Multimodal Technologies and Interaction

MDPI

Article

Deep Learning-Enabled Multitask System for Exercise Recognition and Counting

Qingtian Yu *, Haopeng Wang, Fedwa Laamarti and Abdulmotaleb El Saddik

Multimedia Communications Research Laboratory, University of Ottawa, Ottawa, ON K1N 6N5, Canada; hwang266@uottawa.ca (H.W.); flaam077@uottawa.ca (F.L.); elsaddik@uottawa.ca (A.E.S.)
* Correspondence:qyu104@uottawa.ca

Abstract: Exercise is a prevailing topic in modern society as more people are pursuing a healthy lifestyle. Physical activities provide significant benefits to human well-being from the inside out. Human pose estimation, action recognition and repetitive counting fields developed rapidly in the past several years. However, few works combined them together to assist people in exercise. In this paper, we propose a multitask system covering the three domains. Different from existing methods, heatmaps, which are the byproducts of 2D human pose estimation models, are adopted for exercise recognition and counting. Recent heatmap processing methods have been proven effective in extracting dynamic body pose information. Inspired by this, we propose a deep-learning multitask model of exercise recognition and repetition counting. To the best of our knowledge, this approach is attempted for the first time. To meet the needs of the multitask model, we create a new dataset Rep-Penn with action, counting and speed labels. Our multitask system can estimate human pose, identify physical activities and count repeated motions. We achieved 95.69% accuracy in exercise recognition on the Rep-Penn dataset. The multitask model also performed well in repetitive counting with 0.004 Mean Average Error (MAE) and 0.997 Off-By-One (OBO) accuracy on the Rep-Penn dataset. Compared with existing frameworks, our method obtained state-of-the-art results.

Keywords: exercise; multitask system; heatmap; Rep-Penn dataset

Citation: Yu, Q.; Wang, H.; Laamarti, F.; El Saddik, A. Deep Learning-Enabled Multitask System for Exercise Recognition and Counting. *Multimodal Technol. Interact.* **2021**, *5*, 55. https://doi.org/10.3390/mti5090055

Academic Editor: Mu-Chun Su

Received: 9 July 2021
Accepted: 3 September 2021
Published: 8 September 2021

1. Introduction

Exercise is an inseparable part of people's daily lives, boosting human health physically and mentally. Technology innovation plays an increasingly significant role in improving exercise experience. In particular, Digital Twin coaching [1] is a promising area, which is starting to be explored. It allows us to provide individuals with a digital coach by utilizing advances in machine learning. The idea is inspired by the Digital Twin technology that El Saddik redefined to include the human Digital Twins [2]. This is an important redefinition, as it opens doors to the domain of coaching and sport to benefit from the Digital Twin technology. It is achieved by putting together specific technologies for the purpose of collecting data on the individual performing sport, analyzing the gathered data using machine learning and deep learning, and providing users with feedback and insight on their exercise performance [3].

Human pose estimation, action recognition and repetition counting are significant tasks in Digital Twin coaching. They provide precise body keypoints locations, workout categories and repetitive counting for the trainees. There are many works focusing on individual fields, and our paper integrates the three domains.

Human pose estimation is to recognize and locate the keypoints of the human body from RGB images or frames. These keypoints are connected to build an overview of the human torso. With the powerful feature extraction capabilities of CNN, both 2D and 3D human pose estimation fields have made considerable progress [4–6]. In this paper, we applied 2D human pose estimation because of its simplicity and stability. In general,

2D human pose estimation can be seen from two aspects: the detection problem and the regression problem. The detection-based methods [4,5,7] usually generate the pose estimation maps or heatmaps, whose pixels indicate the probability of one joint's location. Regression-based methods [6,8,9] output joints coordinates directly, which makes the end-to-end training process possible. They also allow joint positions to be leveraged by later tasks. The exponential growth of 2D human pose estimation contributes to many exercise pose guidance works. The user's joint positions are compared with the trainer's body. Accurate feedback helps exercisers correct training postures.

Two-dimensional (2D) action recognition recognizes the action type from RGB images automatically. The key to 2D action recognition is extracting spatial information from a single frame and temporal information from the whole video. Two-dimensional exercise recognition is a special branch of 2D action recognition because it mainly depends on body pose information. Therefore, joint motion information is significant for identifying exercise type, and video frames with whole body movement are necessary for exercise recognition. Many studies focus on utilizing joint motion information in action recognition. To make use of joint locations estimated by regress-based pose estimation methods, Luvizon et al. [10] generated a 3D matrix which is the concatenation of all the joint coordinates. Moreover, many skeleton-based action recognition works [11,12] achieved an excellent performance. Detection-based pose estimation methods are also applied in the action recognition task. Some researchers applied different colours to the heatmaps according to the relative time of the frame [13,14]. Other methods focus on extracting features from heatmaps to obtain intuitive temporal–spatial information [15,16].

Repetitive counting based on videos is always considered a dependant task. Many researchers made great achievements using signal processing methods before the machine learning wave came. Compared with these methods, machine learning methods are powerful at extracting similar information among the frames. Finding commonality between frames makes this task easier [17]. They usually use a symmetric matrix [18] to denote the similarity between two frames. What is more, approaches based on applied topology, sensors and wifi are also effective. Exercise repetitive counting is distinctive because it can leverage joint movement information. Few researchers work in this field. Alatiah et al. [19] utilized joint coordinates to derive the angle of each articulation. The number of cycles is counted by detecting the peaks of the angle change curve. Khurana et al. [20] extracted features from keypoints trajectory, and applied an off-the-shelf multilayer regressor to obtain the counting results.

Many up-to-date detection-based pose estimation methods achieve advanced accuracy, and their byproduct heatmaps include rich body motion information, which can be used in other tasks like exercise recognition and counting. Both tasks benefit from temporal joint motion information, which encourages us to build a multitask model sharing the same input features. Therefore, we establish a new multitask system including a 2D human pose estimator and an exercise recognition and counting multitask model. The whole system is illustrated in Section 3.1.

Our contributions can be summarized as follows:

- Design and advancement of an exercise multitask system combining human pose estimation, exercise recognition and repetition counting. The pose estimation model predicts joint locations, and provides byproduct heatmaps to the recognition and counting networks;
- Development of a strategy to utilize rich body motion information contained in heatmaps in the multitask of exercise identification and repetition counting for the first time;
- Building a new dataset called the Rep-Penn dataset based on the PennAction dataset. It covers seven exercises with nine different cycles and three action speeds.

2. Related Work

Human pose estimation, action recognition and repetition counting have made notable progress individually. There are also many works that focus on the interaction field of the

three domains by sharing features among different tasks. Since this paper works in 2D instead of in 3D, only literature relevant to the 2D field will be presented.

2.1. 2D Human Pose Estimation

As introduced above, 2D human pose estimation is usually divided into two groups: detection-based methods and regression-based methods. Detection-based methods usually output heatmaps, which predict the probability of the joint occurring at each pixel. These methods obtain joint coordinates indirectly, and they need post-processing such as applying a maximum filter to obtain the joint locations.

Wei et al. [5] introduced the heatmap to the human pose estimation task first. The proposed network consists of several stages. Each stage takes both the heatmaps from the previous stage and the feature map of the current stage as the input. Loss calculation is added in each stage to solve the gradient vanishing problem when training. Stacked Hourglass (SHG) Network [4] is a significant architecture, and lays the foundations for many pose estimation related tasks. It is able to catch information of all scales because small feature maps contribute to locating small body parts, while large-scale feature maps help to learn the full body information. MSPN [7] inherits the advantages of the SHG Network by sharing a similar architecture. It is able to utilize small-scale and large-scale features as well. Besides, it leverages a cross-stage aggregation technique by connecting features in different stages. We apply MSPN as our human pose estimator considering its high efficiency and accuracy.

Detection-based methods need non-differentiable processing steps on heatmaps to obtain the joint coordinates. Some researchers tried to replace the non-differentiable steps with differentiable functions so that joint locations can be estimated directly. These regression-based approaches are widely applied as well. Luvizon et al. [6] replaced the non-differentiable argmax function with the differentiable soft-argmax function. It can convert the highest response from feature maps to the coordinates. Nibali et al. [8] proposed a non-trainable differentiable layer called DSNT (Differential Spatial to Numerical Transform) layer. It is capable of calculating numerical coordinates from heatmaps immediately, and it achieves competitive pose estimation results.

Most detection-based methods have larger feature maps compared with regression-based methods. Thus, they have stronger spatial generalization abilities and obtain better accuracy. However, the architectures of these methods are not end-to-end networks, and regression-based methods have the advantage of generating joint positions directly.

2.2. 2D Action Recognition

Video-based 2D action recognition is a complex problem because it involves high-level feature extraction and the time dimension. In recent years, deep learning based methods have received more and more attention because of their strong feature processing abilities. The convolution operation is one of the basic parts of deep learning networks for the action recognition task. Though some researchers proposed single-frame action recognition architectures, which utilized 2D CNN models pretrained on other datasets, they overlooked the significance of temporal information in action recognition.

To make use of temporal information, researchers found that 3D CNN is an intuitive way to leverage temporal information from videos. Tran et al. [21] introduced an advanced version of 3D CNN called C3D by applying better kernel sizes for 3D CNN. Multi-streams approaches also play an important role in action recognition. These methods normally include two streams: the spatial stream and the temporal stream. The spatial stream focuses on getting static information from still frames, and the temporal stream obtains motion information from the whole video.

Despite obtaining the action type directly, many approaches concentrate on utilizing joints locations. Yang et al. [11] proposed a framework combining both CNN and LSTM. They also designed a novel skeleton structure to better express the joint information.

Ludl et al. [12] decoded the skeleton information by generating an Encoded Human Pose Image (EHPI) according to the joint type and location.

To make use of the rich information of the heatmap, which is a byproduct of detection based pose estimation methods, much of the literature created effective methods in leveraging heatmaps. Choutas et al. [13] colourized the heatmaps based on the order of the frames. These maps are temporally aggregated and fed into a classification network. Shah et al. [14] improved this method by reweighing motion information of various joints. Liu et al. [15] accumulated the heatmaps to create two images, which describe the temporal difference of torso shape and pose locations. The two features are fused, and a CNN model is applied to obtain the classification result. These works produced good results but they all need complicated processing on the heatmaps such as colourization. Liu et al. [16] made use of two features derived from heatmaps: DPI (Dynamic Pose Image) and DTI (Dynamic Texture Image). DPI contains numerous joint motion and body shape change information, and DTI stores a large amount of texture information. This method is straightforward and intuitive without complex processing procedures. What is more, DTI alone contains enough features, which we found can help with other tasks such as human exercise counting.

2.3. Repetitive Counting

Repetitive counting is an important task in the Computer Vision field. Much literature on repetitive counting commonly transforms the motion to a one-dimensional signal. Frequency information is extracted by signal processing methods such as Fourier Transform, peak detection and singular value decomposition. These methods assume the motion is periodic and stationary, which is unsuitable in many non-stationary situations. Therefore, Runia et al. [22] replaced Fourier Transform with Wavelet Transform. To handle camera movement and diversity in motion repetitions, they constructed a series of time-varying flow-based signals which are calculated in the motion foreground segmentation. However, this method failed to take contextual information into consideration.

The periodic detection problem can also be treated as a problem of finding commonalities between two video sequences. Panagiotakis et al. [17] proposed a symmetric matrix composed of two action sequences' pairwise difference. A highly efficient graph-based algorithm MUCOS and SMUCOS was proposed for this problem. It is reliable in the unsupervised situation when the semantic content of the videos, the number of periods and the valid duration of the video are unknown. Debidatta et al. [18] introduced a new symmetric matrix that helps achieve up-to-date accuracy. It acts as an intermediate layer to predict the cycle length and valid periodic length.

There are also many other methods that do not fall into the two divisions above. Levy and Wolf [23] employed a CNN model for the whole video to estimate the cycle length. They used two counters to record the number of repetitions so far and the index of the frame in one cycle respectively. The limitation is that cycle length is unchangeable in one video, which is not adaptable for actions with varied frequencies. Zhang et al. [24] proposed a new method by searching the positions of two neighbouring identical cycles. It overcomes the problems caused by diverse cycle lengths.

Finally, to the best of our knowledge, only a few works combine three tasks together [19,20]. Both of them utilize off-the-shelf human pose estimators to calculate the joint coordinates. Alatiah et al. [19] generated a new dataset, UCFRep, by augmenting the UCF101 dataset [25]. They fed the estimated 3D keypoints into a CNN model to derive the action class. For the counting task, parameters including major joints and type of motion are preselected. The major joints' angles of specific exercise are calculated from joint positions. Counting results and the correctness of the exercise are determined by the angle-time plot. This work proposes a reliable real-time system. However, exercise type and respective main joints should be set before counting. Khurana et al. [20] collected exercise videos from gym cameras. Different from our paper, their method can work in the multiple-people situation. The predicted keypoints are gathered to form motion trajectories. Then, the trajectories are grouped to each person. The processed features

are fed into a multilayer classifier and a multilayer regressor to achieve exercise category and counting results respectively. Their work has the advantage of working for multiple persons. However, its accuracy is limited and needs further improvement.

3. Proposed Methods

3.1. System Overview

An illustration of the proposed system is provided in Figure 1. The original inputs are RGB frames from an exercise video. MSPN is a 2D human pose estimation model, and it provides the heatmaps for calculating joint coordinates and the multitask model of exercise recognition & counting. Therefore, the heatmaps are processed in two ways. On the upper branch, max activating locations of the heatmaps are calculated to get the joint positions of the human body. The keypoints are connected to form a body skeleton, which provides visual feedback to the users. On the lower branch, heatmaps are transformed by the heatmap processing methods to extract body motion features. The processed heatmaps are then fed into the multitask model to recognize the exercise type and count the number of cycles. The following sections will introduce each block in detail.

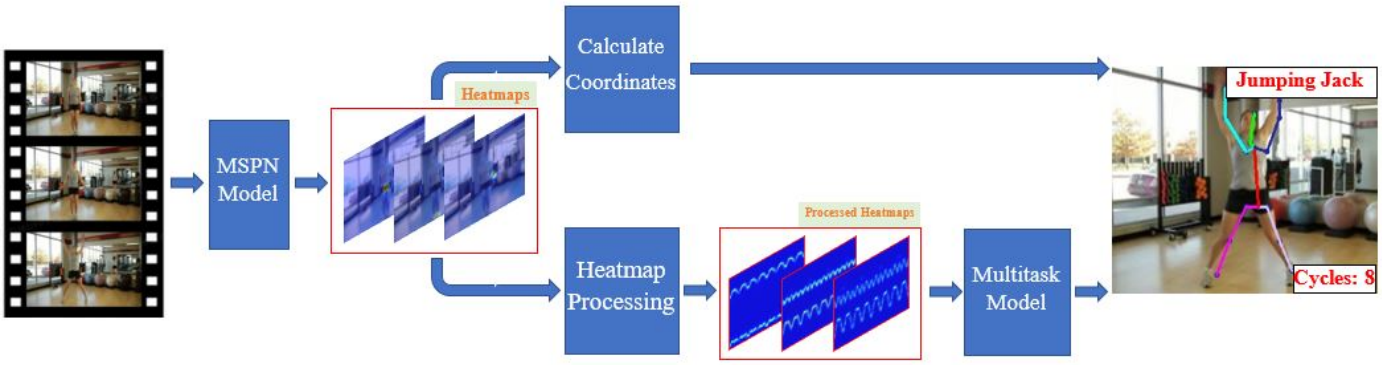

Figure 1. Illustration of the overall multitask system diagram.

3.2. MSPN Model

The MSPN model is a 2D human pose estimation model with up-to-date accuracy and efficiency. Because of the high accuracy and stability of MSPN, We applied it as our human pose estimation model.

3.2.1. Fine-Tuning

The MPII dataset [26] is a widely used dataset, which is tailored for 2D human pose estimation. It contains about 25 K human pose images with 16 body keypoints, but it does not cover the exercise videos we need. The PennAction dataset [27] includes 15 exercises with both actions and pose joints labels, but it only provides 2 K video samples, which is not enough to train a complex human pose estimation model like MSPN. What is more, only 13 joints are labelled without keypoints such as 'thorax', 'pelvis' and 'neck'. To make use of both datasets, we pretrained MSPN on the MPII dataset first. After that, MSPN is fine-tuned by being trained again on the PennAction dataset. The missing joints in the PennAction datasets will be treated as 0 s. It will not affect the training results as the missing joints are not counted in the total loss. Therefore, the model's generalization ability in the PennAction dataset is improved compared with using pretrained weights. It also outputs 16 joints' heatmaps, which help to describe the human body motion better.

3.2.2. Calculate Joint Coordinates

To estimate the joint locations from the heatmaps, extra processing methods should be applied. In this paper, the heatmaps are filtered by a Gaussian filter with a Standard Deviation (σ) equal to 0.5. Then, each heatmap is fed into a maximum filter to find the largest value of the heatmap. The footprint of the maximum filter is 3×3. The position of

the maximum is the predicted joint location. For heatmaps with all 0s, it indicates that the joints are either out of the detection area or get obscured.

3.3. Heatmap Processing

The heatmap is a 2D map that denotes the joint location probability. It can not only be used to calculate joint coordinates, but also represent the joint movement in the video. The joints' motion is a key feature in both exercise recognition and counting. Therefore, MSPN is leveraged as the human pose estimator to provide heatmaps based on the Rep-Penn dataset. It produces K ($K = 16$) heatmaps representing K joints locations per image. The generated heatmaps have the same height ($H = 64$) and width ($W = 64$), and the number of channels equals 1. Assume one video has F frames, we obtain a 4D feature $J \in R^{H \times W \times F \times K}$ by concatenating them together.

However, many up-to-date backbones with effective feature extraction abilities only allow for 3D feature inputs, and 4D features require a much larger model, which increases the complexity of the model greatly and slows down the training speed. Therefore, we calculate the mean of the first dimension and the second dimension of the 4D feature J respectively, and obtain 3D features $A \in R^{W \times F \times K}$ and $B \in R^{H \times F \times K}$:

$$A[w, f, k] = \frac{1}{H} \sum_{h=1}^{H} J[h, w, f, k] \tag{1}$$

$$B[h, f, k] = \frac{1}{W} \sum_{w=1}^{W} J[h, w, f, k]. \tag{2}$$

In this way, J is projected horizontally and vertically. Two 3D features A and B represent the ordinate movement and the abscissa movement respectively. We add A and B together as $P \in R^{(H+W) \times F \times K}$. Compared with J, the number of parameters of P is only $\frac{H+W}{H \times W}$ times of that of J. Suppose $H = W$, the number of parameters is reduced by $\frac{H}{2}$ times, which is 32 times in this work.

The pixel values of the heatmaps are between 0 and 1. We normalize the heatmaps to the scope of 0 to 255. The normalization formula is:

$$N = 255 \times \frac{P - min(P)}{max(P) - min(P)}. \tag{3}$$

Function $max(\cdot)$ calculates the maximum pixel value of the image P, while $min(\cdot)$ measures the minimum value. The normalized matrix N is then resized to $224 \times 224 \times K$.

Each joint has different motion patterns for different exercises. In order to make the joint movement more intuitive, let N^K represent the joint motion of the Kth joint $N^K \in R^{224 \times 224}$. An example of a one-period exercise is shown in Figure 2. This picture shows how the joint moves in a single cycle. The final position is almost the same as the original position, which suggests the end of one cycle. Our generated dataset includes exercises with multiple periods. An illustration of processed heatmaps calculated from a multi-cycle exercise video is displayed in Figure 3.

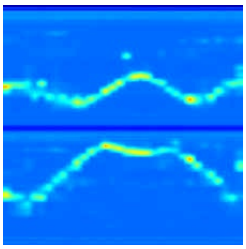

Figure 2. An example of single-cycle processed heatmap. The upper part is the ordinate change curve, and the below part is the abscissa change curve.

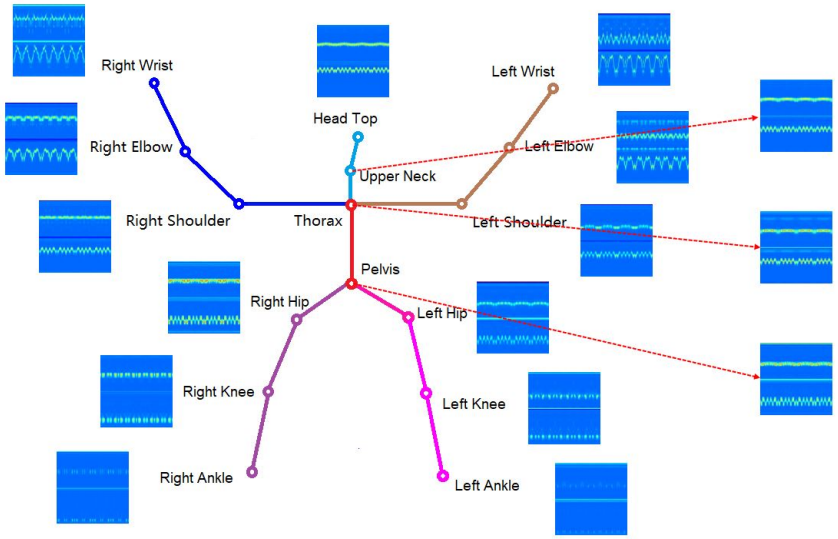

Figure 3. An example of processed heatmaps of a multi-period video. The example is based on an 8-cycle 'Jumping jack' exercise video. There are 16 joints in total. Each joint corresponds to a heatmap figure representing the joint motion in the whole video.

3.4. Multitask Model

In this paper, we proposed a multitask model of 2D exercise recognition and repetition counting. ResNet based networks are adopted as the backbones of both exercise recognition and repetition counting branches. The overview of the multitask model is presented in Figure 4.

We apply the ResNet34 network as the backbone of the exercise recognition part because of its simplicity and effectiveness. Exercise recognition is considered a classification problem. The output is a one-dimensional vector with length 512. To match the output size of ResNet34 and the number of classes of Rep-Penn which is seven, a Fully Connected Network (FCN) made of dense layers is added at the end of ResNet34. The FCN consists of three dense layers with output size 128, 28 and 7. After each dense layer, the Relu activation function is added, except for the last layer. The final layer is followed by a Softmax activation function.

Repetitive counting is treated as a regression problem. Its network is made of ResNet18 and an FCN. The FCN in the counting network includes four dense layers, with neuron sizes 128, 28, 7 and 1. Similar to the recognition task, Relu activation function is followed by each dense layer except the last dense layer. The linear activation function is placed at the end of the network.

Besides the two branches, several shared convolution layers are added to provide global features for these two tasks. The shared layers include two 3×3 convolution layers and two 5×5 convolution layers. All the layers have the same output size as the input.

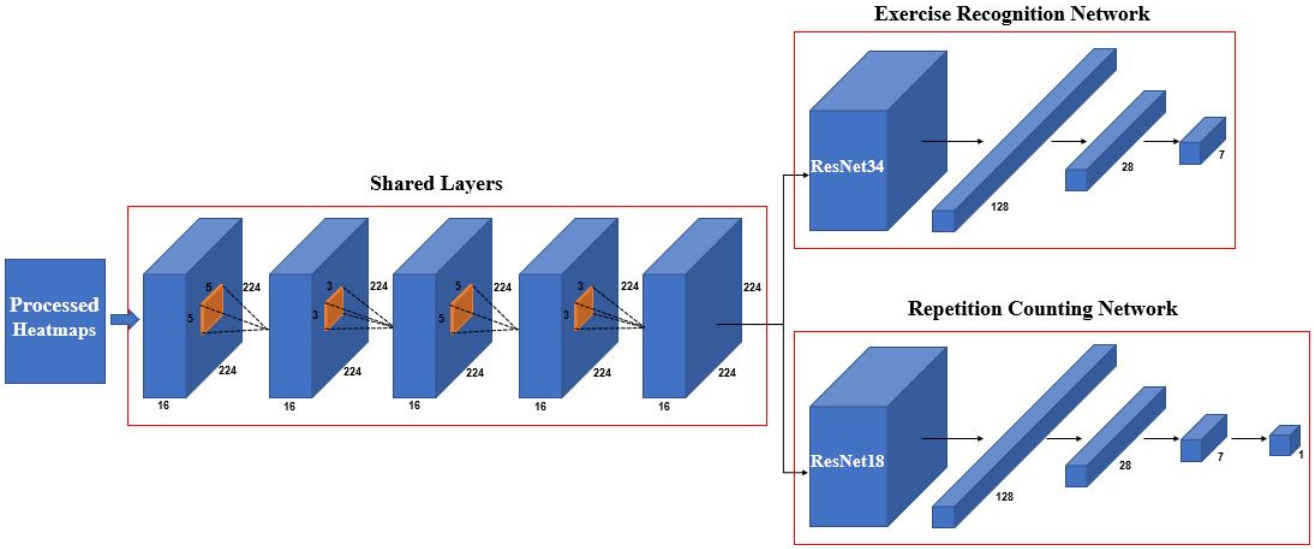

Figure 4. Illustration of the proposed multitask model. It consists of shared layers, an exercise recognition branch and a repetition counting branch.

3.5. Network Training

To train the multitask model of exercise recognition and counting, a two-stage training strategy is applied. At the first stage, we feed the multitask model with the Rep-Penn dataset and action labels. Both recognition network and shared convolution layers are trained, and the counting network is frozen. Categorical Cross-Entropy Loss is applied:

$$CE = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{C} y_{ni} log(p_{ni}), \tag{4}$$

where N is the number of samples, and C is the number of exercise classes. $y_n = [y_{n1}, y_{n2}, ..., y_{nc}]$ is the groundtruth one-hot label of the nth sample. if the sample belongs the ith class, $y_{ni} = 1$. If not, $y_{ni} = 0$. $p_n = [p_{n1}, p_{n2}, ...p_{nc}]$. Each element p_{ni} represents the estimated probability that the nth sample belongs to the ith category.

At the second stage, Rep-Penn dataset and counting labels are provided to the multitask model. The weights of the shared convolution layers along with the recognition layers are frozen, and only the counting network is trained at this stage. The loss function is MSE (Mean Squared Error):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{5}$$

In the formula above, N represents the amount of data. y_i represents the true counting number while $\hat{y}_i$ denotes the predicted counting result. By using the two-stage training strategy, our model can be trained to identify exercise recognition and count repetitions concurrently without extra steps.

To alleviate the effect of people in the training process, we separate individuals in the training dataset and the test dataset. The test dataset excludes pieces of information from the person who is included in the training dataset. Moreover, the inputs of the multitask model are processed heatmaps, which further reduce the influence of the people.

3.6. Rep-Penn Benchmark

In order to meet the needs of 2D exercise recognition and repetition counting multitask, we create the Rep-Penn dataset based upon the PennAction dataset [27]. The PennAction dataset only includes exercise videos with one cycle whilst multi-cycle physical activities are required in this paper. Therefore, we synthesize the Rep-Penn dataset by utilizing the

single-cycle exercise video frames. Inspired by dataset synthesis methods from [18], we connect the forward video with the rewinding video instead of connecting the original video repeatedly. In this way, continuity of the body movement is guaranteed, and exercise of various periods can be generated for repetition counting. Figure 5 applies processed heatmaps to explain how the video concatenation works. The images on the left side of the arrow are processed heatmaps of a one-period exercise video. *P* represents the video in positive order, and *R* represents the reverse order. The image on the right side denotes the processed heatmaps of a multi-cycle video, which is the concatenation of a single-cycle video.

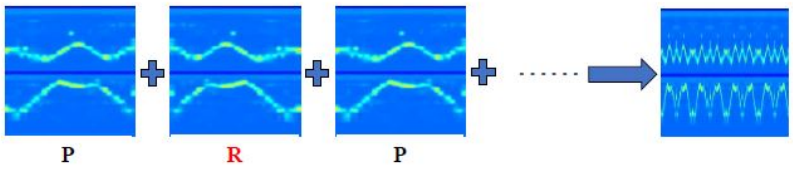

Figure 5. Illustration of creating a multi-cycle exercise video.

This paper is tailored for the multitask system of a one-set exercise, so we produce exercise videos with $6 \sim 14$ cycles considering the normal number of repetitions in a set of exercises. Besides different periods, we take motion speeds into consideration. We sample all the frames by taking one frame every C (C = 1, 2, 3) frames. As a result, three different action speeds are added to increase the diversity of the synthesized dataset. Figure 6 utilizes the heatmaps to illustrate how the number of cycles and speed differ in various workout videos. We can see from this image that different exercises have distinct joint motion patterns. The difference in moving speeds is not easily distinguished due to the heatmap size limitation. The change in the action period can be perceived effortlessly.

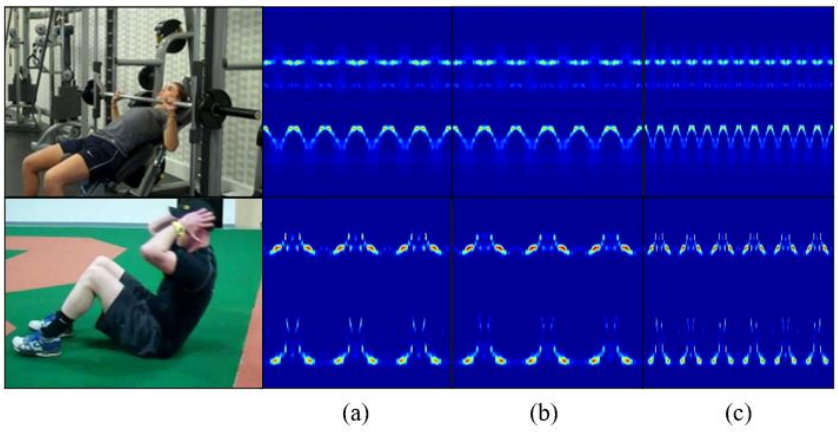

Figure 6. Explanation of videos with different cycles and speeds. This image shows the motion of right wrist in two exercises: bench press and sit up. (**a**) 6 cycles and low speed. (**b**) 6 cycles and fast speed. (**c**) 12 cycles with middle speed.

Therefore, each exercise video in the PennAction is enhanced to 27 videos with the same exercise, different cycles and various speeds. Among the 15 actions in the PennAction dataset, seven of them are included in Rep-Penn: bench press, clean and jerk, jumping jack, pull up, push up, sit up and squat. In total, we obtain 29,187 exercise videos. We separate the training dataset and test dataset at a ratio of 4:1, with 23,220 videos for training and 5967 videos for testing.

4. Experiments

In this section, our proposed multitask model is evaluated on the Rep-Penn dataset. Comparison between our approach and state-of-the-art methods are listed as well.

4.1. Implementation Details

We performed all the experiments on a computer with one Intel Core i7-9700K CPU, two Nvidia GTX1080ti GPUs with 24GB memory and 64GB RAM in total. The operation system of our computer is Ubuntu 16.04. The pose estimation model MSPN, exercise recognition and counting multitask model are all implemented by TensorFlow framework. Point Cloud Library and SciPy Library are used in heatmap processing and data preparation steps.

RMSprop optimizer is chosen for both training steps, and default parameters are applied. The initial learning rates are both set to 0.001, and the numbers of total epochs are both ten. The learning rate of the first training step is divided by ten after one and five epochs, and the learning rate is reduced by ten times at the 3rd and 6th epochs in the second training stage. In the whole training process, the batch size is eight.

4.2. Data Preparation and Evaluation Metrics

The size of original heatmaps produced by MSPN is 64×64. After being processed by the method introduced in Section 3.3, heatmaps are resized to 224×224. 16 heatmaps corresponding to 16 joints in one video are concatenated together to form a $224 \times 224 \times 16$ feature map as shown in Figure 7. The 3D feature maps are considered as basic samples, and they are shuffled before being fed into the recognition & counting multitask model.

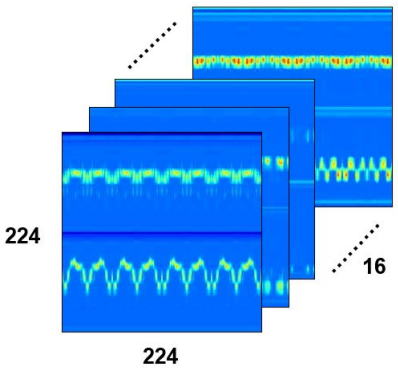

Figure 7. The illustration of stacking 16 heatmaps to one 3D feature map. The size of a single heatmap is 224×224. 16 heatmaps are combined together to form a feature map with the size of $224 \times 224 \times 16$.

We apply four metrics for exercise counting and one metric for action recognition. The four counting metrics include Mean Absolute Error (MAE), Off-By-One (OBO), Average Error (AE) and Standard Deviation (σ). Accuracy is leveraged as our exercise recognition standard.

MAE is the criteria used in many other baseline methods for repetitive counting tasks. We calculate the sum of the absolute difference between the ground truth label G and the predicted counting result P, and then divided by the ground truth G : $\frac{|G-P|}{G}$. Let N represent the number of samples. MAE is the average of the normalized absolute difference in the whole dataset:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\frac{|G_i - P_i|}{G_i}. \tag{6}$$

OBO is also a significant metric in counting tasks. If the difference between the predicted count and the ground truth value is within 1, the sample is correctly classified. Otherwise, it is noted as misclassification.

$$OBO = \frac{Num[(G-P) <= 1]}{Num(G)}, \tag{7}$$

where $Num[\cdot]$ represents the number of valid values. Since repetitive counting is widely considered as a regression problem, OBO can describe the performance of counting predictor well.

AE represents the average counting error. The formula is similar with MAE but it is not divided by groundtruth value G:

$$AE = \frac{1}{N}\sum_{i=1}^{N}|G_i - P_i|. \tag{8}$$

Standard Deviation (σ) measures the amount of variation or dispersion of a set of values. In the counting task, it is usually used along with MAE or AE to denote the dispersion of the counting results.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(G_i - P_i)^2}. \tag{9}$$

For exercise recognition task, we calculate the number of correct predictions (C) divided by the number of samples (N):

$$Accuracy_{exe} = \frac{C}{N}. \tag{10}$$

4.3. Results and Analysis

4.3.1. Exercise Recognition

The test dataset of Rep-Penn includes 5967 test samples covering seven exercises: bench press, clean and jerk, jumping jack, pull up, push up, sit up and squat. Table 1 shows the prediction accuracy of each exercise on our multitask model. We got 95.69% accuracy among the seven exercises, and all the exercises' accuracy is above 90%, except 'Sit up'. 'Clean and jerk' gets the highest accuracy at 99.83% while 'Sit up' reaches around 89% accuracy.

Table 1. Recognition accuracy for each type of exercise.

Exercises	Bench Press	Clean and Jerk	Jumping Jack	Pull Up	Push Up	Sit Up	Squat	Overall
Accuracy	96.53%	99.83%	93.25%	95.91%	93.83%	88.89%	97.65%	95.69%

Table 2 is the confusion matrix of the exercise recognition. Among the seven exercises, 'Clean and jerk' and 'Squat' are the most easy-to-distinguish exercises. We can see that 'Clean and jerk' only has one misclassification among all the 594 test samples, and 'Squat' owns 33 incorrect estimations among 1215 test data. Nevertheless, 'Sit up' has the highest error with 45 incorrect predictions in 405 samples. Most wrong predictions fall on 'Bench press' and 'Pull up', which means that the model tends to mistake 'Sit up' for 'Bench press' or 'Pull up'. For other exercises, most of them have less than seven percent incorrect predictions which are acceptable.

Table 2. Confusion matrix for exercise recognition across 7 exercises.

7 Exercises	Bench Press	Clean and Jerk	Jumping Jack	Pull Up	Push Up	Sit Up	Squat
Bench press	834	0	0	0	30	0	0
Clean and jerk	0	593	0	0	1	0	0
Jumping jack	0	0	428	21	10	0	0
Pull up	2	0	10	984	25	5	0
Push up	21	0	1	27	1140	0	26
Sit up	12	3	0	27	3	360	0
Squat	0	4	0	21	0	8	1371

We apply the two-stage training strategy in training the multitask model. Exercise recognition layers are trained first based on several networks: ResNet18, ResNet34, ResNet50 and ResNet101. The results are displayed in Table 3. From this table, we can see that ResNet34 based network obtains the best exercise recognition accuracy at 95.69%. ResNet18 and ResNet50 achieve very close results with 94.25% and 93.78% respectively. ResNet101 has the poorest accuracy at 91.20%. Even though ResNet 34 is larger than ResNet18, we select ResNet34 as the main part of the exercise recognition branch because precision is our primary goal.

Table 3. Comparison of feature extraction networks for exercise recognition task.

Network	Accuracy (%)	Parameters
Ours-ResNet18	94.25	11,323,303
Ours-ResNet34	95.69	21,442,599
Ours-ResNet50	93.78	26,263,651
Ours-ResNet101	91.20	45,334,115

4.3.2. Repetition Counting

To find the most suitable feature extraction network for the counting task, ResNet18, ResNet34 and ResNet50 are tested as the feature extraction network. The comparison is shown in Table 4. It lists the counting performance of the three networks. ResNet18 is the smallest network with around 11 million parameters. It achieves excellent accuracy with 0.004 MAE and 0.997 OBO. ResNet34 is about two times the size of ResNet18, and it obtains competitive results in the counting task with 0.006 MAE and 0.998 OBO. ResNet50 is the largest network but performs worst. After considering the accuracy and network size of ResNet18 and ResNet34, ResNet18 is selected as the backbone of the counting network in the multitask model. The MAE is less than 0.005, and AE is around 0.04, which means the counting error is small compared with groundtruth counting labels. The Standard Deviation is only 0.172, suggesting small error dispersion in the prediction results. OBO is above 99%, denoting that almost all the test samples are within ± 1 of groundtruth.

Table 4. Comparison of feature extraction networks for counting tasks. Four standards are displayed in this table: (MAE) Mean Average Error, OBO (Off-By-One), AE (Average Error) and σ (Stand Deviation), which are introduced in Section 4.2. This comparison is made under the premise that the exercise recognition backbone is ResNet34.

Network	MAE ↓	OBO ↑	AE ↓	σ ↓	Parameters ↓
ResNet18	0.004	0.997	0.039	0.172	11,323,575
ResNet34	0.006	0.998	0.059	0.189	21,442,871
ResNet50	0.015	0.996	0.154	0.261	26,444,247

4.3.3. Discussion

The processed heatmaps contain both spatial and temporal information about the body joints. It requires the deep learning networks to have a proper number of convolution layers. For exercise recognition, the movements of different joints regarding various exercises are key features. ResNet18 is relatively shallow, and it does not contain enough convolution layers to learn from extracted features. However, the heatmaps are not that complicated. ResNet50 and ResNet101 are too deep to improve the accuracy compared with ResNet34. For repetition counting, the joint movement features are more intuitive as the multitask model only needs to learn how to predict the start point and end point of each repetition. Therefore, ResNet18 is deep enough to make the most use of the heatmaps. Applying deeper convolutional networks will decrease the model's performance.

4.3.4. Output Format

The input of the multitask system is a single-person exercise video. The output is the same video noted by the body skeleton, exercise type and the number of cycles. The examples of output frames are displayed in Figure 8. The body skeleton is coloured and shown on top of the human body. The label in the upper right corner of the frame represents the exercise type, and the number in the bottom right corner denotes the number of repetitions.

Figure 8. Examples of output frames. It includes all the 7 exercises with various periods.

4.4. Comparison with Other Methods

There are only a few papers [19,20] focusing on exercise recognition and repetition counting multitask based on RGB images. Both of them create their own datasets which are unavailable. Alatiah et al. [19] generated training data covering three exercises: pull up, push up and squat. Therefore, we compare with their work in both tasks based on these three same exercises. Khurana et al. [20] included 17 actions in total, among which four exercises are the same as this paper. However, some of them are easily misclassified

due to the limitation of training data. Thus, the seven most frequent exercises among the 17 exercises are selected for comparison in exercise recognition. In repetition counting, we also include Zhang et al.'s paper [24] besides the mentioned two papers because they created a new dataset UCFRep based on UCF101, and they achieved up-to-date accuracy in the counting task. In the end, the effect of heatmaps is explored by comparing our method and applying joint positions directly.

4.4.1. Exercise Recognition

Alatiah et al.'s work [19] shares three of the same exercises: pull up, push up and squat with our work. We trained our multitask model by only using data of these three physical activities. Table 5 compares the accuracy between their work and this paper across the three exercises. It is shown that the precision of 'Pull up' and 'Push up' and Recall of 'Squat' of this work is higher than in their work, but other metrics of our work are a bit lower than theirs. The reason is that they applied the reject option, which rejects the ambiguous samples if the estimated probability is out of the selected confidence intervals. Despite this, our method is still competitive as all three metrics are within 0.03 of Alatiah's results.

Table 5. Comparison with paper [19] over 3 exercises: pull up, push up and squat. The metrics with * denotes the result of this paper while the metrics without * represent the accuracy of paper [19].

Class	Precision	Recall	F1-Score	Precision *	Recall *	F1-Score *
Pull up	0.975	0.982	0.979	0.979	0.955	0.968
Push up	0.975	0.968	0.972	0.987	0.947	0.967
Squat	0.992	0.989	0.990	0.943	0.992	0.967

We also compare our proposed method with Khurana et al.'s [20] research. They collected exercise videos from gym cameras. In summary, videos including 17 exercises were gathered, and they achieved overall 80.60% exercise recognition accuracy. Since their dataset is unavailable, we compare the recognition accuracy of the seven most frequent exercises they collected with our work. Their 85.7% accuracy over seven exercises is lower than our 95.69% accuracy. The comparison above shows that our exercise recognition branch has up-to-date accuracy.

4.4.2. Repetition Counting

In this section, we compared our work with paper [19] across three identical exercises first. Afterwards, three papers [19,20,24] and our work are combined for comparison.

Compared with [19], our AE is ± 0.242 lower than [19]'s ± 1 across the three exercises. Their counting part is based on a joint changing curve. It requires preselected action type and counts the repetitions by detecting signal peaks. However, our work is based on machine learning, and we do not need extra steps to derive the results.

The comparison between our method and three other works [19,20,24] is listed in Table 6. Note that the result is affected by the dataset limitation. Compared with [24], our MAE 0.004 is lower than their 0.147, and OBO 0.99 is much higher than their 0.79. The standard deviation of our work 0.187 is also smaller than the 0.243 of the method in [24]. Their dataset, UCFRep, is more challenging than Rep-Penn. However, our dataset includes videos with more cycles than UCFRep (max of cycles: seven), and more exercise types than UCFRep (number of exercises: five).

In [19,20], they did not clarify how they calculated the error. We refer to the related literature, and consider it as AE because it represents the average error per prediction. It is shown that our AE is more accurate than theirs. Moreover, the standard deviation of our paper, 0.187, is lower than that in [20] (2.64). Our counting branch is verified to be effective and accurate.

Table 6. Counting accuracy comparison between the four works. Khurana's work is based on 17 exercises gathered from a gym camera. Alatiah's research and Zhang's work leverage data from UCF101 dataset with 3 exercises and 24 daily activities individually. Our paper includes 7 exercises, and the data are originally derived from PennAction dataset.

Method	MAE ↓	OBO ↑	AE ↓	σ ↓
Khurana et al. [20]	-	-	±1.7	2.64
Alatiah et al. [19]	-	-	±1	-
Zhang et al. [24]	0.147	0.79	-	0.243
Proposed method	0.004	0.99	±0.039	0.187

4.4.3. Comparison with Joint-Based Methods

To explore the influence of applying heatmaps in our method, using joint locations and heatmaps are compared in this section. We calculate the joint coordinates, and connect the ordinates and abscissas respectively to create a similar feature map to that in Figure 2. The comparison is listed in Table 7. Exercise recognition accuracy and OBO of applying heatmaps are higher than those of using joint positions. MAE, AE and σ of utilizing heatmaps are lower than those of leveraging joint locations. Therefore, all the metrics of using heatmaps are better than utilizing joint coordinates. It proves that heatmaps can better express the movement and distribution of body joints. Applying heatmaps achieves advanced performance in both tasks.

Table 7. Results of exercise recognition and counting using different inputs: heatmaps and joint coordinates.

Input Data	Exercise Recognition	Repetition Counting			
	Accuracy (%)	MAE ↓	OBO ↑	AE ↓	σ ↓
Heatmaps	95.69	0.004	0.99	±0.039	0.187
Joint coordinates	93.43	0.012	0.98	±0.055	0.203

4.5. Further Discussion

To the best of our knowledge, off-the-shelf datasets do not support human exercise recognition and repetitive counting multitask. Action recognition datasets like UCF101 [25] and PennAction [27] include in-door exercises, but there are no counting labels available. Datasets tailed to repetitive counting like QUVA [22] and YTsegments [23] are short of data regarding indoor exercises. In this case, we did not utilize common datasets to evaluate different methods. All our experiments are based on the Rep-Penn dataset.

As introduced above, Rep-Penn is generated by concatenating single-period exercise frames from the PennAction dataset. It excludes the exercises with shifting view angles. Besides, Rep-Penn only includes continuous workouts with fixed cycle lengths. The limitations of the PennAction dataset and the Rep-Penn dataset also apply to the proposed multitask model. It lacks the ability to recognize exercises when the camera is moving or when the body moves intermittently. Therefore, various exercise data should be collected and trained on the multitask model. Moreover, the proposed system only works for a single person. A multi-person multitask system can be explored to allow the system to work for multiple exercisers.

5. Conclusions

In this paper, we propose a multitask system including an off-the-shelf 2D human pose estimation model MSPN and a multitask model of 2D exercise recognition and repetition counting. We are among very few works to propose a system covering these three fields together. Furthermore, to the best of our knowledge, it is the first time that a 2D exercise recognition and counting multitask model learned from heatmaps produced by the human

pose estimator. The high accuracy achieved by heatmap-based pose estimation methods encourages us to explore the rich information contained in the heatmaps. Inspired by the outstanding performance of heatmap processing methods invented by Liu et al. [16], we utilize this methodology to the multitask for the first time. Due to the dataset limitation, we create a new dataset, Rep-Penn, based on the PennAction dataset. Various cycles and action speeds are added to enrich the Rep-Penn dataset. Compared with related work, we reach solid accuracy in both exercise recognition and counting. Our future work will focus on collecting more diverse data, and will develop multi-person models in the multitask system.

Author Contributions: Conceptualization, Q.Y., F.L. and A.E.S.; Data curation, Q.Y.; Formal analysis, Q.Y. and F.L.; Investigation, Q.Y. and H.W.; Methodology, Q.Y. and H.W.; Project administration, F.L. and A.E.S.; Software, Q.Y. and H.W.; Supervision, F.L. and A.E.S.; Validation, Q.Y.; Visualization, Q.Y. and H.W.; Writing—original draft, Q.Y.; Writing—review & editing, Q.Y., H.W., F.L. and A.E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Gámez Díaz, R.; Yu, Q.; Ding, Y.; Laamarti, F.; El Saddik, A. Digital Twin Coaching for Physical Activities: A Survey. *Sensors* **2020**, *20*, 5936. [CrossRef]
2. El Saddik, A. Digital Twins: The Convergence of Multimedia Technologies. *IEEE Multimed.* **2018**, *25*, 87–92. [CrossRef]
3. Saddik, A.E.; Laamarti, F.; Alja'Afreh, M. The Potential of Digital Twins. *IEEE Instrum. Meas. Mag.* **2021**, *24*, 36–41. [CrossRef]
4. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 483–499.
5. Wei, S.E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional Pose Machines. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4724–4732.
6. Luvizon, D.C.; Tabia, H.; Picard, D. Human pose regression by combining indirect part detection and contextual information. *Comput. Graph.* **2019**, *85*, 15–22. [CrossRef]
7. Li, W.; Wang, Z.; Yin, B.; Peng, Q.; Du, Y.; Xiao, T.; Yu, G.; Lu, H.; Wei, Y.; Sun, J. Rethinking on Multi-Stage Networks for Human Pose Estimation. *arXiv* **2019**, arXiv:abs/1901.00148.
8. Nibali, A.; He, Z.; Morgan, S.; Prendergast, L. Numerical Coordinate Regression with Convolutional Neural Networks. *arXiv* **2018**, arXiv:abs/1801.07372.
9. Sun, X.; Xiao, B.; Wei, F.; Liang, S.; Wei, Y. Integral Human Pose Regression. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
10. Luvizon, D.C.; Picard, D.; Tabia, H. 2d/3d pose estimation and action recognition using multitask deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5137–5146.
11. Yang, Z.; Li, Y.; Yang, J.; Luo, J. Action recognition with spatio–temporal visual attention on skeleton image sequences. *IEEE. Trans Circuits. Syst. Video. Technol.* **2018**, *29*, 2405–2415. [CrossRef]
12. Ludl, D.; Gulde, T.; Curio, C. Simple yet efficient real-time pose-based action recognition. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 581–588.
13. Choutas, V.; Weinzaepfel, P.; Revaud, J.; Schmid, C. PoTion: Pose MoTion Representation for Action Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7024–7033. [CrossRef]
14. Shah, A.; Mishra, S.; Bansal, A.; Chen, J.C.; Chellappa, R.; Shrivastava, A. Pose And Joint-Aware Action Recognition. *arXiv* **2020**, arXiv:2010.08164.
15. Liu, M.; Yuan, J. Recognizing Human Actions as the Evolution of Pose Estimation Maps. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1159–1168. [CrossRef]
16. Liu, M.; Meng, F.; Chen, C.; Wu, S. Joint dynamic pose image and space time reversal for human action recognition from videos. In Proceedings of the AAAI Conference on Artificial Intelligence, Hawaii, HI, USA, 27 January–1 February 2019; pp. 8762–8769.
17. Unsupervised Detection of Periodic Segments in Videos. In Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 923–927. [CrossRef]

18. Dwibedi, D.; Aytar, Y.; Tompson, J.; Sermanet, P.; Zisserman, A. Counting Out Time: Class Agnostic Video Repetition Counting in the Wild. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Online, 14–19 June 2020; pp. 10384–10393.
19. Alatiah, T.; Chen, C. Recognizing Exercises and Counting Repetitions in Real Time. *arXiv* **2020**, arXiv:abs/2005.03194.
20. Khurana, R.; Ahuja, K.; Yu, Z.; Mankoff, J.; Harrison, C.; Goel, M. GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *185*, 1–17. [CrossRef]
21. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 4489–4497. [CrossRef]
22. Runia, T.F.H.; Snoek, C.G.M.; Smeulders, A.W.M. Real-World Repetition Estimation by Div, Grad and Curl. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake, UT, USA, 18–22 June 2018; pp. 9009–9017. [CrossRef]
23. Levy, O.; Wolf, L. Live repetition counting. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; Volume 2015, pp. 3020–3028. [CrossRef]
24. Zhang, H.; Xu, X.; Han, G.; He, S. Context-aware and scale-insensitive temporal repetition counting. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Online, 14–19 June 2020; pp. 667–675. [CrossRef]
25. Soomro, K.; Zamir, A.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv* **2012**, arXiv:abs/1212.0402.
26. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3686–3693.
27. Zhang, W.; Zhu, M.; Derpanis, K.G. From Actemes to Action: A Strongly-Supervised Representation for Detailed Action Understanding. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2248–2255. [CrossRef]