it should be mentioned that the proposed solution falls into the fourth group.

The first group emphasizes learning based on monitoring the supply of labeled datasets to create a mapping between the received image and the desired coordinates. As examples of the supervised learning, Lee and Chan [2] have proposed a common multifunctional deep learning framework for estimating three-dimensional positioning and identifying two-dimensional body organs. Tekin *et al*. [3] have provided a structured deep learning framework that includes a pre-trained automated encoder for reconstructing human three-dimensional situations. However, in addition to the need for educational data, the segregation and analysis of residual learning error is also one of the challenges in this area [4]. In other words, the segmentation is more related to the separation of image components by the network convolution section or to the error of the network mapping section after the convolution section

The second group, unsupervised learning, does not require the provision of labeled datasets. However, there are challenges in extracting the desirable features and segregating them. In this

regard, the researchers have tried not to pay the cost of labeled data by overcoming this challenge.
As an example of unsupervised learning, Kodo *et al*. [5] have proposed learning three-dimensional positioning without the use of three-dimensional datasets. In this method, using generative adversarial networks, a three-dimensional image is projected from two-dimensional common joints in an image. In another approach, Chen *et al*. [6] have used an unsupervised domain adapter network for this purpose. Tripati *et al*. [7] have also extracted two-dimensional joints as the input and three-dimensional skeleton as the output using the Pose Net unsupervised neural network framework.
In the third group, semi-supervised learning is considered as an intermediate approach. For example, [8] introduces a semi-supervised regulatory framework for estimating the human condition. Unlabeled data is used to compensate for the complexities of the input space, and is modeled by the nearest neighbor. It has also been shown in [9] that 3D video situations can be effectively estimated with a fully torsional model based on discrete time complexities, and a simple and effective semi-supervised training method is presented that uses unlabeled video data, and in case of the shortage of data, it has used labeled data. First, they started with the 2D key points projected for the unlabeled video, and then they estimated the 3D positions, and finally, they went back to the 2D input key points. It should be noted that estimating the human three-dimensional state of an integrated image requires large amounts of tagged two-dimensional and controlled three-dimensional data contained in the tagged data set, which are costly. In [10], to reduce this dependency, a contradictory semi-supervised learning multiplex framework is presented that uses the similarity of unlabeled and uncontrolled mode information. In this method, synchronous multiplayer videos of human movements are used as an additional signal of poor monitoring to guide the regression of the human 3D state. This framework uses hard negative mining based on temporal relationships in multiplayer videos to achieve a fixed embedded multipurpose display.
The fourth group has a different approach. In this approach, the main goal is to provide a sparse display model by combining basic models to approximate a given system. Therefore, in estimating a three-dimensional position, a linear combination of several two-dimensional base positions can be used. In [11], to retrieve the exact three-dimensional position of a set of two-dimensional joints, the sparse display model has been used as one of the effective ways. In the sparse display model, a three-dimensional position is presented as a linear combination of several main base positions. The advantage of approaches based on the sparse display model is that it does not require paired training data (2D, 3D). Although the solitude-based approach is appealing because of its simplicity, it often encounters estimation errors. This error is related to the estimated three-dimensional position and the expected three-dimensional position, which is not directly measurable in the sparse display model [12-15]. As another example in [13], the three-dimensional position is taught from the training dataset. This model is then reinforced by showing sparse by adding greedy principles to the model. Along the way, Wang *et al*. [12] have used $L1$ norm to measure the difference between the two-dimensional input and the projected joints. Zhou *et al*. [14] have optimized the model [13]. In [16], an image sequence is used to three-dimensionally estimate the complete position of the human body. This approach uses a deep fully convolutional network to predict the location uncertainty of two-dimensional joints. The three-dimensional state extraction is then accomplished through an expectation-maximization algorithm throughout the sequence. Fan *et al*. [17] have divided the whole-body gesture training space into sub-spaces with smaller dimensions, and then used the dictionary to teach the block structure based on this sub-space.
In this work, an attempt has been made to introduce a different approach in achieving the goal based on the two-dimensional position of the joints from the image in three-dimensional space. In this regard, first, using the Yolo neural network, the two-dimensional position of the joints in the image is determined and extracted. It should be noted that the athlete uses a set of simple wearable landmarks Figure 1 on his clothing to help extract the position of the joints. The Yolo network extracts the position of the joints by extracting the position of these landmarks in the two-dimensional image. The skeletal model is then calculated based on the position of the joints in two-dimensional space. The proposed solution uses a set of 3D sparse models based on simple 3D basic models. The sparse model includes correct skeletal positions. This three-dimensional model is mapped in two directions on two-dimensional space. Then, according to the skeletal model extracted from the Yolo network, the best and closest mapped models are selected. The difference between the two models shows the athlete's movement error.

This error can be reported in two-dimensional space and/or with the help of a three-dimensional sparse model transferred to three-dimensional space and used to correct sports movement. In the second part, the extraction of the position of the joints in two-dimensional space is examined in detail. The third section discusses the three-dimensional sparse model and skeletal reconstruction. Then, in the fourth section, the proposed solution is verified. The fifth section is devoted to summarizing and future suggestions.

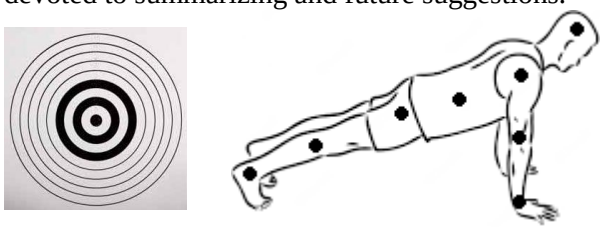


Figure 1. Wearable landmark (left), landmark on human body (right).

2.Extracting Position of Joints in Two-dimensional Space

In this work, the use of Yolo neural networks is suggested for the stable extraction of the position of body parts in the image. Yolo, as an example of area-based torsional networks, can be very effective in this regard. Figure 2 shows the Yolo network approach. As shown in the figure, the four steps in this approach are clear. First, a set of proposals is selected in the image (for example, 2000 proposals). Then, each proposal is resized to a predefined size to be used by torsional neural networks for feature extraction. Finally, a separator is used for separation and labeling. It should be noted that in the Yolo network, object recognition is seen as a regression problem that extends directly from image pixels to box coordinates and class probabilities. There is only one torsional grid that receives the image by resizing the input and then simultaneously predicts several boxes with the probability of classes.

2.1.Yolo architecture

In the Yolo architecture, the input image with dimensions of 448 × 448 × 3 is divided into an S × S × 1 network and sent to a torsional grid. The output of the torsional grid will be a matrix measuring S × S× 30. Each of the S × S matrix elements has an output equivalent to one cell in the S × S network. The S × S × 30 output contains the coordinates of the boxes and the probabilities. If we are in the training process, the output of S × S × 30 along with the actual boxes or the ground truth is given to the loss function. The value of S in the first version of Yolo is equal to 7. If we are in the testing process, the output of S × S × 30 is given to the non-maximum suppression algorithm so that the weak boxes are eliminated and only the correct boxes are displayed in the output. Yolo includes a torsional neural network with twenty-four torsional layers for feature extraction as well as two fully-connected layers to predict the coordinates and probability of objects. The Yolo network architecture is shown in Figure 3.

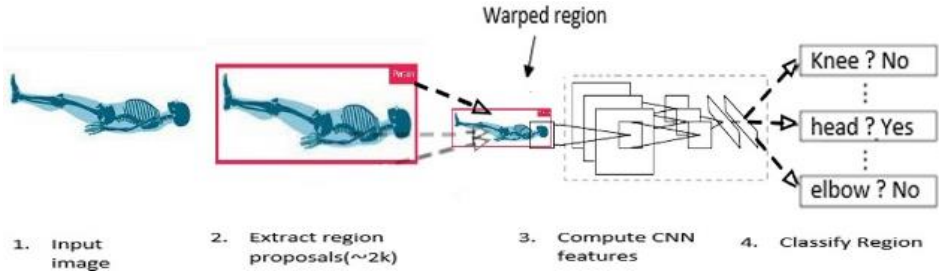


Figure 2. Yolo network approach.

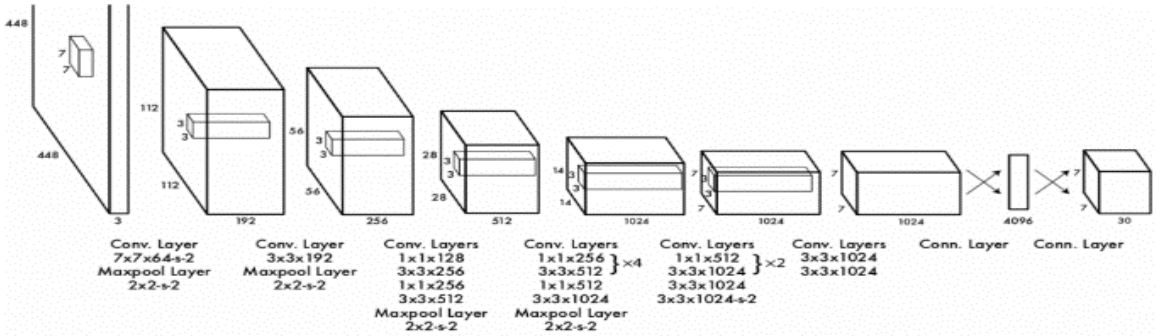


Figure 3. Yolo network architecture with twenty-four torsional layers.

In the fast version of Yolo, a neural network with fewer torsional layers is used. In this version, instead of 24 torsional layers (main yolo), 9 torsional layers are used, and of course, the number of filters per layer in fast yolo is less than the main yolo. The input size of both networks is 448 × 448 × 3, and the output of the network is a 7 × 7 × 30 of the projection's tensor. The Leaky ReLU activation function is used in all layers. Also, the output size of the network is 7 × 7 × 30. In this structure, the input images are divided into a 7 × 7 network. Therefore, the output of 7 × 7 corresponds to the gridded input image. Each entry of matrix at 7 × 7 output corresponds to a cell in the gridded input image. In addition, each output of this 7 × 7 output matrix has a vector of length 30. This vector contains probability prediction information and box coordinates. In this way, each cell of this 7 × 7 matrix can draw two boxes. 5 parameters (x, y, w, h, confidence) are needed to draw each box. The x and y parameters show the coordinates of the row and column of the source box (center of the box). The coordinates w and h correspond to the width and height of the box, respectively. With these four parameters, the box can be drawn. The fifth parameter is the confidence multiplier. It should be noted that Yolo uses a modified version of sum of the squared error function. Equation (1) presents the mentioned loss function.

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\begin{bmatrix}(x_i-\hat{x}_i)^2+\\(y_i-\hat{y}_i)^2\end{bmatrix}+$$
$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\begin{bmatrix}(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+\\(\sqrt{h_i}-\sqrt{\hat{h}_i})^2\end{bmatrix}+$$
$$\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}(c_i-\hat{c}_i)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj} \quad (1)$$
$$(c_i-\hat{c}_i)^2\sum_{i=0}^{S^2}\mathbb{1}_{i}^{obj}\sum_{c\in classes}(p_i(c)-\hat{p}_i(c))^2.$$

In this regard, using the relation of the sum of squares error, the positions of the origins of the two projected and real boxes (x, y) are compared. The indices i and j represent the cells (49 cells) and boxes (B), respectively. $\mathbb{1}_{i}^{obj}$ denotes if object is present in cell i. $\mathbb{1}_{ij}^{obj}$ denotes the j[th] bounding box responsible for prediction of object in the cell i. Dual Sigma is responsible for examining individual cells and boxes. $\boldsymbol{\lambda_{coord}}$ and $\boldsymbol{\lambda_{noobj}}$ are the regularization parameters required to balance the loss function. In the second part of the formula, an almost similar relationship is seen with the first part. However, instead of x and y, w and h are used. The purpose here is to compare the width and height of the projected box with the actual box. In the image, objects of different sizes from very small to very large can appear. When the boxes of these objects are compared to the real ones, all the boxes of any size will be compared by one criterion. It should be mentioned that the error in large boxes is not the same as the error in small boxes, thus one pixel of error in a large box should be less penalty than one pixel of error in a small box. The rationale for this relationship is to penalize large boxes less than small boxes. The third and fourth parts of the formula provide the reliability for the presence or absence of an object in the box. The third part is for the confidence multiplier of the boxes that contain the object, and the fourth part is for the boxes that do not contain any object. Behind the sigmas of the fourth section is a hyper parameter λ. The value of this parameter is considered 0.5; because in every image many boxes do not contain an object and the number of boxes without an object outnumbers the boxes with an object. Therefore, in order for the amount of loss of objectless boxes not to prevail over the boxes with object, a coefficient of 0.5 is placed behind it to reduce the amount of waste of objectless boxes.

3.Three-dimensional Sparse Model and Skeleton Reconstruction

In this process, the skeleton is given with N joints, and the three-dimensional position of the skeleton is represented by a set of three-dimensional joints, namely $Y = \{j_i\}_{i=1}^{N} \in \mathbb{R}^{3\times N}$. j_i is the three-dimensional coordinate of the i-th joint. The corresponding two-dimensional position (2) is shown in the projected image as follows [22]:

$$x = (S\begin{pmatrix}1&0&0\\0&1&0\end{pmatrix}R)Y + T. \quad (2)$$

where $\mathrm{X} = \{\mathrm{j}'_{\mathrm{i}}\}_{\mathrm{i}=1}^{\mathrm{N}} \in \mathbb{R}^{2\times \mathrm{N}}$ is two-dimensional joints projected from three-dimensional joints. In (2), $\mathrm{R} \in \mathbb{R}^{3\times 3}$, S and T are the rotation, scaling, and transformation matrices, respectively. R and T are together the calibration parameters of the external camera. When we have $\mathrm{S} = \mathrm{diag}(\mathrm{s_x}, \mathrm{s_y})$, then the scaling is set along the x and y directions, which reflect the inherent camera calibration parameters. In addition, the T transfer matrix is usually omitted because the data is already centralized. As a result, the above equation is simplified as follows:

$$X = (SR^{*})Y \quad (3)$$

where $R^* = I_{2\times3}$. Equation (3) is nothing more than a linear projection from a three-dimensional to a two-dimensional state, and is formulated with a projected model by a weak-perspective camera. Such an approximation is reasonable when the depth of the object along the path of view (as opposed to the distance from the camera) is small. With the sparse representation model [13], the three-dimensional position can be approximated as a linear combination from k to the value of the three-dimensional base.

$$Y = \sum_{i=1}^{k} C_j B_j \tag{4}$$

where $B_j \in \mathbb{R}^{3\times N}$ is a ground state, and c_j is the corresponding multiplier. The basic state is already trained by the sparse learning algorithms [18-20]. By placing (4) in (3), we conclude:

$$X = SR^* B^* c \tag{5}$$

where $B^* = \{B_j\}_{j=1}^{k}$ and $c = [c_1, \dots c_k]^T$. It is also obvious that $Y = B^* c$. In this case, both the two-dimensional modes of the X matrix and the calibration parameters of the inherent camera (i.e. S) are known. The problem of estimating the matrix Y has become the problem of solving the parameters of the external camera R^* and the coefficient vector $c = [c_1, \dots c_k]^T$ from the sparse display model. In addition, c is expected to have a fraction of non-zero inputs only according to the SR model. Therefore, the problem is formulated as an optimization problem:

$$\min_{R^*,c} \|c\|_0 \quad \text{s.t.} \quad X=(SR^*)(B^*c) \tag{6}$$

Since (6) directly solves the NP problem, the objective function is also relaxed as follows:

$$\min_{R^*,c} \|c\|_1 \quad s.t. \quad X = (SR^*)(B^*c) \tag{7}$$

The problem of minimization in (7) is usually solved in two different ways. In [13], standard anthropometrics is performed in (7) and solved by the matching tracking algorithm. Zhou *et al*. [14], on the other hand, modify it as a convex shape. Therefore, an efficient algorithm is proposed to solve the convex optimization problem. However, as mentioned earlier, the errors due to the norm approximation $\ell 1$ and the three-dimensional to two-dimensional projection have not been carefully examined in any of them. To reduce the two types of estimation errors $\ell 1$ due to the norm approximation and three-dimensional to two-dimensional predictions in the solitude-based estimation model, first consider the sparse representation by re-weighting.

$$\min_{R^*,c} \|w_c\|_1 \quad s.t. \quad X = (SR^*)(B^*c) \tag{8}$$

In this regard, the W matrix starts with the identity matrix, and is updates in each iteration. The intermediate solutions S, R*, and B are used to place in W. The resulting W is used in turn to solve S, R*, and B in the next iteration. Unlike the weighting scheme in [15], [21], w_j has been comparatively updated using the original 3D position. The weight upgrade steps are as follows:

Step 1: Select the basis of the positions from the comprehensive dictionary with indicators of non-zero inputs in c.

Step 2: Place the weight w_j based on the similarity between the two-dimensional position of the input image and the main position selected by (9).

$$w_j = \begin{cases} e^{\frac{-\left\|X-(SR^*B_{j,\,j\in I_C})\right\|_2}{2\sigma^2}} & if \\ 0 & otherwise \end{cases} j \in I_C \tag{9}$$

where σ is the width of the Gaussian nucleus. I_c represents a set of indices with non-zero inputs in c.

In practice, the constraint in (8) makes the presence of noise challenging, and the difference between X and the projected estimate (SR*) (B* c) is expected to be as small as possible. As a result, the Lagrangian multiplier (in (10) has been considered to solve (8) as follows:

$$\min_{R^*,c} \frac{1}{2}\left\|X-(SR^*)(B^*c)\right\|_F^2 + \alpha\|w_c\|_1 . \tag{10}$$

Here, $\alpha > 0$ is the parameter that strikes a balance between error and regularity. (10) is the loss function of this method in which $\|.\|_F$ is the Frobenius norm. The variables R*, C, and W are alternately optimized by updating one of them (while the others are fixed). When c reaches the desired level, the human three-dimensional position $\hat{Y}$ can be reconstructed using the original three-dimensional position model in (4).

Equation (10) allows a slight difference between a given two-dimensional position and an estimated two-dimensional position (estimated from a projected three-dimensional position) in terms of fault tolerance. This difference is called the residual source ε_s, and is as follows:

$$\varepsilon_s = X - SR^*\hat{Y}. \tag{11}$$

in which, $\hat{Y}$ is estimated the three-dimensional position.

It is assumed that the correct basis of the three-dimensional position, Y_g, and the input of the two-imensional position, X is also a direct two-dimensional plot of Y_g, i.e. $X = SR^* Y_g$. Thus (11) is written as follows:

$$\varepsilon_s = SR^* Y_g - SR^* \hat{Y} \tag{12}$$

The organized form of (12) is as follows:

$$\varepsilon_s = SR^* (Y_g - \hat{Y}) \tag{13}$$

The term $Y_g - \hat{Y}$ is nothing more than the discrepancy between estimated 3D and the true 3D pose. This discrepancy is given as residual target ε_t in our paper. As a result, the relation between source residual ε_s and target residual $\boldsymbol{\varepsilon_t}$ is presented as follows:

$$\varepsilon_t = (SR^*)^{-1} \varepsilon_s = R^{*\dagger} S^{-1} \varepsilon_s . \tag{14}$$

where $R*\dagger$ is the pseudo-inverse of the rotation matrix $R*$. To this end, the source residual that is obtained by our minimization procedure can be related to the residual target. From this viewpoint, it is very important to know the discrepancy between the estimated 3D and the true 3D pose.

To estimate the three-dimensional position, it is necessary to estimate the difference between $\hat{Y}$ and Y_g. With the help of ε_t, $\hat{Y}$ of the sparse display model is adapted. As a result, the estimated final three-dimensional position is expressed as follows:

$$\hat{Y}_{final} = \hat{Y} + \varepsilon_t = \hat{Y} + R^{*\dagger} S^{-1} \varepsilon_s \tag{15}$$

At the end of the optimization, the minimum error, ε_s, is obtained in the projected two-dimensional range. According to (15), this residue can be used to estimate the difference in three-dimensional estimates from the correct basis. In addition, to obtain a more accurate position $\hat{Y}$, the statistical range of the mean squares of the error is extended by imposing the residual source ε_s in (10). This is to minimize the effects of random disturbances during the optimization process and to smooth error reduction. In this way, the additive error of two consecutive iterations is considered, i.e. (10) is rewritten as a stronger loss function $\mathcal{L}_2(R^*, c; X)$.

$$\begin{aligned} &\min_{R^*,c} \frac{1}{2} \left\| X - (SR^*)(B^* c) + \beta \Re_S \right\|_F^2 \\ &+\alpha \|W_C\|_1 = \min_{R^*,c} \frac{1}{2} \|X - (SR^{*l})(B^* c^l) \\ &+\beta X - \beta (SR^{*l-1}) \Big\|_F^2 = \min_{R^*,c} \frac{1}{2} \|(1+\beta) X \\ &- \left[(SR^{*l})(B^* c^l) + \beta (SR^{*l-1})(B^* c^{l-1}) \right] \Big\|_F^2 \\ &+\alpha \|W_C\|_1 \alpha \end{aligned} \tag{16}$$

where β is the equilibrium coefficient, and l also represents the current iteration. In the optimization iteration, the remaining expression is computed by the previous iteration solution. "Equation (16)" minimizes two-dimensional position input X and $\frac{(SR^{*l})(B^* c^l) + \beta (SR^{*l-1})(B^* c^{l-1})}{1+\beta}$.

4. Verification and Laboratory Results

In order to extract the three-dimensional state of the human body and also to evaluate the amount of error, the Adidas database [23] including correct and incorrect sports movements has been used Figure 4. Figure 5 describes the testing process. As shown in Figure (5-a), the input image is an image of an athlete whose body joints are labeled. Figure (5-b) shows the adaptation of the skeleton on the athlete's body. The test scenario is in the way that the person performs sports movements in front of a camera. The camera is placed on a tripod at a distance of about one to two meters from the person and monitors the person. Then, based on the proposed solution on the athlete's body, the coordinates of the joints are measured in two-dimensional space. The two-dimensional skeleton is matched with the two-dimensional skeletons extracted from the reference model and stored in the form of a sparse matrix. The comparison of the nearest two-dimensional model extracted from the reference with the athlete's movement model is evaluated to measure the athlete's movement error (Figure (5-c)).

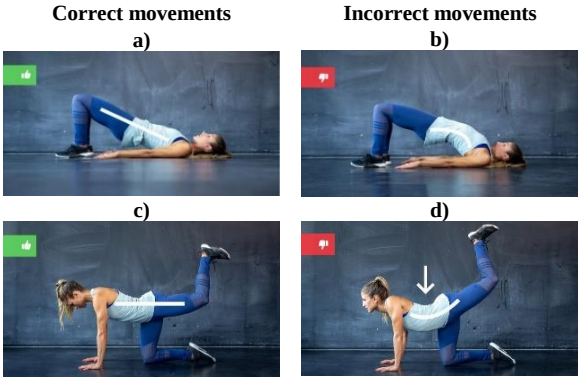


Figure 4. An example of an Adidas database image, (a and c) correct gestures, (b and d) incorrect gestures.

For example, Figures 6 show the correct and incorrect cases of movement of plank exercise, which includes three rows of input image, matching skeleton on the body, and extraction skeleton.

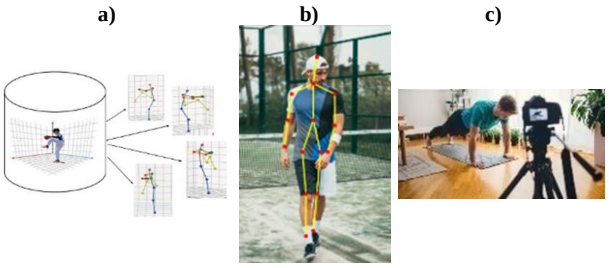


Figure 5. Testing process: a) a person under monitoring of camera performs sports movements, b) Extraction of the skeleton from the athlete's movement, c) Extraction of two-dimensional model from reference three-dimensional model at different angles.

In the correct movement, the elbow is placed directly on the ground and parallel to the shoulder to form a 90-degree angle with the ground, the back of the body should be perfectly flat, and the spine should be in a neutral position. The person's hips should not be bent. In the incorrect position 1, the hips are upwards, and in the incorrect position 2, the hips are downwards. These deviations cause Planck to move incorrectly.

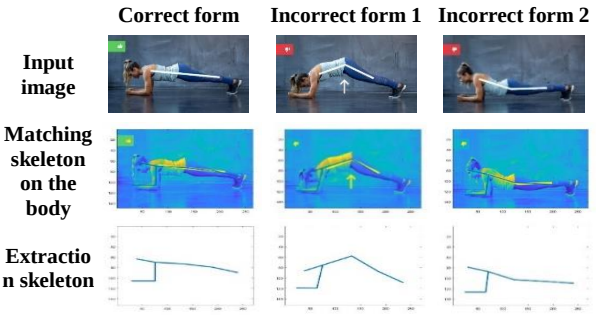


Figure 6. Planck`s correct and incorrect movement.

Table 1. comparing values of correct and incorrect Plank movement's coordinates in two-dimensional space.

No	Organ name	Correct movement coordinates	Incorrect movement coordinates	Difference
1	Head	(157,253)	(154,364)	(3,93)
2	shoulder	(305,279)	(301,301)	(4,2)
3	Elbow	(304,423)	(255,478)	(49,55)
4	Wrist	(95,423)	(95,477)	(0.54)
5	Hip	(533,292)	(531,231)	(2,61)
6	Knee	(753,307)	(738,348)	(15,41)
7	Ankle	(959,358)	(939,436)	(20,78)

Table 1 compares the Planck's correct and incorrect movement coordinate values with each other. Tables 2 and 3 do the same comparison for the time that the input image is affected by retractable Gaussian noise and impulse noise. Studying Tables 2 and 3 show that increasing the noise may cause some data to be missing, also under the influence of this noise, the amount of variance and the mean of the error increase, i.e. by increasing the Gaussian noise variance from 0.01 to 0.1 (i.e. by increasing the amount of noise by 10 times), the mean error increases by about 2.5 times, and the error variance increases by about 3 times. Meanwhile by increasing the magnitude of the impulse noise from 0.01 to 0.1 (i.e. by increasing the amount of noise by 10 times), the average error increases by about 2 times, and the variance of the error increases by about 3 times.

Table 2. Comparing values of correct and incorrect Plank movement's coordinates in two-dimensional space after Gaussian noise effect with zero mean and variance (values * not found by Yolo network).

Organ name	Noise variance 0.01	Noise variance 0.02	Noise variance 0.05	Noise variance 0.1
Head	*	*	*	*
Shoulder	(395,280)	(304,280)	*	*
Elbow	(302,423)	(303,424)	(302,424)	(302,424)
Wrist	(97,423)	(97,423)	*	*
Hip	(533,291)	(533,300)	*	*
Knee	(756.308)	(756,307)	(757,307)	(758,307)
Ankle	(960,355)	(960,351)	(960,350)	(958,351)
Average error rate	(1.33,1)	(1.33,283)	(2.33,3)	((2.66,2.66)
Average variance rate	(1.46,1.2)	(1.06,13.36)	(2.33,19)	(4.33.14.33)

Table 3. Values of Yolo output coordinates impregnated with additive impulse noise with variable size.

Organ name	Noise variance 0.01	Noise variance 0.02	Noise variance 0.05	Noise variance 0.1
Head	*	*	*	*
Shoulder	(306,280)	(307,279)	(305,280)	(304,278)
Elbow	(305,423)	(306,422)	(304,422)	(303,423)
Wrist	96,422)	(98,423)	(99,423)	*
Hip	(533,292)	(534,291)	(535,293)	*
Knee	(755,308)	*	(755,308)	(755,308)
Ankle	(959,357)	(959,354)	(960,354)	(959,355)
Average error rate	(0.83,0.26)	(16,1.2)	(1.5,1.33)	(1,1.25)
Average variance rate	(0.56,0,26)	(1.3,2.7)	(2.3,1.86)	(0.66,1.58)

Table 4. Proposed maximum error for each sports movement across entire dataset`

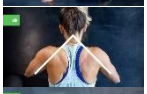
Exercise name	Correct form	Incorrect form	Maximum error
Bridge			4.14 cm 7.68 pixels
Donkey Kick			4.14 cm 7.68 pixels
Push-up			4.14 cm 7.68 pixels
Plank			4.14 cm 7.68 pixels
Squat			7.24 cm 7.68 pixels
Lunge			7.24 cm 7.68 pixels
Side Lunge			7.24 cm 7.68 pixels
Triceps Dip			7.24 cm 7.68 pixels

Table 4 reports the maximum error of the method on the dataset. In calculating the values related to the mentioned table, the following items are considered: 1) The image's resolution according to the YOLO input image standard is considered 256 x 256 [24]. 2) The maximum calculation error has been calculated based on AP75 of YOLO outcomes [25]. 3) According to [24], the maximum size for small objects is considered 0.12 of the image sizes. 4) As the landmarks introduced in the manuscript are considered in the category of small objects, the maximum diameter of the landmark is 30.72 pixels (256*0.12). 5) Due to the image frame's proportionality with its resolution, each pixel in the image will be equivalent to 0.944 cm in length and 0.54 cm in height. 6) Therefore, the maximum calculation error in the length related to the extraction of the location of the landmarks is 7.68 pixels or 7.24 cm. Also, the maximum calculation error in the height related to the extraction of the landmark's location is 4.14 pixels or 7.68 cm. It is necessary to explain that, according to the type of exercise, the error of each exercise in the dataset will be different.

5. Conclusion and Future Suggestions

In this work, we attempted to use Yolo neural network to extract the two-dimensional position of the joints from the image of the athlete. The Yolo network extracts the position of the joints by extracting the position of the wearable landmarks in the two-dimensional image. Then the skeletal model based on the position of the joints in two-dimensional space is calculated and compared with two-dimensional maps of the three-dimensional reference skeletal model. The difference between the two models shows the athlete's movement error. The proposed model shows effective stability. By increasing the cumulative Gaussian noise variance of the input data from 0.01 to 0.1, the resistance of the model is 2.5 ± 3 times the error compared to the normal state. Also, for cumulative impulse noise with a magnitude between 0.01 and 0.1, the resistance of the model is 2 ± 3 times the error compared to the normal state. The verification of this approach shows its accuracy and stability against additive Gaussian and impulse noises.

As a future work, and in order to develop the present approach, two suggestions can be pointed out: 1) Given that Yolo is a relatively heavy neural network, it is recommended to provide an alternative light model that can meet the great goals of research on mobile applications.

2) The proposed algorithm is based on wearable labels. Redesigning the mentioned algorithm that can calculate the position of body parts independently of these labels is suggested as a future work.

References

[1] A. Mousavi, A. Sheikh Mohammad Zadeh, M. Akbari, and A. Hunter, "*A New Ontology-Based Approach for Human Activity Recognition from GPS Data*," J. AI Data Min., vol. 5, no. 2, pp. 197–210, 2017, [Online]. Available: http://jad.shahroodut.ac.ir/article_889.html.

[2] S. Li and A. B. Chan, "*3D human pose estimation from monocular images with deep convolutional neural network*," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, Vol. 9004, pp. 332–347, doi: 10.1007/978-3-319-16808-1_23.

[3] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua, "*Structured Prediction of 3D Human Pose with Deep Neural Networks*," in Procedings of the British Machine Vision Conference 2016, 2016, vol. 2016-September, pp. 130.1-130.11, doi: 10.5244/C.30.130.

[4] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "*A simple yet effective baseline for 3d human pose estimation*," IEEE Int. Conf. Comput. Vis., May 2017, Accessed: Jun. 08, 2020. [Online]. Available: http://arxiv.org/abs/1705.03098.

[5] Y. Kudo, K. Ogaki, Y. Matsui, and Y. Odagiri, "*Unsupervised adversarial learning of 3d human pose from 2d joint locations*," arXiv:1803.08244v1, 2018, [Online]. Available: http://arxiv.org/abs/1803.08244.

[6] C. H. Chen et al., "*Unsupervised 3D pose estimation with geometric self-supervision*," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Vol. 2019-June, pp. 5707–5717, 2019, doi: 10.1109/CVPR.2019.00586.

[7] S. Tripathi, S. Ranade, A. Tyagi, and A. Agrawal, "*PoseNet3D: Unsupervised 3D Human Shape and Pose Estimation*," arXiv:2003.03473v1, 2020.

[8] N. Pourdamghani, H. R. Rabiee, F. Faghri, and M. H. Rohban, "*Graph based semi-supervised human pose estimation: When the output space comes to help*," Pattern Recognit. Lett., Vol. 33, No. 12, pp. 1529–1535, 2012, doi: 10.1016/j.patrec.2012.04.012.

[9] D. Pavllo, Z. Eth, and C. Feichtenhofer, "*3D human pose estimation in video with temporal convolutions and semi-supervised training*," CVPR, 2019.

[10] R. Mitra, N. B. Gundavarapu, A. Sharma, A. Ai, and A. Jain, "*Multiview-Consistent Semi-Supervised Learning for 3D Human Pose Estimation*," 2020.

[11] B. A. Olshausen and D. J. Field, "*Sparse coding with an overcomplete basis set: A strategy employed by V1?*," Vision Res., Vol. 37, No. 23, pp. 3311–3325, Dec. 1997, doi: 10.1016/S0042-6989(97)00169-7.

[12] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao, "*Robust estimation of 3D human poses from a single image*," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Sep. 2014, pp. 2369–2376, doi: 10.1109/CVPR.2014.303.

[13] V. Ramakrishna, T. Kanade, and Y. Sheikh, "*Reconstructing 3D human pose from 2D image landmarks*," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012, Vol. 7575 LNCS, No. PART 4, pp. 573–586, doi: 10.1007/978-3-642-33765-9_41.

[14] X. Zhou, M. Zhu, S. Leonardos, and K. Daniilidis, "*Sparse Representation for 3D Shape Estimation: A Convex Relaxation Approach*," IEEE Trans. Pattern Anal. Mach. Intell., Vol. 39, No. 8, pp. 1648–1661, Sep. 2017, doi: 10.1109/TPAMI.2016.2605097.

[15] E. J. Candès, M. B. Wakin, and S. P. Boyd, "*Enhancing sparsity by reweightedℓ1 minimization*," J. Fourier Anal. Appl., Vol. 14, No. 5–6, pp. 877–905, Dec. 2008, doi: 10.1007/s00041-008-9045-x.

[16] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "*Sparseness meets deepness: 3D human pose estimation from monocular video*," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2016, Vol. 2016-Decem, pp. 4966–4975, doi: 10.1109/CVPR.2016.537.

[17] X. Fan, K. Zheng, Y. Zhou, and S. Wang, "*Pose locality constrained representation for 3D human pose reconstruction*," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, Vol. 8689 LNCS, No. PART 1, pp. 174–188, doi: 10.1007/978-3-319-10590-1_12.

[18] M. Aharon, M. Elad, and A. Bruckstein, "*K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation*," IEEE Trans. Signal Process., Vol. 54, No. 11, pp. 4311–4322, Nov. 2006, doi: 10.1109/TSP.2006.881199.

[19] A. Rakotomamonjy, "*Applying alternating direction method of multipliers for constrained dictionary learning*," Neurocomputing, Vol. 106, pp. 126–136, Apr. 2013, doi: 10.1016/j.neucom.2012.10.024.

[20] B. Di Liu, Y. X. Wang, B. Shen, X. Li, Y. J. Zhang, and Y. J. Wang, "*Blockwise coordinate descent schemes for efficient and effective dictionary learning*," Neurocomputing, Vol. 178, pp. 25–35, Feb. 2016, doi: 10.1016/j.neucom.2015.06.096.

[21] W. Li et al., "*Maxdenominator Reweighted Sparse Representation for Tumor Classification*," Sci. Rep., Vol. 7, No. 1, pp. 1–13, Apr. 2017, doi: 10.1038/srep46030.

[22] M. Jiang, Z. Yu, Y. Zhang, Q. Wang, C. Li, and Y. Lei, "*Reweighted sparse representation with residual compensation for 3D human pose estimation from a single RGB image*," Neurocomputing, Vol. 358. pp. 332–343, 2019, doi: 10.1016/j.neucom.2019.05.034.

[23] H. Medvesek, "*Most Common Exercise Mistakes: Are You Doing It Wrong*?" https://www.runtastic.com/blog/en/bodyweight-exercise-mistakes/ (accessed Dec. 18, 2020).

[24] J. Redmon and A. Farhadi, "*YOLO v.3*," Tech Rep., pp. 1–6, 2018, [Online]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf.

[25] J. Xiao, "*ExYOLO: A small object detector based on YOLOv3 Object Detector*," Procedia CIRP, Vol. 188, No. 2019, pp. 18–25, 2021, doi: 10.1016/j.procs.2021.05.048.

اصلاح حرکات ورزشی مبتنی بر استخراج دوبعدی موقعیت مفاصل به کمک شبکه عصبی یولو و تطبیق با مدل خلوت اسکلتی سه بعدی

انیس راحتی و کامبیز رهبر*

گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، واحد تهران جنوب، دانشگاه آزاد اسلامی، تهران، ایران.

ارسال ۲۰۲۲/۰۶/۰۶؛ بازنگری ۲۰۲۲/۰۸/۰۵؛ پذیرش ۲۰۲۲/۰۹/۱۶

چکیده:

انجام صحیح حرکات ورزشی در تضمین سلامت بدن بسیار مهم است. در این پژوهش سعی شده است تا با استفاده از رویکردی متفاوت بر اساس موقعیت دوبعدی مفاصل از روی تصویر، اصلاح حرکات در فضای سه بعدی حاصل شود. تصویر ورودی، یک تصویر برچسب گذاری شده روی مفصل‌های بدن است. شخص در مقابل دوربین حرکات ورزشی را انجام می‌دهد. سپس مختصات مفصل‌ها در فضای دوبُعدی سنجیده می‌شود. اسکلت دوبُعدی با اسکلتهای دوبُعدی استخراج شده، از مدل خلوت اسکلتی مرجع تطبیق داده می‌شود تا اصلاحات حرکتی حاصل شود. راست آزمایی و دقت رویکرد یاد شده روی مجموعه داده استاندارد آدیداس انجام شده است. دهدهمچنین بازدهی آن تحت تاثیر نویز گوسی تجمعی و ضربه مطالعه شده است. میانگین خطای مدل در تشخیص حرکت اشتباه در مجموعه حرکات ورزشی ۵٫۶۹ پیکسل گزارش می‌شود.

کلمات کلیدی: تشخیص حرکات ورزشی، اصلاح حرکات ورزشی، استخراج دوبعدی موقعیت مفاصل، برچسب گذاری مفاصل، شبکه عصبی یولو، مـدل خلوت اسکلتی سه بعدی.