Deep Learning-Enabled Multitask System for Exercise Recognition and Counting

by

Qingtian YU

Thesis submitted to the University of Ottawa
In partial fulfillment of the requirements
For the M.A.Sc. degree in
Electrical and Computer Engineering

uOttawa

School of Electrical Engineering and Computer Science, Faculty of Engineering,
University of Ottawa

Abstract

Exercise is a prevailing topic in modern society as more people are pursuing a healthy lifestyle. Physical activities provide unimaginable benefits to human well-being from the inside out. 2D human pose estimation, action recognition and repetitive counting fields developed rapidly in the past several years. However, few works combined them together as a whole system to assist people in evaluating body poses, recognizing exercises and counting repetitive actions. The existing methods estimate pose positions first, and utilize human joints locations in the other two tasks. In this thesis, we propose a multitask system covering the three domains. Different from the methodology used in the literature, heatmaps which are the byproducts of 2D human pose estimation models are adopted for exercise recognition and counting. Recent heatmap processing methods are proven effective in extracting dynamic body pose information. Inspired by this, we propose a new deep-learning multitask model of exercise recognition & repetition counting, and apply these approaches to the multitask 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. A two-stage training strategy is applied in the training process. Our multitask system can estimate human pose, identify physical activities and count repeated motions. We achieved 95.69% accuracy in exercise recognition on 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 Rep-Penn dataset. Compared with existing frameworks, our method obtained state-of-the-art results.

Acknowledgements

I would like to express my deep gratitude to my supervisor, Prof. Abdulmotaleb El Saddik for giving me the opportunity to be a member of MCRlab. It's my great honour to do research with him. His passion, vision and motivation inspired me to overcome difficulties and work for my goal. He provided me with insightful suggestions not only in research but also in life. I am really grateful for his kindness, humour and friendship. It's a great privilege to work with him under his guidance.

I also want to say thank you to all the MCRlab members. I really appreciate their advice and guidance in my thesis. Their questions in the lab meeting always make me clearer about my way forward. In particular, I am really grateful to Haopeng Wang for his patient guidance and selfless help. I am also thankful to Dr. Fedwa Laamarti for her encouragement and insightful advice.

Finally, I would like to thank my parents for their unconditional support mentally and financially. Their love, kindness and understanding encourage me to finish my thesis and prepare for my future.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Exercise is an inseparable part of people's daily life, which boosts human health physically and mentally. Technology innovation plays an increasingly significant role in improving exercise experience. In particular, Digital Twin coaching [23] is a promising area that is starting to be explored. It allows providing 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 [19]. 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 in collecting data on the individual performing sport, analyzing gathered data using machine learning and deep learning, and providing users with feedback and insight on their exercise [54].

Human pose estimation, action recognition and repetition counting are significant tasks in Digital Twin coaching. They provide precise body keypoints locations, exercise category and repetitive counting for the trainees. Many works focus on individual fields, but few works treat them as a whole system. The existing systems perform poorly in action recognition and counting, which need to be further improved. To fill this gap, we introduce a deep-learning based multitask system tailored to exercise including 2D human pose estimation, exercise recognition and repetition counting.

The goal of 2D 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 (Convolutional Neural Network), the 2D human pose estimation field has made considerable progress [47, 74, 46]. Its exponential growth contributes to exercise pose guidance [33, 73]. The user's joint positions are compared with the trainer's body. Accurate feedbacks help exercisers correct training postures.

In general, 2D human pose estimation can be divided into two subtasks: SPPE (Single Person Pose Estimation) and MPPE (Multi Person Pose Estimation). SPPE aims to find all the keypoints of one person. The task can be seen from two aspects: detection problem or regression problem [45]. The detection-based methods usually generate the pose estimation maps or heatmaps [74]. The heatmap is a 2D feature map like Figure 1.1, whose pixels indicate the probability of one joint's location. The left image is the original image. The right one is the initial image covered by a heatmap. This heatmap predicts the location of the head. Its highest value is denoted as the red spot which is

exactly the head position. One joint corresponds to one heatmap. To obtain the joint coordinates, post processing is required to find the position with the largest probability.

On the contrary, regression-based methods can provide joints coordinates directly. Carreira et al. [8] introduced a self-refining model which corrects the joint positions by checking prediction errors iteratively. Paper [65, 46, 48] added a differential network between heatmaps and joints' coordinates which makes the end-to-end training process possible. The regression-based methods allow the joint coordinates to be used directly by subsequent tasks. However, they tend to have poorer accuracy than detection-based methods because of smaller feature maps.

Figure 1.1: An example of a heatmap representing the head position.

MPPE is usually divided into two methods: Top-down and Bottom-up [16]. Top-down: Detecting all the people's bounding boxes first, and then SPPE is applied to each person. Bottom-up: Locating all the keypoints in one frame. The keypoints are then grouped to each subject. The strength of Top-down methods is higher accuracy, but Bottom-up approaches have faster speed. Since MPPE is not the focus of this thesis, it won't be discussed in detail further.

2D action recognition is to recognize 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. For many action recognition methods, spatial information includes body poses, carrying objects and background. Temporal

information is the change of spatial information in the video. 2D exercise recognition is a special branch of 2D action recognition because it only requires body pose information in the whole video. Therefore, joint motion information is significant in identifying exercise type, and frames with whole-body movement are necessary for exercise recognition.

Lots of research focuses on utilizing joint motion information in action recognition, which provides us with tons of inspiration for exercise recognition. To make use of joint locations estimated by regress-based pose estimation methods, paper [45, 44] generates a 3D matrix which is the concatenations of all the joint coordinates. Moreover, many skeleton-based action recognition works [76, 43] also achieve excellent performance.

Detection-based pose estimation methods are also applied in the action recognition task. Paper [13] applied different colours to the heatmaps according to the relative time of the frame. Some researchers accumulated the heatmaps to create two images that describe the shape transformation and joint location changes throughout the video [40]. Liu et al. [41] presented novel heatmap processing methods, and they produced two feature images DPI (Dynamic Pose Image) and DTI (Dynamic Texture Image). DPI contains numerous joint motion and body shape change information while DTI stores large amounts of texture information. DPI and DTI are fused in the end, and this method achieves outstanding accuracy. Moreover, experiments suggest that DPI alone is able to reach competitive action recognition results.

These methods mainly focus on processing heatmaps to get intuitive temporal-spatial information for the action recognition task. However, they ignore the fact that the processed heatmaps contain rich body movement information which can be utilized in other tasks like the repetitive counting task.

Repetitive counting based on videos is always considered as a dependant task. Many researchers made great achievements using signal processing methods before the machine learning wave came [39, 4, 50, 3, 63, 51]. Compared with these methods, machine-learning based methods are powerful to extract similar information among the frames. Finding commonality between frames makes this task easier [15, 49]. They usually use

a symmetric matrix [18] to denote the similarity between two frames. What's more, Topology-based methods [71, 68], sensor-based methods [24, 63], wifi-based approaches [42] are also effective.

Exercise repetitive counting is distinctive because it can leverage joint movement information. Few researchers work in this field. Alatiah et al. [1] focus on 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. [35] 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 & counting multitask model. The whole system is illustrated in Section 3.1.

1.2 Challenges

The goal of the thesis is to build a multitask system including 2D human pose estimation, 2D exercise recognition and counting for exercises based on videos. Detection-based 2D pose estimation methods normally get better estimation accuracy, and the byproduct heatmaps contain rich pose location and body shape information. However, a single heatmap is meaningless as it only represents one joint's location in a particular frame. Simply tiling all the heatmaps in one video would result in a huge input feature map which requires a complicated model. Therefore, how to process the heatmaps to a small-scale feature map that contains enough joint motion information is a challenging task.

Most off-the-shelf datasets don't support human exercise recognition and repetitive

counting multitask. Action recognition dataset like UCF-101 [62] and PennAction [80] include exercises, but there are no counting labels available. Datasets tailed to repetitive counting like QUVA [53] are short of data regarding exercises. In this case, generating a dataset that includes a variety of repetitive exercises with both action type and counting labels is a big challenge for us. The dataset should also take factors like body moving speeds, different cycles and action continuity into consideration.

Another challenge is the multitask model network. Exercise recognition and counting networks can not be placed in series because neither task leverages the output of the other one. Meanwhile, both tasks can learn from body joint movement information. Therefore, we can build a multitask model by applying a parallel architecture. Global features can be shared between these two networks. In addition, Exercise recognition is considered as a classification problem while counting is regarded as a regression problem. Both tasks are complicated, and strong feature extraction networks should be applied.

In summary, the challenges of this thesis are concluded as follows:

- The detection-based pose estimation methods generate heatmaps which include numerous motion information. It's necessary to keep the joint motion information while processing the heatmaps into small feature maps.

- The available datasets for action recognition or repetitive counting are not suitable for 2D exercise recognition and counting multitask. A new dataset should be created to contain continuous daily exercises. Various action speeds and periods should be taken into account.

- A parallel multitask model should be designed for 2D exercise recognition and repetition counting. Powerful feature extraction backbones should be selected for both tasks.

1.3 Objectives

The objectives of our thesis are stated as follows:

1. Design a multitask system including a 2D human pose estimator and a multitask model which can recognize exercise type and count the repetitions at the same time. Suitable feature extraction backbones should be selected for the multitask model to achieve better accuracy in both tasks.

2. Utilize a suitable heatmap processing method. The processed heatmaps should have informative features of joint movement to be well fitted for both exercise recognition and counting tasks.

3. Create a new dataset that includes multi-cycle exercise videos with workout categories and counting labels. It should also take action continuity and different moving speeds into consideration.

1.4 Thesis Statement

Huge improvements have been achieved for 2D pose estimation, action recognition and repetitive counting since the wide application of machine learning. In this work, we introduce the effort of combining the three tasks together for exercise. We utilize MSPN human pose estimation model [37] as our pose estimator to produce heatmaps. Motivated by liu et al's heatmap processing methods [41], we create a new multitask model of exercise recognition and counting using processed heatmaps. In order to train the multitask model, a new dataset Rep-Penn is generated based on PennAction dataset [80]. A two-stage training strategy is applied in the training process, and we achieve excellent accuracy in both exercise recognition and counting tasks.

1.5 Contribution

Our contributions are concluded as follows:

1. 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 repetitive counting parts.

2. 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.

3. Building a new dataset called Rep-Penn based on PennAction dataset. It covers 7 exercises with 9 different cycles and 3 action speeds.

1.6 Thesis Outline

The thesis is organized as follows:

- Chapter 2 introduces the related works regarding 2D human pose estimation, 2D action recognition, repetitive counting and the multitask works at the intersection of these three fields. The basic principle of ResNet is also illustrated.
- Chapter 3 presents the multitask system and the methodology applied in this thesis.
- Chapter 4 gives a detailed explanation of the datasets we leveraged and proposed in this work. Besides, the applied evaluation metrics are listed.
- Chapter 5 presents the experiment results and comparison between related works. Also, this chapter offers a detailed analysis and discussion.
- Chapter 6 concludes this thesis and points out the future work of this work.

1.7 Scholarly Output

Our publications are presented as follows:

1. Deep Learning-Enabled Multitask System for Exercise Recognition and Counting (Accepted by Multimodal Technologies and Interaction in September 2021)

2. Digital Twin Coaching for Physical Activities: A Survey[23]

Chapter 2

Related Work

In this chapter, 2D human pose estimation, 2D action recognition, repetition counting and ResNet related works are introduced in detail. In 2D human pose estimation, only SPPE methods are introduced in Section 2.1 as this thesis focuses on single person instead of multi persons. Both detection-based methods and regression-based methods are illustrated clearly. In Section 2.2, 2D action recognition related literature is presented. Papers utilizing joint location information in action recognition are listed individually. Research related to repetition counting is presented in Section 2.3. Works regarding

action recognition & counting multitask are also provided. At the end of this chapter, ResNet and its derived networks are introduced in Section 2.4.

2.1 2D Human Pose Estimation

The purpose of 2D SPPE is to estimate the human keypoints positions. SPPE task 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 get joint coordinates indirectly, and they need post-processing like applying a maximum filter to obtain the joint locations.

Wei et al. [74] 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 input. Loss calculation is added in each stage to solve the gradient vanishing problem when training.

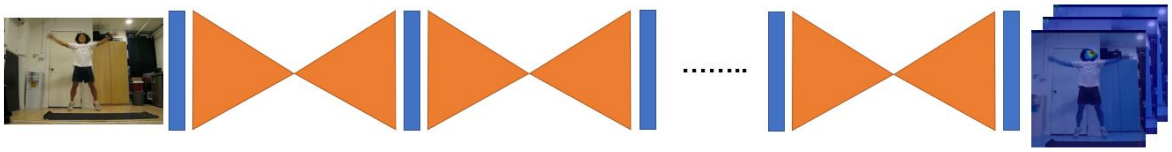

Figure 2.1: An illustration of the Stacked Hourglass Network architecture.

Stacked Hourglass(SHG) Network [47] is a significant architecture and lays the foundations for many pose estimation related tasks [5, 37]. As you can see from Figure 2.1, the inputs are RGB images and the output are heatmaps. The whole network is composed of several stages. Each stage is the same, and noted as orange blocks consisting of low-high and high-low steps. The high-low process reduces the picture resolution from high to low and the low-high operation raises the image from low resolution to high resolution. In this way, the network can learn from both small-scale and large-scale features.

The goal of the SHG network is to catch information from feature maps of all scales because small feature maps contribute to locating small body parts while large-scale fea-

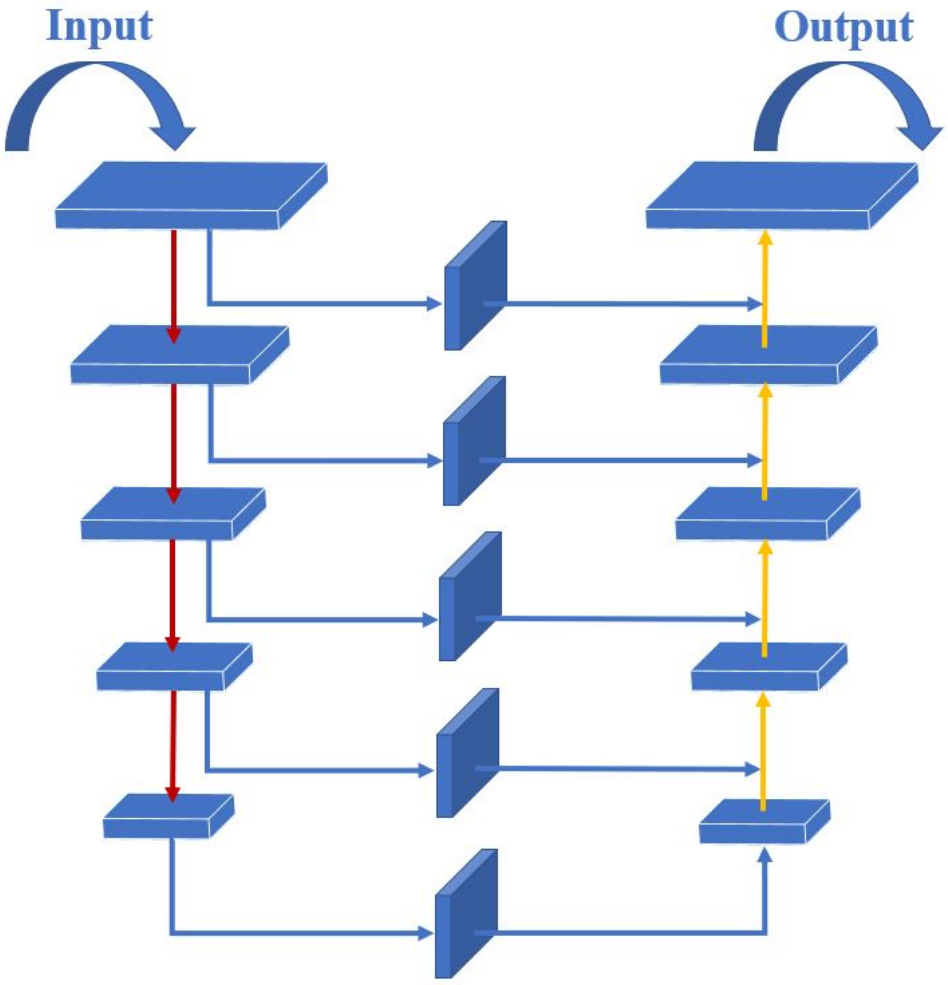

Figure 2.2: The architecture of a single-stage Stacked Hourglass Network.

ture maps help learn the full body information. Figure 2.2 illustrates a single hourglass module. It is composed of several downsampling and upsampling steps. The red arrows represent downsampling operations while the yellow ones represent upsampling operations. Each box in this network denotes a basic residual module which is displayed in Figure 2.3. After each downsampling step, a new branch is created and convoluted by a basic residual block. When the network reaches the lowest position where the resolution is the smallest, it starts upsampling and combining the generated branches when the feature size is the same. Maxpooling operations are used as the downsampling method while the nearest neighbour interpolation is applied as the upsampling method. Element-wise addition is used when features are fused.

The framework of a basic residual model is displayed in Figure 2.3. The upper path is made of 3 convolution layers including two 1×1 convolution layers and one 3×3 convolution layer. The bottom path is a shortcut. If M and N are inconsistent, a 1×1 convolution layer will be applied to make M and N uniform. The number of output channels of each convolution layer is displayed under the kernel size in each block. The basic residual module is the basic part of the Stacked Hourglass Network. The module doesn't change data size, and only depth is changed.

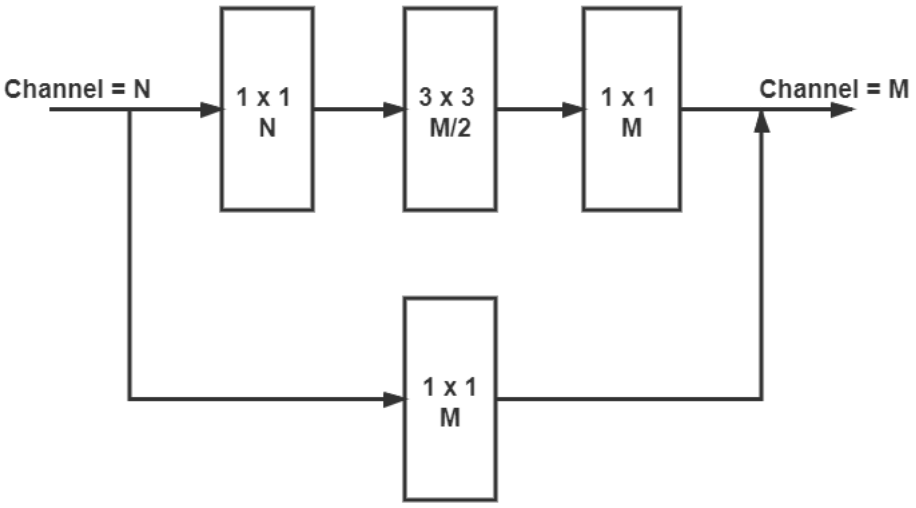

Figure 2.3: The architecture of the basic residual module.

An important technique leveraged by the SHG network is intermediate supervision. After each stage, a set of heatmaps are produced separately. The loss function is applied to the heatmaps in each stage. The later stage will refine the feature maps produced by the preceding stages as the training progresses.

Paper [78] found that the 4-stage SHG network can achieve over 95% generalization ability compared with the 8-stage SHG network. Thus, they only trained a 4-stage SHG network using Knowledge Distillation [28] which means training the smaller network under the supervision of the original network.

MSPN (Multi-stage Pose Estimation Network) [37] 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 because of its high efficiency and accuracy. A detailed illustration of MSPN is presented in Section 3.2.1.

Unlike detection-based methods which need non-differentiable processing steps on heatmaps to get the joint coordinates, regression-based methods are able to get joint locations directly. Paper [67] is among the first to apply DNN (Deep Neural Network) to pose estimation task. The Google researchers applied a cascade of pose regressors to locate the human joints with iterative refinement. However, the model has limited generalization ability.

Some researchers tried to replace the non-differentiable steps with differentiable functions so that they can build an end-to-end human pose estimation model. Luvizon et al. [46] replaced the non-differentiable argmax with differentiable soft-argmax function. It can convert the highest response from feature maps to the coordinates. In paper [48], they proposed non-trainable differentiable layers called DSNT (Differential Spatial to Numerical Transform) layers. DSNT is composed of heatmap normalization and coordinate calculation, and it achieves competitive pose estimation results. Suna et al. [65] applied integral regression in both 2D and 3D human pose estimation tasks, and achieved

excellent results especially in the 3D field.

Detection-based methods have larger feature maps compared with regression-based methods. Thus, they have strong spatial generalization abilities and achieve better accuracy. However, detection-based methods are not an end-to-end network, and regression-based methods have the advantage of generating joint locations 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 [45]. 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. Karpathy et al. [34] proposed a single-frame action recognition architecture. This method can utilize 2D CNN models which are pretrained on other datasets. However, temporal information is significant in the action recognition task. Some 2D CNN based papers added a time-distributed layer to get temporal information in the videos like paper [44].

To make use of temporal information, researchers found many solutions. 3D CNN [31] is an intuitive way to acquire temporal information from videos. Tran et al. [70] introduced an advanced version of 3D CNN called C3D. They found a better kernel size for 3D CNN and proved their architecture has better performance. Moreover, researchers improved C3D and proposed Residual CNN which is two times faster and smaller with comparative performance [69]. In paper [75], they built an asymmetric 3D convolution depth model which further improves efficiency and effectiveness.

Some works also focus on utilizing pretrained 2D CNN based models while taking temporal information into consideration [9, 29]. Paper [9] introduced a new 3D CNN architecture called I3D (Two-Stream Inflated 3D CNN) which is based on 2D CNN inflation. Their method has the advantage of using the parameters from 2D CNNs

trained on ImageNet [7] by concatenating inflated filters and kernels from existing 2D models. Paper [29] converted pretrained filters of 2D CNN to a 3D structure without prepossessing, and their parallel 3D CNN architecture remains competitive.

RNN (Recurrent Neural Network) and LSTM (Long Short-Term Memory) are also widely used in action recognition tasks [56, 21]. RNN is a recurrent network by performing the same function on each input data. The output of the current stage not only depends on the present input data, but also on the previous stage. LSTM is a better version of RNN which makes it easier to remember past data.

Another way for action recognition is using multi-streams [6]. These methods usually apply several CNNs to make the best use of appearance and motion information. One representative article is [61]. Its model includes two streams: Spatial stream CNN and Temporal Stream CNN. The spatial stream focuses on getting static information from still frames while the Temporal Stream CNN obtains motion information by optical flow images. One of the main problems is that this network lacks information change between these two streams. Paper [22] worked on this limitation and built a bridge that allows information transfer in the two features.

Despite obtaining the action type directly, many researchers work on action recognition tasks utilizing pose estimation results. For regression-based human pose estimation methods, the joints positions are computed directly. Cheron et al. [11] leveraged informative areas around the human joints of the image. Both RGB and optical flow of these parts are fed into the CNN model to get the action results. Paper [45] utilized the 2D joints coordinates to create a 3D image-like matrix that represents the temporal movement of all the joints. Some researchers [43] also decoded the skeleton information using different colours according to the type of joints. An Encoded Human Pose Image (EHPI) is generated according to the joint type and frame number. Sun et al. [64] fed the joints keypoints to an embedding model, and their model is effective in view-invariant action recognition.

To make use of the rich information of the heatmap which is a byproduct of detection-

based pose estimation methods, lots of literature created effective methods in leveraging heatmaps. In the paper [13], they colorized the heatmaps based on the order of the frames. These maps are temporally aggregated and fed into a classification network. Shah et al. [59] improved this method by reweighing motion information of various joints. Segu et al. [58] applied this idea to the 3D pose estimation and action recognition field. Liu et al. [40] 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 get the classification result. These works got good results but they all need complicated processing on the heatmaps like colourization. In paper [41], researchers made use of two features derived from pose estimation maps: DPI and DTI. This method is straightforward and intuitive without complex processing procedures. What's more, DTI alone contains both joint movement information and body shape information which we found can help other tasks like human exercise counting.

2.3 Repetitive Counting

Repetitive Counting is an important task in the Computer Vision field. Many works on repetitive counting commonly transform the motion to a one-dimensional signal. Frequency information is extracted by signal processing methods like Fourier Transform [20, 52, 15, 39, 4, 50, 3], peak detection [66, 63] or singular value decomposition [12, 51]. These methods assume the motion is periodic and stationary, which is unsuitable in many non-stationary situations. Therefore, Paper [53] 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 context information into consideration.

The periodic detection problem can also be treated as a problem of finding commonalities between two videos sequences [15, 49]. In the paper [14], the researchers

represented the video sequences as histogram-form features, and leveraged a Branch and Bound (R&B) algorithm to find the common events in two videos. Shariat et al. [60] proposed an advanced segmental alignment model which can find the segmental boundaries of common events and pair them automatically. Their methods have better noise immunity when matching sequences. In [49], the researchers 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 situations when the semantic content of the videos, the number of periods and the valid duration of the video are unknown. Unlike the former methods, Debidatta et al. [18] introduced a new symmetric matrix. It acts as an intermediate layer to predict the cycle length and valid periodic length. Using this method, it achieved up-to-date accuracy in benchmark QUVA [53].

There are also many other methods that don't fall into the two divisions above. Levy and Wolf [36] employed a CNN model for the whole video to estimate the cycle length. After that, 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. Topology based method is also an important branch [71, 68]. The ideas are based on approaches from applied topology. They leveraged cohomology-based methods to recover the location of object movements by generating a symbolic motion cycle. The created closed curves describe a repeated movement for recurrent animation.

Some researchers also learn from action-recognition tasks [77]. They extracted deep features from BN-Inception Network [30], and transformed the high-dimensional features to a one-dimensional wavelet using PCA (Principal Component Analysis). Signal processing methods are applied to the extracted wavelet to get the counting result.

Due to the close relationship between action recognition and counting as both tasks focus on spatial difference throughout the video, many works combine these two tasks together. In the paper [10], they leveraged data from accelerators, and applied classical

methods such as naive Bayes classifier, hidden Markov models and peak detection. They achieved over 90 percent accuracy in exercise recognition and the miscount rate is about 5 percent. An inertial measurement unit is also used along with the accelerator [57]. They achieved 92% action recognition accuracy and 2.42% miscounting results in 16 gym exercises. Smartphones and smartwatchers are also widely used in this task, and they obtained better results due to the improvements of algorithms and built-in sensors [24, 63]. Though these methods achieved good accuracy, they cannot be widely applied due to hardware limitations.

Wifi-signals are also used in this multitask. Xiwen et al. [42] proposed a CSI (Channel State Information) based method. Peak-finding algorithms are applied to count repetitive actions, and action type is recognized using KNN (K-Nearest Neighbours) classifier. Even though the wifi-based method is free of wearing devices, submitting and receiving devices are still necessary.

Finally, to the best of our knowledge, paper [1, 35] are the most related works as my thesis, which combine three tasks together. Alatiah et al. [1] generated a new dataset by augmenting UCF101 dataset [62]. OpenPose [7] was applied to estimate 3D human pose. The estimated keypoints are fed 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. The counting result 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. [35] collected exercise videos from gym cameras. They obtained keypoints utilizing Wang et al.'s [72] method. The keypoints are gathered to form motion trajectories. Different from this thesis, their method works in the multiple-people situation. Therefore, the trajectories are processed and clustered. The processed features are fed into a multilayer classifier and a multilayer regressor to achieve exercise category and counting results respectively. This paper has the advantage of working for multiple

persons. However, its accuracy is limited and needs further improvement.

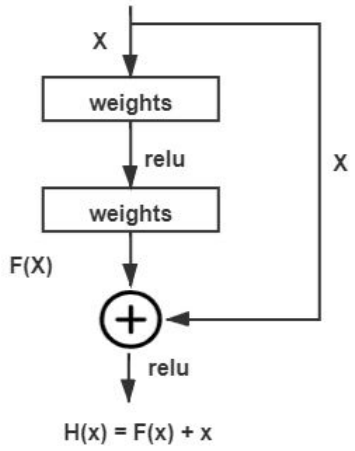

Figure 2.4: An illustration of the residual learning block

2.4 Resnet

ResNet [26] was proposed by Kaiming He in 2015. It won first place in ImageNet [17] classification task. Because of its simplicity and practicality, many target detection and image classification problems are completed on the basis of ResNet, which has become an important cornerstone structure in the field of computer vision.

The ResNet is introduced to solve the gradients vanishing or exploding problems when the network becomes deeper. The researchers found that using residual learning block (Figure 2.4) can help alleviate this issue. Suppose the input of the original network is x and the output is $H(x)$. Let $H(x) = F(x) + x$. Therefore, the network only needs to learn $F(x)$ which is a much easier task than learn $H(x)$. What's more, the deeper model doesn't get larger errors than the shallow model.

There are five commonly-used ResNet networks: ResNet18, ResNet34, ResNet50, ResNet101 and ResNet151. The five models are all composed of three parts: input layers, convolution blocks (including four stages) and output layers. They share the same input layers including one 7×7 convolution layer and one 3×3 maxpooling layer.

The output layers are the same which is made of the average pooling layer and FCN (Fully Connected Network). The only difference is the convolution blocks. The stages of ResNet18 and ResNet34 apply the basic block shown in Figure 2.5 (a). It is composed of two 3×3 convolution layers. Both the feature size and the number of channels is not changed inside the block. For the ResNet50, ResNet101 and ResNet151, the basic block is changed to Figure 2.5 (b). It is made of two 1×1 convolution layers and one 3×3 convolution layer. Compared to Figure 2.5 (a), it reduces the computation and introduce more non-linear mapping.

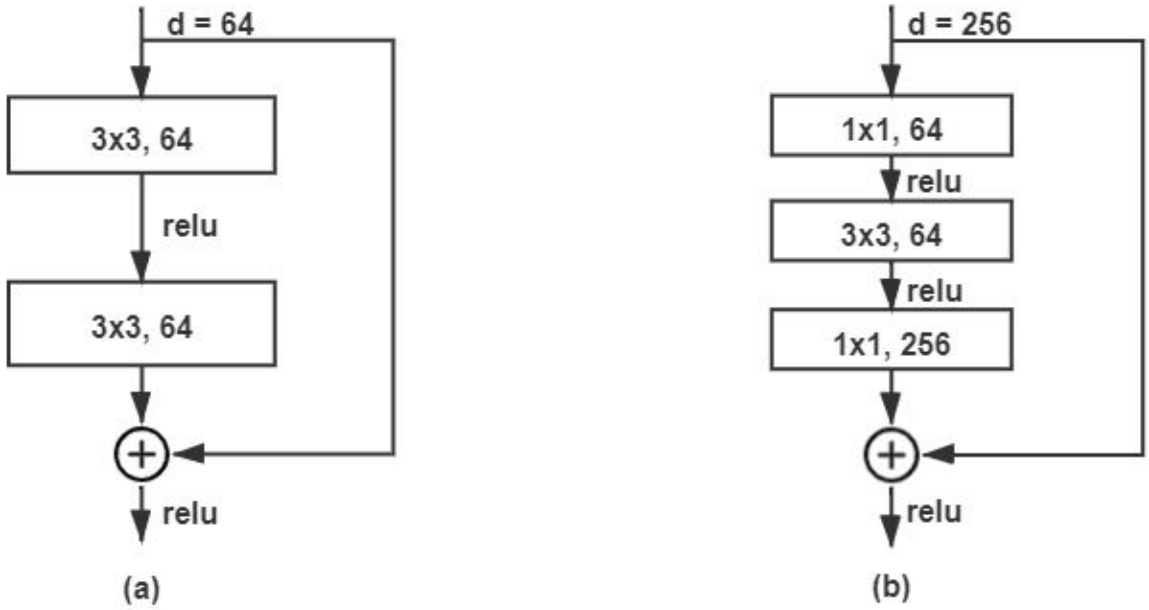

Figure 2.5: An example of basic building blocks in ResNet.

An example of the original residual unit and the updated residual unit architecture is shown in Figure 2.6. X_L represents the output of the previous residual Unit. X_{L+1} denotes the output of the present residual unit. Conv is short for the convolution layer. BN represents the Batch-normalization layer and ReLU means the ReLU activation layer.

In the original residual unit Figure 2.6 (a), a Relu activation function is added after shortcut, which restricts the performance of the model. In the [27], ResNet-V2 is introduced in Figure 2.6 (b). Compared with the old version, it places the ReLU layer inside the residual block which improves its anti-gradient-vanishing abilities. Both our action

recognition model and counting model are built based on ResNet Unit V2. Details are discussed in Section 3.4.

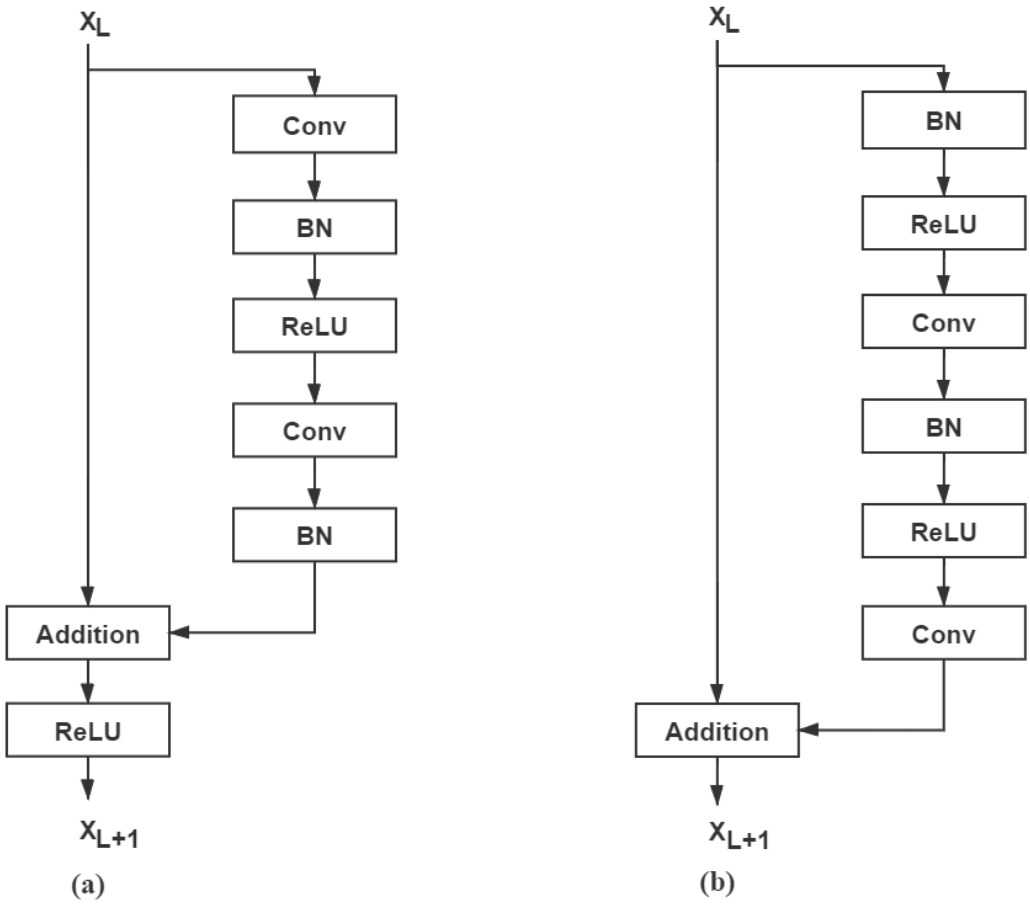

Figure 2.6: The architecture of the original residual unit (a) and residual unit V2 (b).

2.5 Summary

As introduced above, many works focus on the interaction field of the 2D human pose estimation, 2D action recognition and repetitive counting, but few combine all of them together. The papers [1, 35] that cover the three fields applied keypoints positions calculated by a pose estimator to the other two tasks. Their methods have some drawbacks like preselected parameters and low accuracy in exercise recognition and counting.

Inspired by [41], we find that the byproduct of detection-based pose estimators contains rich human motion and body shape information. The heatmaps can not only be used in 2D action recognition tasks, but also exercise counting tasks because of their intuitive movement features. What's more, ResNet models' strong feature extraction abilities can be utilized as the key part of our multitask model. In the next chapter, the system overview, details of the pose estimation, heatmap processing methods, the multitask model architecture and the training strategy of the multitask model will be illustrated clearly.

Chapter 3

Methodology

In this chapter, we explain the proposed multitask system containing an off-the-shelf 2D human pose estimator MSPN and an exercise recognition & counting multitask model. This chapter is organized as follows. A general outlook of the multitask system is introduced in Section 3.1. MSPN architecture, training strategies of MSPN and calculating joint coordinates steps are illustrated in Section 3.2. Section 3.3 presents the detailed heatmap processing methods. Section 3.4 illustrates the proposed multitask model of exercise recognition and repetition counting. A two-stage training strategy applied in

the multitask model training process is introduced in Section 3.5.

3.1 System Overview

An illustration of the proposed system is provided in Figure 3.1. 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 left branch, max activating locations of the heatmaps are calculated to get the joint positions of the human body. The keypoints are connected together to form a body skeleton. On the right 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.

3.2 Pose Estimation

As mentioned in Section 1.1, heatmaps are the byproducts of human pose estimation models. Thus, we introduce the architecture of the human pose estimator MSPN, training strategies utilized in MSPN and the details of predicting joints locations from heatmaps in this session. The details of the MSPN model are introduced in Section 3.2.1. Fine-tuning is presented in Section 3.2.2.1, and Coarse-to-fine and OHKM (Online Hard Keypoints Mining) training strategies are illustrated in Section 3.2.2.2. The processing steps from heatmaps to joint coordinates are presented in Section 3.2.3.

3.2.1 MSPN Model

The MSPN model is a 2D human pose estimation model with up-to-date accuracy and competitive speed. Figure 3.2 shows the architecture of a two-stage MSPN. The in-

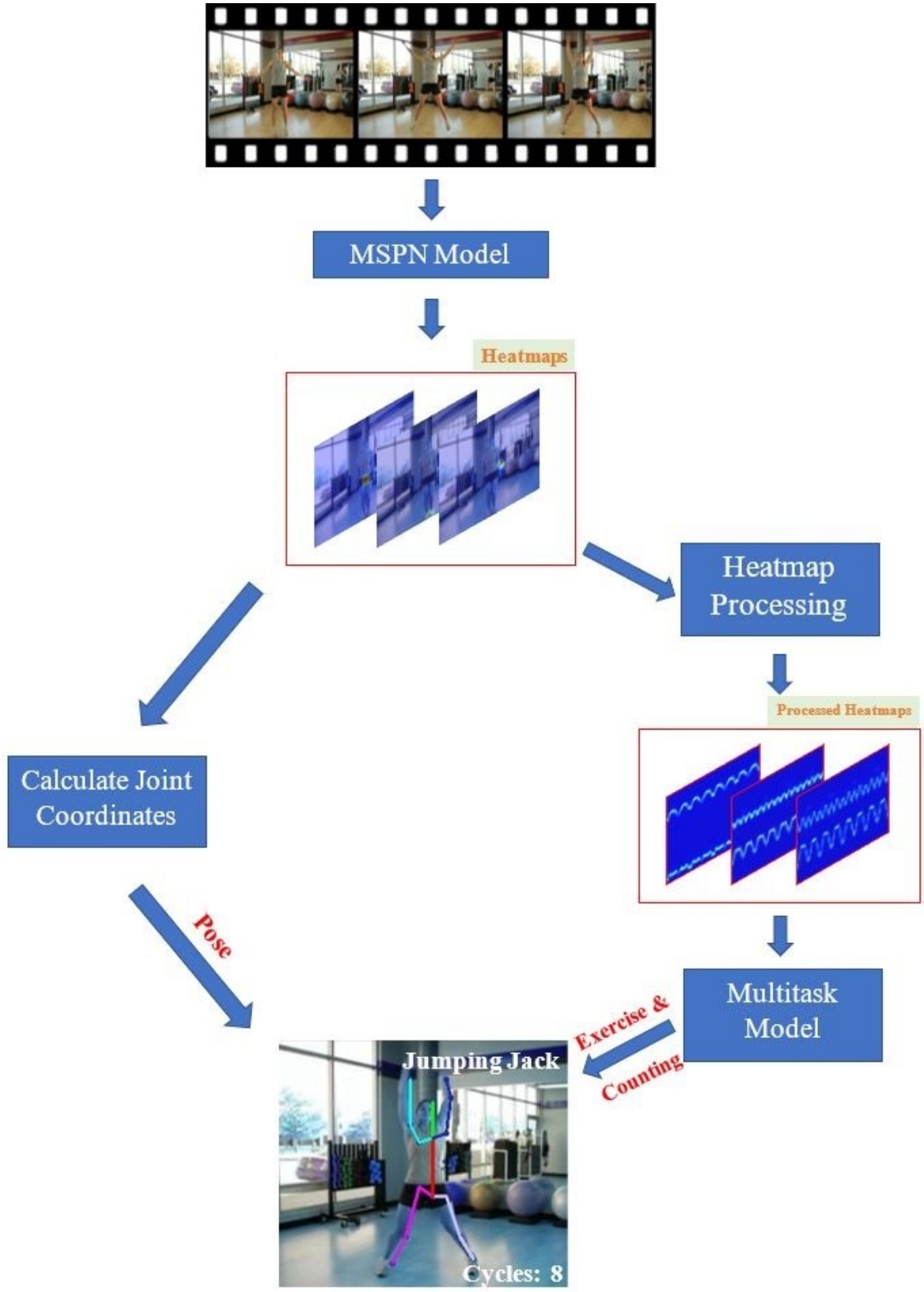

Figure 3.1: An illustration of the whole multitask system.

puts of MSPN are RGB images and outputs are heatmaps. The whole network includes two stages represented by the grey shades. A single stage is composed of four downsampling units and four upsampling units. The red arrows represent the downsampling data flow while the yellow arrows denote the upsampling data flow. The output of the downsampling unit is both the input of the next downsampling unit (if it is not the last downsampling unit in one stage) and the upsampling unit on the same level in the same stage. Each stage has four outputs which are denoted by the black arrows. The cross-stage aggregation technique is applied in the MSPN. As you can see from the green lines, it connects two units from the previous stage to the downsampling unit in the current stage. This technique helps share features between different stages.

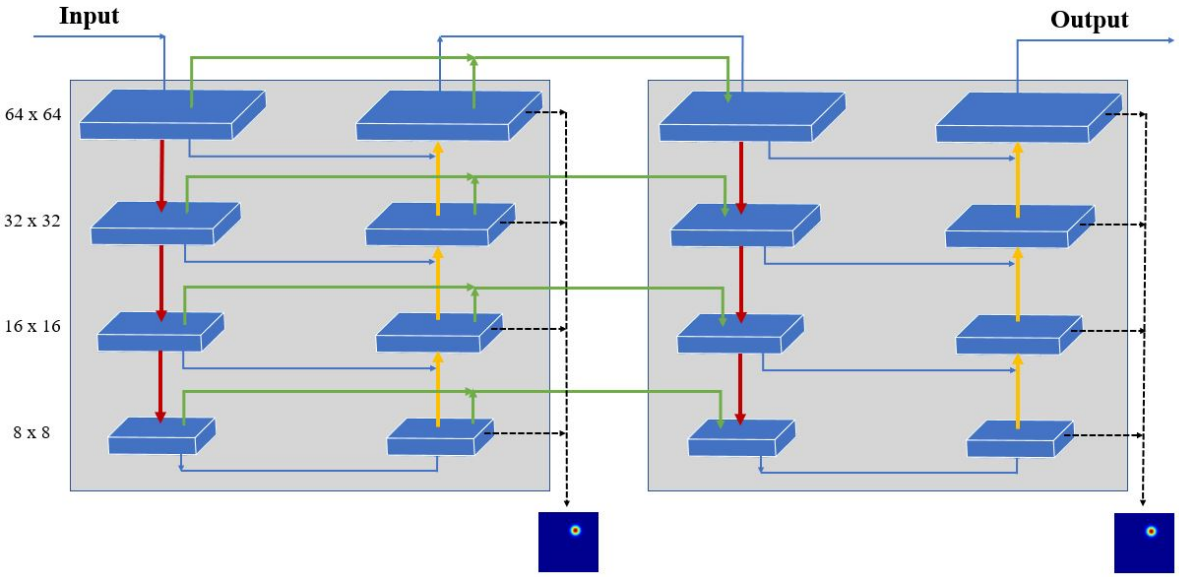

Figure 3.2: A view of a two-stage MSPN architecture.

3.2.2 Training strategies

3.2.2.1 Fine-tuning

Because of the high accuracy and stability of MSPN, We applied it as our human pose estimation model. MPII dataset [2] is a widely used dataset which is tailored for 2D human pose estimation. It contains plenty of human pose images with 16 body keypoints

but it doesn't cover the exercises videos we need. PennAction dataset [80] includes exercises with both actions and pose joints labels, but it only provides 2K video samples, which is not enough to train a complex human pose estimation model like MSPN. What's more, only 13 joints are labelled without keypoints like 'thorax', 'pelvis' and 'neck'. To make use of both datasets, we pretrained MSPN on MPII dataset first. After that, MSPN is fine-tuned by being trained again on PennAction dataset based on pretrained weights. The missing joints in PennAction datasets will be treated as 0s. It won't affect the training results as these joints are not counted in the total loss. Therefore, The model's generalization ability in PennAction dataset is improved compared with using pretrained weights. It also outputs 16 joints' heatmaps which help describe the human body motion better.

3.2.2.2 Coarse-to-fine and OHKM

To ensure MSPN's good performance, two important training strategies are introduced as follows. The coarse-to-fine strategy is an important technique in the MSPN training. Different sizes of groundtruth heatmaps are applied to different stages. The latter the stage is, the smaller the groundtruth heatmap is. The reason is that the estimated heatmaps are refined gradually in multi-stages, and the model can learn more and more accurate joint positions at a later stage. Using this technique is able to improve the localization accuracy. Another technique is OHKM. In the last supervision block of the last stage, only K ($K = 8$ in the thesis) largest losses among all the joints' losses are added together. The joints with small losses are excluded in the loss calculation. OHKM can help the pose estimation model focus on these few joints with the largest errors, and improve the model's prediction accuracy further.

The parameters of the training process are introduced in Section 5.2.1 and L2 loss is applied as the loss function:

$$L2 = \sum_{n=1}^{N} (y_n - \hat{y_n})^2 \tag{3.1}$$

In this formula, N represents the number of samples. y_n and $\hat{y_n}$ denote the true joint coordinates and predicted joint coordinates respectively.

3.2.3 Calculate joint coordinates

To estimate the joint locations from heatmaps, extra processing methods should be applied. In this thesis, the heatmaps are filtered by a Gaussian filter with Standard Deviation (σ) equal 0.5. Then, the heatmaps are 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

In this section, heatmap processing procedures are introduced in detail. 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 a human pose estimator to provide heatmaps based on Rep-Penn dataset. It produces K ($K = 16$ in our case) 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

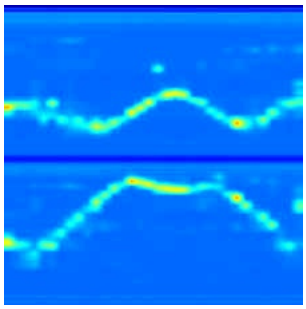

Figure 3.3: An example of a single-cycle processed heatmap.

respectively, and obtain 3D features $H \in R^{W \times F \times K}$ and $W \in R^{H \times F \times K}$:

$$H[w, f, k] = \frac{1}{H} \sum_{h=1}^{H} J[h, w, f, k] \tag{3.2}$$

$$W[h, f, k] = \frac{1}{W} \sum_{w=1}^{W} J[h, w, f, k] \tag{3.3}$$

In this way, J is projected horizontally and vertically. Two 3D features H and W represent the ordinate movement and the abscissa movement respectively. We add H and W 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:

$$N = 255 \times \frac{P - min(P)}{max(P) - min(P)} \tag{3.4}$$

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 3.3. This picture shows how the joint moves in a single cycle. The upper part is the ordinate change curve, and the below part is the abscissa change curve. Our generated dataset includes exercises with multiple periods. An illustration of processed heatmaps calculated from a

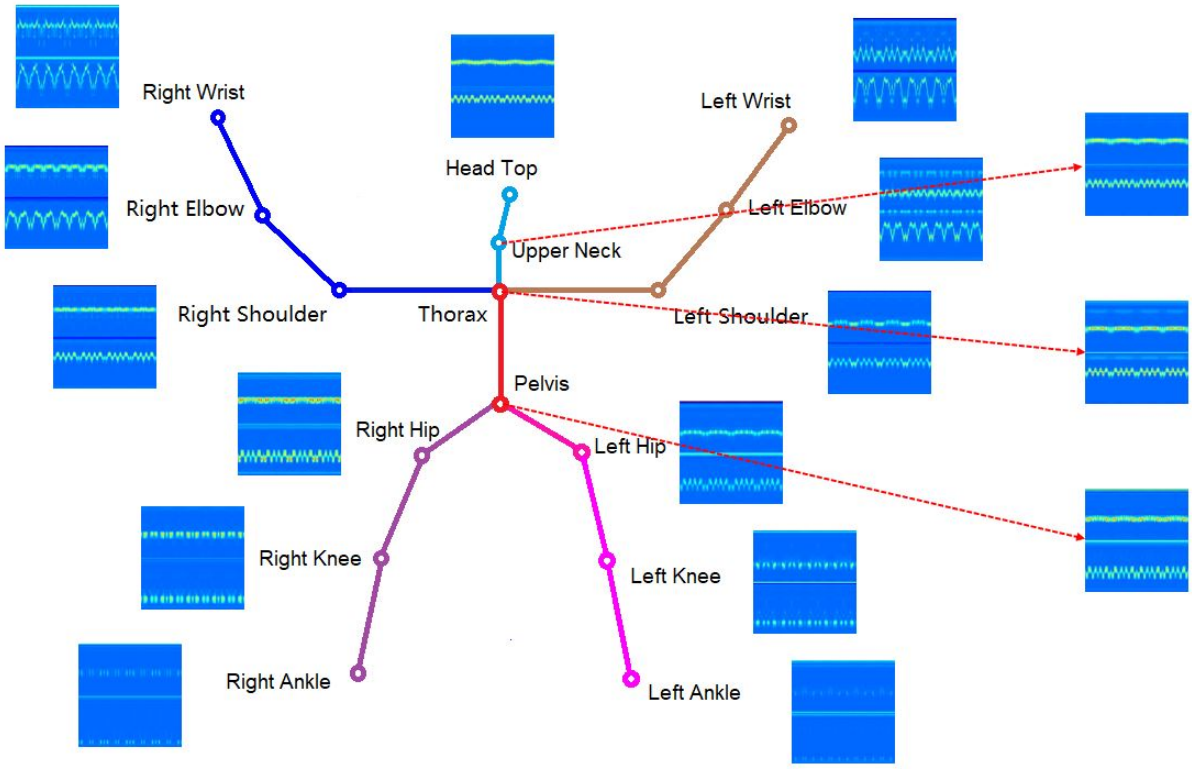

Figure 3.4: An example of processed heatmaps of a multi-period video.

multi-cycle exercise video is displayed in Figure 3.4. This 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 thesis, 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.

We apply the ResNet34 network as the backbone of the exercise recognition part because of its simplicity and effectiveness. The entire ResNet34 network is shown in Figure 3.5. The whole network is mainly composed of four stages with 3, 4, 6 and 3 identical blocks respectively. The input size is 224×224×16, and the output size is a one-dimensional vector with length 512. Exercise recognition is considered a classification

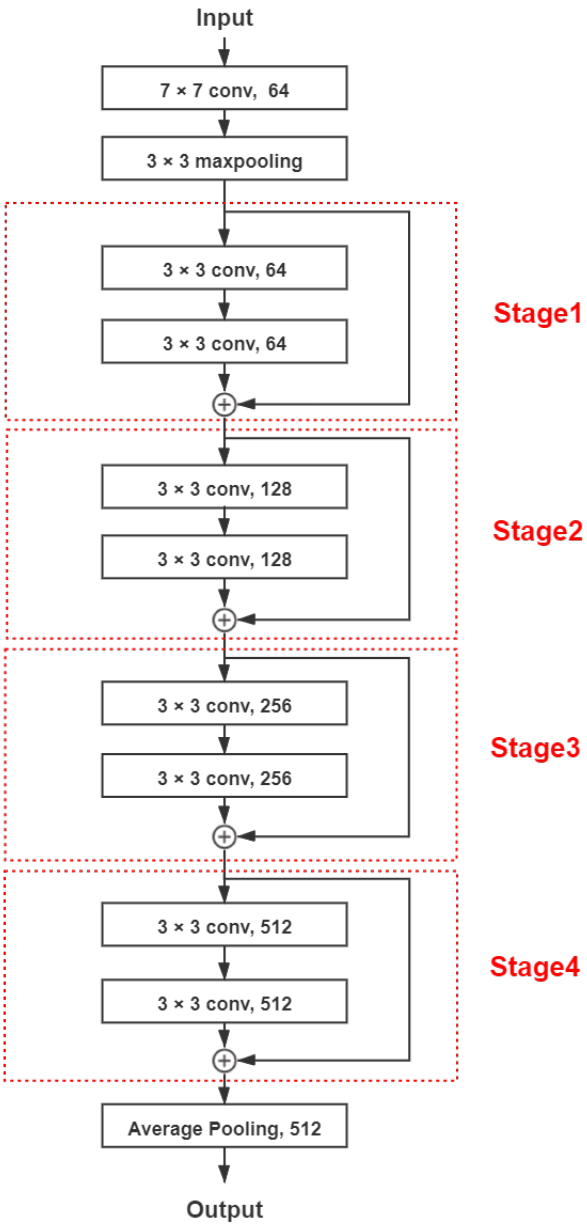

Figure 3.5: An illustration of the architecture of ResNet18 and ResNet34.

problem. To match the output size of ResNet34 and the number of classes of Rep-Penn which is 7, a Fully Connected Network (FCN) made of dense layers is added at the end of ResNet34 as displayed in Figure 3.6. The action recognition branch consists of the ResNet34 network and a Fully Connected Network (FCN). The inputs are features from Rep-Penn. The output is a vector of length 7 representing the probability of each action. The dense layer is a commonly used neural network layer. Each neuron of the dense layer receives input from all the neurons of the previous layer. The FCN consists of 3 dense layers with output sizes 128, 28 and 7. After each dense layer, the Relu activation function is added except the last layer. The final layer is followed by a Softmax activation function which is widely applied to normalize network output to a probability distribution over predicted output classes.

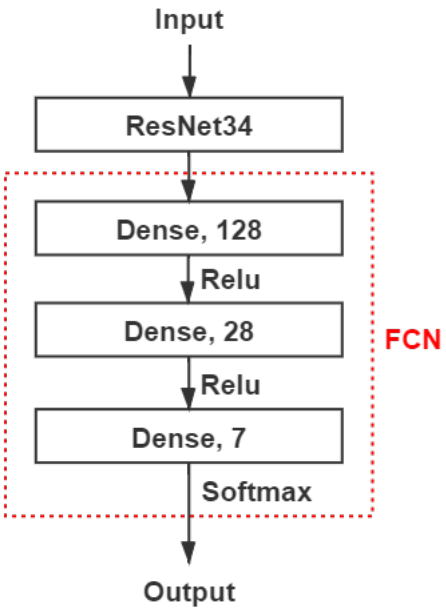

Figure 3.6: An illustration of the exercise recognition branch.

The repetitive counting task is based on ResNet18 which shares a similar architecture as illustrated in Figure 3.5. Contrary to ResNet34, every stage in ResNet18 owns 2 identical blocks. In this thesis, repetition counting is treated as a regression problem. Therefore, the output of the counting layers should be a single value. Figure 3.7 displays

the architecture of the counting branch. The input is the same as the exercise recognition task with the size of 224×224×16. It outputs the predicted counting results directly. The FCN in the counting network includes 4 dense layers, with neuron size 128, 28, 7 and 1. Similar to the recognition task, the 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.

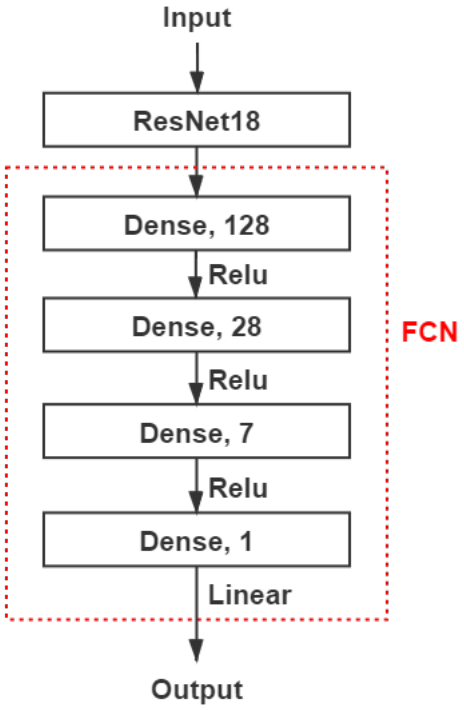

Figure 3.7: An illustration of the counting branch.

.

The proposed multitask system network is introduced in Figure 3.8 by combing the exercise recognition and counting networks together. Yellow blocks and green blocks are exercise recognition and counting branches respectively. Besides the two branches, several shared convolution layers are added to provide global features for these two tasks. The shared layers include two 3x3 convolution layers and two 5x5 convolution layers. All the layers have the same output size as the input. Shared convolution layers are represented

by orange blocks.

3.5 Network Training

In this section, the details of training the multitask model are introduced. As mentioned in Section 3.4, the exercise recognition task is based on ResNet34 while the repetition counting task is based on ResNet18. They share the same input data from Rep-Penn, and the input sizes are both 224×224×16. In this case, we are able to train a multitask model which can recognize and count repetitive exercises at the same time.

To achieve this goal, a two-stage training strategy is applied. At the first stage, we feed the multitask model with Rep-Penn dataset and action labels. Both recognition network and shared convolution layers are trained, and the counting network is frozen. The trained layers are marked by orange and yellow blocks in Figure 3.8. We consider the exercise recognition task as a classification problem and Categorical Cross-Entropy Loss is applied:

$$CE = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{C} y_{ni} log(p_{ni}) \tag{3.5}$$

where N is the number of samples, and C is the number of exercise class. $y_n = [y_{n1}, y_{n2}, ..., y_{nc}]$ 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 (orange and yellow blocks in Figure 3.8) are frozen, and only the counting network (green blocks in Figure 3.8) is trained at this stage. The loss function is MSE (Mean Squared Error):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.6}$$

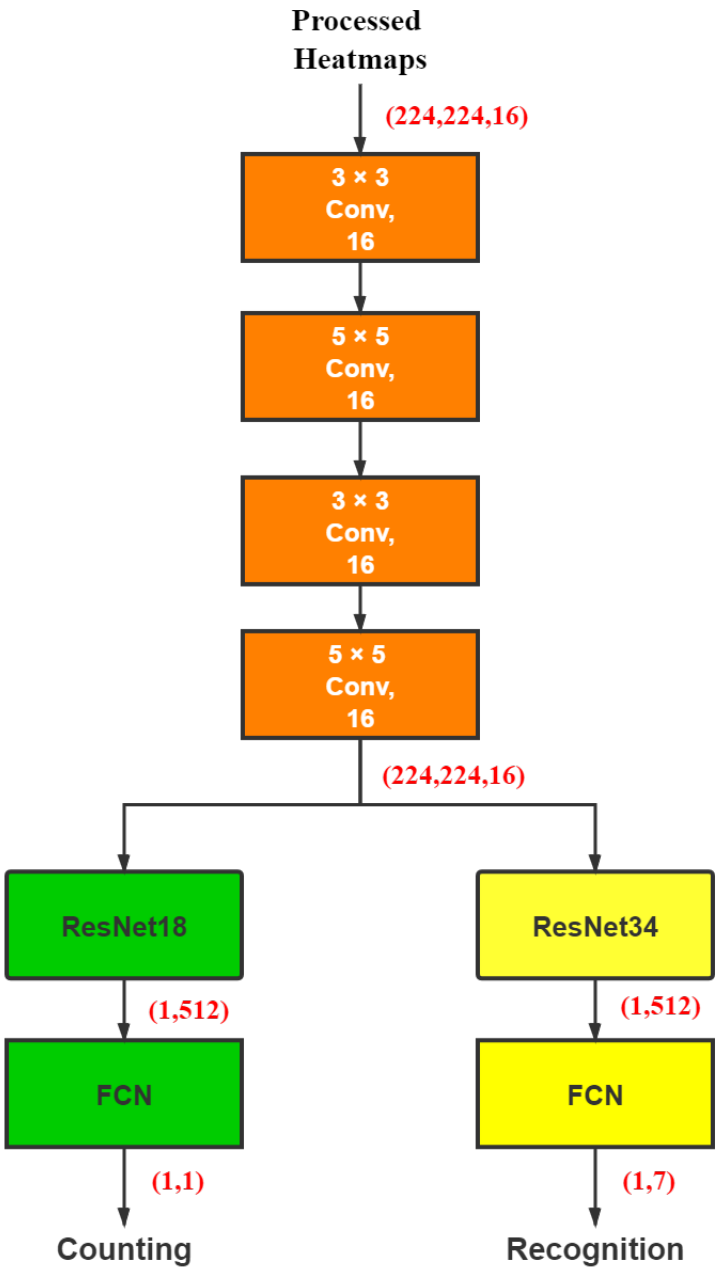

Figure 3.8: Proposed multitask model for exercise recognition and counting.

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.

3.6 Summary

In this chapter, we introduce the proposed multitask system and the methodology applied to it. Fine-tuning, Coarse-to-fine and OHKM help train an advanced MSPN, which produces accurate heatmaps. Processing steps for heatmaps extract rich motion features for the exercise recognition & counting multitask model. The architecture of the multitask model is presented including mutual layers and two branches for each task. Finally, a two-stage training strategy is utilized in training the multitask model, allowing the multitask model to estimate exercise type and count repeated actions simultaneously. The datasets and the evaluation metrics we apply will be introduced in the next chapter.

Chapter 4

Data and Evaluation Metric

In this chapter, we introduce two commonly used 2D human pose estimation datasets MPII [2] and PennAction [80] and our generated dataset Rep-Penn. Evaluation metrics for exercise recognition and counting tasks are also illustrated. Many 2D human pose estimation datasets are available such as MPII [2], PennAction [80], LSP [32], FLIC [55] and COCO [38]. Among these datasets, both MPII and PennAction contain rich data for 2D human pose estimation. What's more, PennAction includes several exercises that we need. Rep-Penn dataset is generated based on PennAction dataset. The details of

MPII and PennAction and the process of creating Rep-Penn dataset are introduced in Section 4.1. Evaluation metrics are presented in Section 4.2.

4.1 Datasets

4.1.1 MPII

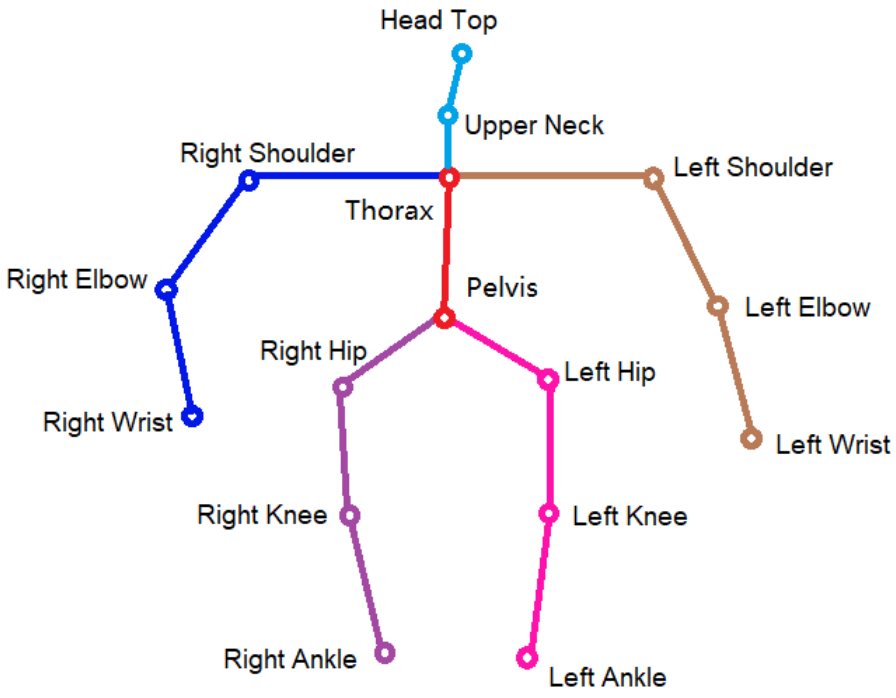

Figure 4.1: A view of joints of MPII dataset.

MPII [2] is a 2D human pose dataset established by Mykhaylo el at. in 2014. This dataset is composed of about 25K real-world activity images including 40K people. Among the images, 15K images are for training purposes, 3K images are validation samples and 7K are used for testing. The images are collected from Youtube. Each image is manually labelled with 16 joints locations. The overview of the joints is displayed in Figure 4.1. There are 16 joints in total including head, neck, thorax, pelvis, shoulder, elbow, wrist, hip, knee and ankle.

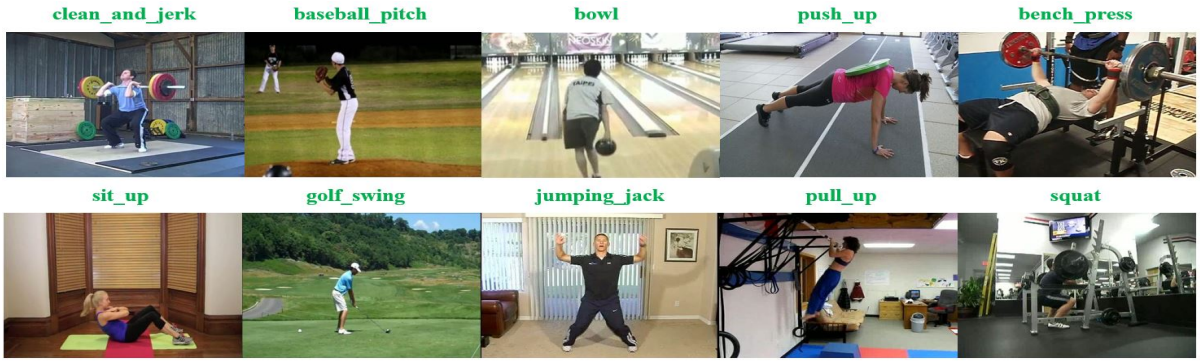

Figure 4.2: An example of actions in PennAction dataset

4.1.2 PennAction

Weiyu Zhang et al. [80] published PennAction dataset in 2013. It includes 2326 video sequences of different actions which come from diverse online video repositories, such as YouTube. There are 15 action classes in total (Figure 4.2): baseball pitch, baseball swing, bench press, bowling, clean and jerk, golf swing, jump rope, jumping jacks, pull up, push up, sit up, squat, strum guitar, tennis forehand and tennis serve.

Each video is labelled with action type, bounding box size and coarse viewpoint, and manually separated into several frames. All the frames are RGB images, and the resolution size is within 640×480. Thirteen body joints are labelled for each frame, which is shown in Figure 4.3. They are head, shoulder, elbow, hand, hip, knee and ankle. Each joint is accompanied by a visibility label which denotes if the joint is visible or not.

4.1.3 Rep-Penn

In order to meet the needs of 2D exercise recognition & repetition counting multitask, We create Rep-Penn dataset based upon PennAction dataset. PennAction dataset only includes exercise videos with one cycle whilst multi-cycle physical activities are required in this thesis. Therefore, we synthesize the Rep-Penn dataset by using the single-cycle exercise video frames. Instead of connecting the end of the video repeatedly, we connect the forward video with the rewinding video. In this way, continuity of the body movement

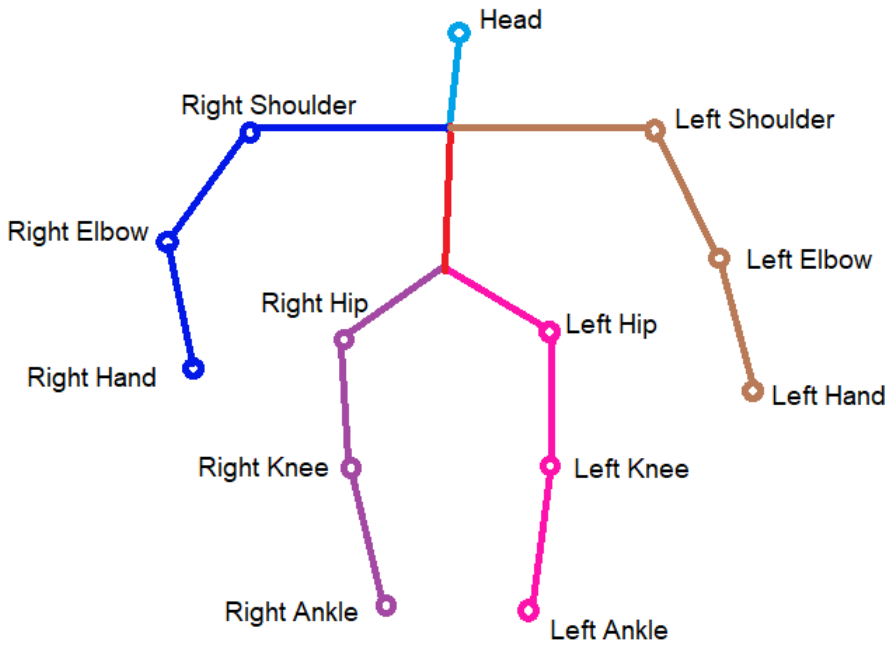

Figure 4.3: A view of joints of PennAction dataset.

is guaranteed, and exercise of various periods can be generated for repetition counting. Figure 4.4 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 repetition of a single-cycle video.

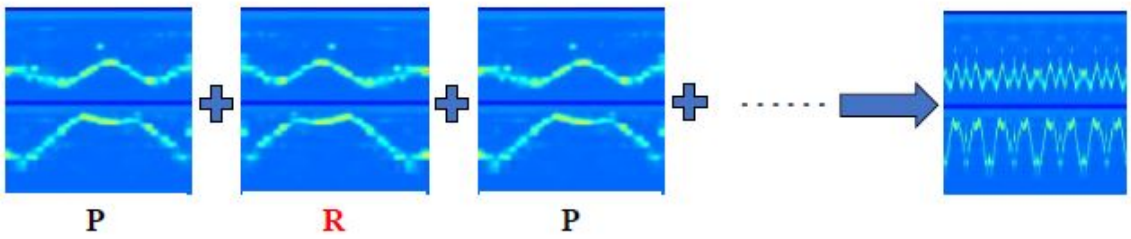

Figure 4.4: An illustration of creating a multi-cycle exercise video.

In this thesis, we produce exercise videos with $6 \sim 14$ cycles considering the normal quantity of repetitions in a set of exercises. Besides different periods, we take motion

speeds into consideration. We sample all the frames by taking 1 frame every C ($C = 1$, 2, 3 in the thesis) frames. As a result, three different action speeds are added to increase the diversity of the synthesized dataset. Figure 4.5 utilizes the heatmaps to illustrate how the number of cycles and speed differ in various workout videos. This image shows the motion of the right wrist in two exercises: bench press and sit up. We can see from this image that different exercises have distinct joint motion patterns. The difference in moving speed is not easily distinguished due to the heatmap size limitation. The change in the action period can be perceived effortlessly.

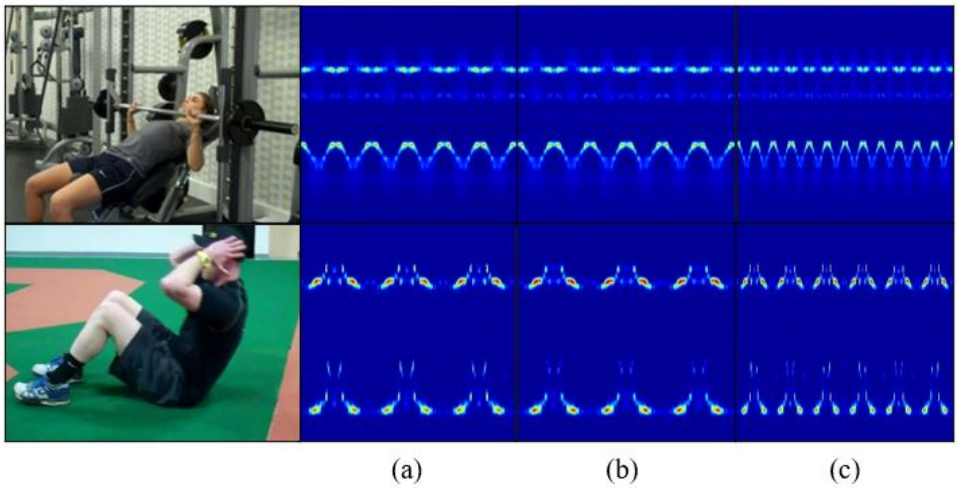

Figure 4.5: The motion of right wrist in bench press and push up. (a) 6 cycles and low speed. (b) 6 cycles and fast speed. (c) 12 cycles with middle speed.

Therefore, each exercise video in PennAction is enhanced to 27 videos with the same exercise, different cycles and various speeds. Among the 15 actions in 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 29187 exercise videos. We separate the training dataset and test dataset at a ratio of 4:1, with 23220 videos for training and 5967 videos for testing.

4.2 Evaluation Metrics

In this section, we introduce four metrics for the exercise counting task 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 predicted counting result P, and then divided by the ground truth G : $\frac{|G-P|}{G}$. MAE is the average of the normalized absolute difference in the whole dataset:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\frac{|G_i - P_i|}{G_i} \tag{4.1}$$

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.

$$OBO = \frac{Num[(G-P) <= 1]}{Num(G)} \tag{4.2}$$

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 to MAE but it isn't divided by groundtruth value G:

$$AE = \frac{1}{N}\sum_{i=1}^{N}|G_i - P_i| \tag{4.3}$$

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

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(G_i - P_i)^2} \tag{4.4}$$

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

$$Accuracy_{exe} = \frac{C}{N} \tag{4.5}$$

4.3 Summary

In this chapter, three datasets including MPII, PennAction and Rep-Penn and evaluation metrics are introduced. MPII and PennAction contain large numbers of RGB images with accurate pose labels. The generated dataset Rep-Penn is synthesized based on PennAction by repeating single-period videos. It covers 7 different exercises with 9 various periods and 3 particular action speeds. What's more, the evaluation metrics for both exercise recognition and counting are introduced. The main evaluation metrics of the counting task are MAE and OBO. MAE suggests the prediction error, and OBO indicates the ratio of samples whose predicted value is within ± 1 of the groundtruth. The experiment results and the comparison between related works are provided in the next chapter.

Chapter 5

Experiments

In this chapter, we introduce the data preparation, the experiment results and the comparison between other literature. The data preparation for both the human pose estimation model MSPN and the exercise recognition & counting multitask model are introduced in Section 5.1. Experiment environment and training parameters are presented in detail in Section 5.2.1. We evaluate our method on generated Rep-Penn dataset. The results of exercise recognition and counting tasks are listed and analyzed separately in Section 5.2.2. Related works regarding the two tasks are compared with this thesis in

Section 5.3. It proves that this thesis achieves excellent accuracy in both tasks. The limitation is discussed in Section 5.4.

5.1 Data preparation

To derive the heatmaps from the pose estimation model MSPN, we reproduce MSPN according to his paper [37]. Before randomly feeding the data from MPII or PennAction dataset into the MSPN model, each image is augmented by cropping, flipping, rotating and scaling operations randomly. The rotation range is -30 $\sim$ 30, and the scale range is 0.8 $\sim$ 1.2. The images are cropped to the size of 256 $\times$ 256.

The size of the output of MSPN is 64 $\times$ 64. After being processed by the method mentioned in Section 3.3, heatmaps are resized to 224 $\times$ 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 5.1. The 3D feature maps are considered as basic samples, and they are shuffled before being fed into the recognition & counting multitask model.

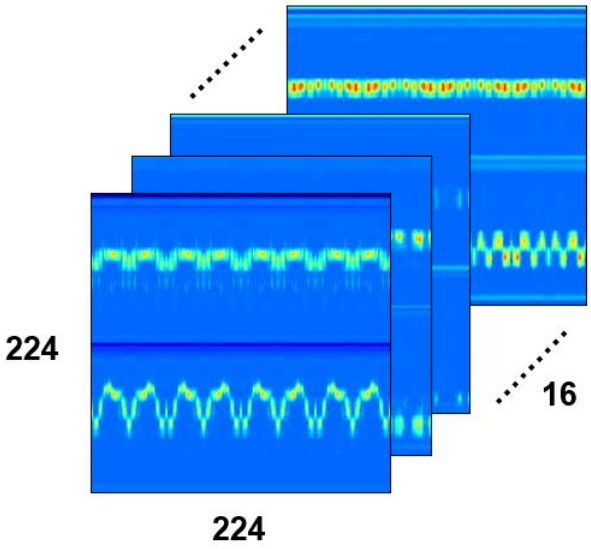

Figure 5.1: The illustration of stacking 16 heatmaps to one 3D feature map.

5.2 Experiments

5.2.1 Experiment setup

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 & counting multitask model are all implemented by TensorFlow framework. Point Cloud Library and SciPy Library are used in heatmap processing and data preparation steps.

There are two training processes in this thesis in total: 2D human pose estimation model MSPN and 2D exercise recognition & counting multitask model. For the MSPN model, we select Adam as the optimizer. The start learning rate is 0.0001 and decay rate $(\beta 1, \beta 2)$ is 0.9 and 0.99 respectively. The epsilon is set to 1e-8, and decay is set to 1e-5. The number of epochs is set to 30, and the first five epochs are leveraged as the warm-up stage. The learning rate increases from 0.0001 at epoch 1 to 0.001 at epoch 5 linearly, and then it drops to almost 0 at the last epoch. The datasets we use to train MSPN are MPII dataset and PennAction dataset. The parameters above cater to both datasets.

For the multitask model, RMSprop optimizer is chosen for both tasks, and default parameters are applied. The initial learning rates are both set to 0.001, and the numbers of total epochs are both 10. The learning rate of training exercise recognition branch is divided by 10 after 1 and 5 epochs, and the learning rate is reduced by 10 times at the 3rd and 6th epoch in training counting part. In the whole training process, the batch size is 8.

5.2.2 Results and Analysis

5.2.2.1 Exercise Recognition

The test dataset of Rep-Penn includes 5967 test samples covering 7 exercises: bench press, clean and jerk, jumping jack, pull up, push up, sit up and squat. Table 5.1 shows the prediction accuracy of each exercise on our multitask model. We got 95.69% accuracy among the 7 exercises, and all the exercise' accuracy is above 90% except 'Sit up'. 'Clean and jerk' gets the highest accuracy at 99.83% while 'Sit up' reaches around 89% accuracy.

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 5.1: Recognition accuracy for each type of exercises

Table 5.2 is the confusion matrix of the exercise recognition task. The exercise types in the first column are groundtruth labels while the names in the first row are the predicted exercises. For example, the first number in the second row 834 shows that 834 test samples whose labels are 'Bench press' are predicted as 'Bench press', which are correct predictions. The 5th number in the second row 30 denotes that 30 'Bench press' samples are predicted as 'Push up' which are incorrect. Therefore, The numbers on the diagonal represent the number of correct predictions of each exercise whilst the other non-zero numbers are all incorrect predictions.

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 1 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 7 percent incorrect predictions which are acceptable.

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

Table 5.2: Confusion matrix for exercise recognition across 7 exercises

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 5.3. 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.

Network	Accuracy(%)	Parameters
Ours-ResNet18	94.25	11323303
Ours-ResNet34	95.69	21442599
Ours-ResNet50	93.78	26263651
Ours-ResNet101	91.20	45334115

Table 5.3: Comparison of Feature extraction network for exercise recognition task

5.2.2.2 Repetition Counting

To find the most suitable feature extraction network for the counting task, ResNet18, ResNet34 and ResNet50 are tested to act as the feature extraction network. The comparison is shown in Table 5.4. 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 the 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.

Network	MAE ↓	OBO ↑	AE ↓	σ ↓	Parameters ↓
ResNet18	0.004	0.997	0.039	0.172	11323575
ResNet34	0.006	0.998	0.059	0.189	21442871
ResNet50	0.015	0.996	0.154	0.261	26444247

Table 5.4: Comparison of Feature extraction network 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.

5.2.2.3 Discussion

The processed heatmaps contain both spatial and temporal information of 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 doesn't 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.

5.3 Comparison with other methods

There are only a few papers [1, 35] focusing on exercise recognition & repetition counting multitask based on RGB images. Both of them create their own datasets which are unavailable. Alatiah et al. [1] generated training data covering three exercises: pull up, push up and squat. Therefore, we compare our model with their work in both tasks based on these 3 same exercises. Khurana et al. [35] included 17 actions in total, among which 4 exercises are the same as this thesis. However, some of them are easily misclassified due to the limitation of training data. Thus, 7 most frequent exercises among 17 exercises are selected for comparison in exercise recognition. In repetition counting, we also include Zhang et al.'s paper [79] 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.

5.3.1 Exercise Recognition

Paper [1] and this thesis share three exercises: pull up, push up and squat in common. We trained our multitask model by only using data of these three physical activities. Table

5.5 compares the accuracy between their work and this thesis across the three exercises. It's shown that Precision of 'Pull up' and 'Push up' and Recall of 'Squat' of this thesis are higher than their work, but other metrics of our work are a bit lower than theirs. The reason is that they applied the reject-option technique[25] which rejects the ambiguous samples if the estimated probability is out of the selected confidence intervals. Despite this, our thesis is still competitive as all the three metrics are within 0.03 of Alatiah's results.

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

Table 5.5: Comparison with paper [1] over 3 exercises: pull up, push up and squat. The metrics with * denotes the result of this thesis while the metrics without * represent the accuracy of paper [1].

We also compare our proposed method with paper Khurana et al.'s [35] research. They collected exercise videos from gym cameras. In sum, videos including 17 exercises are gathered, and they achieved overall 80.60% exercise recognition accuracy. Since their dataset is unavailable, we compare the recognition accuracy of the 7 most frequent exercises they collected with our work. Their 85.7% accuracy over 7 exercises is lower than our 95.69% accuracy. The comparison above proves that our exercise recognition branch has up-to-date accuracy.

5.3.2 Repetition Counting

In this session, we compared our work with paper [1] across three identical exercises first. Afterwards, three literature [1, 35, 79] and this thesis are combined for comparison.

Compared with [1], our AE is ± 0.242 lower than [1]'s ± 1 across the 3 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 don't need extra steps to derive the results.

Method	MAE ↓	OBO ↑	AE ↓	σ ↓
Khurana *et al.*[35]	-	-	±1.7	2.64
Alatiah *et al.*[1]	-	-	±1	-
Zhang *et al.*[79]	0.147	0.79	-	0.243
Proposed method	0.004	0.99	±0.039	0.187

Table 5.6: 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. This thesis includes 7 exercises, and our data are originally derived from PennAction dataset.

The comparison between this thesis and three works [1, 35, 79] is listed in Table 5.6. Note that the result is not fair enough due to the dataset limitation. Our MAE 0.004 is lower than [79]'s 0.147 and OBO 0.99 is much higher than their 0.79. Their dataset UCFRep is more challenging than Rep-Penn. However, our dataset includes videos with more cycles than UCFRep (max of cycles: 7), and more exercises types than their dataset (number of exercises: 5).

In paper [1] and [35], they didn't 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 framework is more accurate than theirs.

5.3.3 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 as Figure 2. The comparison is listed in Table 5.7. 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. Our heatmap-based method achieves greater performance than the non-heatmap method.

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

Table 5.7: Results of exercise recognition and counting using different inputs: heatmaps and joint coordinates.

5.4 Limitation

Due to the dataset limitation, the comparison among other up-to-date methods is based on different datasets. Action recognition datasets such as PennAction [80] and UCF-101 [62] contain exercise videos, but they exclude counting labels. The commonly used counting benchmarks are QUVA Repetition [53] and YTsegments [36]. QUVA dataset includes 100 videos with few exercise videos. YTsegments is unavailable at present. Therefore, we didn't compare related methods using these public datasets.

Moreover, Rep-Penn is generated by concatenating single-period exercise frames from PennAction dataset. It excludes the exercises with shifting view angles. Besides, Rep-Penn only includes continuous workouts with fixed cycle length. The limitation of PennAction dataset and 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.

5.5 Summary

In this chapter, we list the results of our multitask model on Rep-Penn dataset and the comparison with other related works. Our results prove that the proposed multitask model has very good performance in both tasks. The comparison between this thesis and other papers are mostly based on different datasets due to dataset limitation. However, we did compare using the available datasets, and found that our accuracy is higher than their methods. The comparison between our heatmap-based method and the method without heatmaps also proves that the proposed method is reliable. In the next chapter, the thesis is concluded and future work is presented in detail.

Chapter 6

Conclusion and Future Work

In this thesis, we propose a multitask system including an off-the-shelf 2D human pose estimation model MSPN and a multitask model of 2D exercise recognition & repetition counting. To the best of our knowledge, we are among the first to propose a system covering these three fields together. Furthermore, it's 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. [41], 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 PennAction dataset. Various cycles and action speeds are added to enrich Rep-Penn dataset. The architecture of the multitask model is based on ResNet networks, with ResNet34 for the exercise recognition task and ResNet18 for the counting task. A two-stage training strategy is adopted to allow the multitask model to train corresponding layers of exercise recognition and counting on each stage, enabling the model to identify exercises and count repetitions concurrently.

The multitask model is trained on Rep-Penn dataset. It reaches solid accuracy in both exercise recognition and counting. It achieves 95.69% exercise recognition accuracy across 7 indoor exercises. The counting error MAE is 0.004 and OBO accuracy is 0.99. Due to the scarcity of the available datasets and related works, we compare our work with other articles [1, 79, 35] based on different datasets. Both exercise recognition accuracy and counting accuracy of this thesis are among the topmost in these papers. Our multitask model is also proved effective when compared with the non-heatmap method.

Our future work will focus on exploring advanced heatmap processing methods and reduce the size of the multitask system. Moreover, various exercise data should be collected to further improve the model's performance.

References

[1] Talal Alatiah and Chen Chen. "Recognizing Exercises and Counting Repetitions in Real Time". In: *CVPR* abs/2005.03194 (2020). arXiv: 2005.03194. URL: https://arxiv.org/abs/2005.03194.

[2] Mykhaylo Andriluka et al. "2D Human Pose Estimation: New Benchmark and State of the Art Analysis". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014.

[3] Ousman Azy and Narendra Ahuja. "Segmentation of periodically moving objects". In: *Proceedings - International Conference on Pattern Recognition*. Institute of Electrical and Electronics Engineers Inc., 2008. ISBN: 9781424421756. DOI: 10.1109/icpr.2008.4760949.

[4] Alexia Briassouli and Narendra Ahuja. "Extraction and analysis of multiple periodic motions in video sequences". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.7 (2007), pp. 1244–1261. ISSN: 01628828. DOI: 10.1109/TPAMI.2007.1042.

[5] Adrian Bulat et al. "Toward fast and accurate human pose estimation via soft-gated skip connections". In: *CVPR* abs/2002.11098 (2020). arXiv: 2002.11098. URL: https://arxiv.org/abs/2002.11098.

[6] Carlos Caetano et al. "Magnitude-Orientation Stream network and depth information applied to activity recognition". In: *Journal of Visual Communication and Image Representation* 63 (Aug. 2019), p. 102596. ISSN: 10473203. DOI: 10.1016/j.jvcir.2019.102596.

[7] Zhe Cao et al. “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1 (2021), pp. 172–186. DOI: 10.1109/TPAMI.2019.2929257.

[8] Joao Carreira et al. “Human pose estimation with iterative error feedback”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem (2016), pp. 4733–4742. ISSN: 10636919. DOI: 10.1109/CVPR.2016.512. arXiv: 1507.06550.

[9] João Carreira and Andrew Zisserman. “Quo Vadis, action recognition? A new model and the kinetics dataset”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (2017), pp. 4724–4733. DOI: 10.1109/CVPR.2017.502. arXiv: 1705.07750.

[10] Keng Hao Chang, Mike Y. Chen, and John Canny. “Tracking free-weight exercises”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4717 LNCS (2007), pp. 19–37. ISSN: 16113349. DOI: 10.1007/978-3-540-74853-3_2.

[11] Guilhem Cheron, Ivan Laptev, and Cordelia Schmid. “P-CNN: Pose-Based CNN Features for Action Recognition”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.

[12] Dmitry Chetverikov and Sándor Fazekas. “On motion periodicity of dynamic textures”. In: *BMVC 2006 - Proceedings of the British Machine Vision Conference 2006* (2006), pp. 167–176. DOI: 10.5244/c.20.18.

[13] Vasileios Choutas et al. “PoTion: Pose MoTion Representation for Action Recognition”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), pp. 7024–7033. ISSN: 10636919. DOI: 10.1109/CVPR.2018.00734.

[14] Wen-Sheng Chu, Feng Zhou, and Fernando De la Torre. "Unsupervised Temporal Commonality Discovery". In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 373–387. ISBN: 978-3-642-33765-9.

[15] Ross Cutler and Larry S. Davis. "Robust real-time periodic motion detection, analysis, and applications". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 781–796. ISSN: 01628828. DOI: 10.1109/34.868681.

[16] Qi Dang et al. "Deep learning based 2D human pose estimation: A survey". In: *Tsinghua Science and Technology* 24.6 (2019), pp. 663–676. ISSN: 18787606. DOI: 10.26599/TST.2018.9010100.

[17] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[18] Debidatta Dwibedi et al. "Counting Out Time: Class Agnostic Video Repetition Counting in the Wild". In: *CVPR* abs/2006.15418 (2020). arXiv: 2006.15418. URL: https://arxiv.org/abs/2006.15418.

[19] Abdulmotaleb El Saddik. "Digital Twins: The Convergence of Multimedia Technologies". In: *IEEE MultiMedia* 25.2 (2018), pp. 87–92. DOI: 10.1109/MMUL.2018.023121167.

[20] Tadilo Endeshaw, Johan Garcia, and Andreas Jakobsson. "Classification of indecent videos by low complexity repetitive motion detection". In: *Proceedings - Applied Imagery Pattern Recognition Workshop* (2008), pp. 2–8. ISSN: 15505219. DOI: 10.1109/AIPR.2008.4906438.

[21] M. Farrajota, João M.F. Rodrigues, and J. M.H. du Buf. "Human action recognition in videos with articulated pose information by deep networks". In: *Pattern Analysis and Applications* (Nov. 2018). ISSN: 14337541. DOI: 10.1007/s10044-018-0727-y.

[22] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. "Convolutional Two-Stream Network Fusion for Video Action Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[23] Rogelio Gámez Díaz et al. "Digital Twin Coaching for Physical Activities: A Survey". In: *Sensors* 20.20 (2020). ISSN: 1424-8220. DOI: 10.3390/s20205936. URL: https://www.mdpi.com/1424-8220/20/20/5936.

[24] Muhammad Gandomkar, Reza Sarang, and Ziba Gandomkar. "TrainingPal: An Algorithm for Recognition and Counting Popular Exercises Using Smartphone Sensors". In: *26th Iranian Conference on Electrical Engineering, ICEE 2018* (2018), pp. 1471–1476. DOI: 10.1109/ICEE.2018.8472444.

[25] Blaise Hanczar and Edward R. Dougherty. "Classification with reject option in gene expression data". In: *Bioinformatics* 24.17 (July 2008), pp. 1889–1895. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btn349. URL: https://doi.org/10.1093/bioinformatics/btn349.

[26] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[27] Kaiming He et al. "Identity mappings in deep residual networks". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9908 LNCS (2016), pp. 630–645. ISSN: 16113349. DOI: 10.1007/978-3-319-46493-0_38. arXiv: 1603.05027.

[28] Geoffrey E. Hinton, Oriol Vinyals, and J. Dean. "Distilling the Knowledge in a Neural Network". In: *ArXiv* abs/1503.02531 (2015).

[29] Yukun Huang, Yongcai Guo, and Chao Gao. "Efficient Parallel Inflated 3D Convolution Architecture for Action Recognition". In: *IEEE Access* 8 (2020), pp. 45753–45765. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2978223.

[30] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 448–456. URL: http://proceedings.mlr.press/v37/ioffe15.html.

[31] Shuiwang Ji et al. "3D Convolutional neural networks for human action recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 221–231. ISSN: 01628828. DOI: 10.1109/TPAMI.2012.59.

[32] Sam Johnson and Mark Everingham. "Clustered pose and nonlinear appearance models for human pose estimation". In: *British Machine Vision Conference, BMVC 2010 - Proceedings* ii (2010), pp. 1–11. DOI: 10.5244/C.24.12.

[33] Aouaidjia Kamel et al. "An Investigation of 3D Human Pose Estimation for Learning Tai Chi: A Human Factor Perspective". In: *International Journal of Human-Computer Interaction* 35.4-5 (2019), pp. 427–439. ISSN: 15327590. DOI: 10.1080/10447318.2018.1543081. URL: https://doi.org/10.1080/10447318.2018.1543081.

[34] Andrej Karpathy et al. "Large-scale Video Classification with Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014.

[35] Rushil Khurana et al. "GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2, 4, Article* 185 (2018), p. 17. DOI: 10.1145/3287063. URL: https://doi.org/10.1145/3287063.

[36] Ofir Levy and Lior Wolf. "Live repetition counting". In: *Proceedings of the IEEE International Conference on Computer Vision*. Vol. 2015 Inter. 2015, pp. 3020–3028. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.346.

[37] Wenbo Li et al. “Rethinking on Multi-Stage Networks for Human Pose Estimation”. In: *CVPR* abs/1901.00148 (2019). arXiv: 1901.00148. URL: http://arxiv.org/abs/1901.00148.

[38] Tsung Yi Lin et al. “Microsoft COCO: Common objects in context”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8693 LNCS.PART 5 (2014), pp. 740–755. ISSN: 16113349. DOI: 10.1007/978-3-319-10602-1_48. arXiv: 1405.0312.

[39] F. Liu and R.W. Picard. “Finding periodicity in space and time”. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 376–383. DOI: 10.1109/ICCV.1998.710746.

[40] Mengyuan Liu and Junsong Yuan. “Recognizing Human Actions as the Evolution of Pose Estimation Maps”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), pp. 1159–1168. ISSN: 10636919. DOI: 10.1109/CVPR.2018.00127.

[41] Mengyuan Liu et al. “Joint Dynamic Pose Image and Space Time Reversal for Human Action Recognition from Videos”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (July 2019), pp. 8762–8769. DOI: 10.1609/aaai.v33i01.33018762.

[42] Xiwen Liu et al. “Wi-CR: Human Action Counting and Recognition with Wi-Fi Signals”. In: *2019 4th International Conference on Computing, Communications and Security, ICCCS 2019* (2019). DOI: 10.1109/CCCS.2019.8888113.

[43] Dennis Ludl, Thomas Gulde, and Cristóbal Curio. “Simple yet efficient real-time pose-based action recognition”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 581–588. DOI: 10.1109/ITSC.2019.8917128.

[44] Diogo Luvizon, David Picard, and Hedi Tabia. “Multi-task deep learning for real-time 3D human pose estimation and action recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* (2020).

[45] Diogo C Luvizon, David Picard, and Hedi Tabia. "2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5137–5146. ISBN: 9781538664209. DOI: 10.1109/CVPR.2018.00539. arXiv: 1802.09232.

[46] Diogo C Luvizon, Hedi Tabia, and David Picard. "Human pose regression by combining indirect part detection and contextual information". In: *Computers & Graphics* 85 (2019), pp. 15–22.

[47] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9912 LNCS. Mar. 2016, pp. 483–499. ISBN: 9783319464831. DOI: 10.1007/978-3-319-46484-8_29. arXiv: 1603.06937. URL: http://arxiv.org/abs/1603.06937.

[48] Aiden Nibali et al. "Numerical Coordinate Regression with Convolutional Neural Networks". In: *arXiv* (Jan. 2018). arXiv: 1801.07372. URL: http://arxiv.org/abs/1801.07372.

[49] Costas Panagiotakis, Giorgos Karvounas, and Antonis Argyros. "Unsupervised Detection of Periodic Segments in Videos". In: *Proceedings - International Conference on Image Processing, ICIP*. 2018, pp. 923–927. ISBN: 9781479970612. DOI: 10.1109/ICIP.2018.8451336. URL: http://www.ics.forth.gr/cvrl/pd.

[50] E. Pogalin, A.W.M. Smeulders, and A.H.C. Thean. "Visual quasi-periodicity". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587509.

[51] Yuhui Quan, Yan Huang, and Hui Ji. "Dynamic texture recognition via orthogonal tensor dictionary learning". In: *Proceedings of the IEEE International Conference*

on Computer Vision 2015 Inter (2015), pp. 73–81. ISSN: 15505499. DOI: 10.1109/ICCV.2015.17.

[52] A. N. Rajagopalan and R. Chellappa. "Higher-order spectral analysis of human motion". In: *IEEE International Conference on Image Processing* 3 (2000), pp. 2–5. DOI: 10.1109/icip.2000.899337.

[53] Tom F. H. Runia, Cees G. M. Snoek, and Arnold W. M. Smeulders. "Real-World Repetition Estimation by Div, Grad and Curl". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9009–9017. DOI: 10.1109/CVPR.2018.00939.

[54] Abdulmotaleb El Saddik, Fedwa Laamarti, and Mohammad Alja'Afreh. "The Potential of Digital Twins". In: *IEEE Instrumentation Measurement Magazine* 24.3 (2021), pp. 36–41. DOI: 10.1109/MIM.2021.9436090.

[55] Ben Sapp and Ben Taskar. "MODEC: Multimodal decomposable models for human pose estimation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2013), pp. 3674–3681. ISSN: 10636919. DOI: 10.1109/CVPR.2013.471.

[56] Krishanu Sarker et al. "Towards Robust Human Activity Recognition from RGB Video Stream with Limited Labeled Data". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, pp. 145–151. DOI: 10.1109/ICMLA.2018.00029.

[57] Christian Seeger, Alejandro Buchmann, and Kristof Van Laerhoven. "MyHealthAssistant: A Phone-based body sensor network that captures the wearer's exercises throughout the day". In: *BODYNETS 2011 - 6th International ICST Conference on Body Area Networks* (2012), pp. 1–7. DOI: 10.4108/icst.bodynets.2011.247015.

[58] Mattia Segu et al. "Depth-Aware Action Recognition: Pose-Motion Encoding through Temporal Heatmaps". In: *arXiv preprint arXiv:2011.13399* (2020).

[59] Anshul Shah et al. "Pose And Joint-Aware Action Recognition". In: *CVPR* abs/2010.08164 (2020). URL: https://arxiv.org/abs/2010.08164.

[60] Shahriar Shariat and Vladimir Pavlovic. "Robust time-series retrieval using probabilistic adaptive segmental alignment". In: *Knowledge and Information Systems* 49.1 (2016), pp. 91–119. ISSN: 02193116. DOI: 10.1007/s10115-015-0898-4. arXiv: 1609.08201.

[61] Karen Simonyan and Andrew Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *CVPR* abs/1406.2199 (2014). arXiv: 1406.2199. URL: http://arxiv.org/abs/1406.2199.

[62] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv:1212.0402* (2012).

[63] Andrea Soro et al. "Recognition and repetition counting for complex physical exercises with deep learning". In: *Sensors (Switzerland)* 19.3 (2019). ISSN: 14248220. DOI: 10.3390/s19030714.

[64] Jennifer J. Sun et al. "View-Invariant Probabilistic Embedding for Human Pose". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 53–70. ISBN: 978-3-030-58558-7.

[65] Xiao Sun et al. "Integral Human Pose Regression". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[66] Ashwin Thangali and Stan Sclaroff. "Periodic motion detection and estimation via space-time sampling". In: *Proceedings - IEEE Workshop on Motion and Video Computing, MOTION 2005* (2005), pp. 176–182. DOI: 10.1109/ACVMOT.2005.91.

[67] Alexander Toshev and Christian Szegedy. "DeepPose: Human Pose Estimation via Deep Neural Networks". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1653–1660.

[68] Christopher J. Tralie and Jose A. Perea. "(Quasi)periodicity quantification in video data, using topology". In: *arXiv* 11.2 (2017), pp. 1049–1077. ISSN: 23318422.

[69] Du Tran et al. "ConvNet architecture search for spatiotemporal feature learning". In: *arXiv* section 3 (2017). ISSN: 23318422. arXiv: 1708.05038.

[70] Du Tran et al. "Learning spatiotemporal features with 3D convolutional networks". In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 4489–4497. ISSN: 15505499. DOI: 10.1109/ICCV.2015.510. arXiv: 1412.0767.

[71] Mikael Vejdemo-Johansson et al. "Cohomological learning of periodic motion". In: *Applicable Algebra in Engineering, Communications and Computing* 26.1-2 (2015), pp. 5–26. ISSN: 09381279. DOI: 10.1007/s00200-015-0251-x.

[72] Heng Wang et al. "Dense trajectories and motion boundary descriptors for action recognition". In: *International journal of computer vision* 103.1 (2013), pp. 60–79.

[73] Jianbo Wang et al. "AI Coach: Deep Human Pose Estimation and Analysis for Personalized Athletic Training Assistance". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: Association for Computing Machinery, 2019, pp. 374–382. ISBN: 9781450368896. DOI: 10.1145/3343031.3350910. URL: https://doi.org/10.1145/3343031.3350910.

[74] Shih En Wei et al. "Convolutional pose machines". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-Decem. IEEE Computer Society, Dec. 2016, pp. 4724–4732. ISBN: 9781467388504. DOI: 10.1109/CVPR.2016.511. arXiv: 1602.00134.

[75] Hao Yang et al. "Asymmetric 3D Convolutional Neural Networks for action recognition". In: *Pattern Recognition* 85 (2019), pp. 1–12. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2018.07.028. URL: https://www.sciencedirect.com/science/article/pii/S0031320318302632.

[76] Zhengyuan Yang et al. “Action recognition with spatio–temporal visual attention on skeleton image sequences”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.8 (2018), pp. 2405–2415.

[77] Jianqin Yin et al. “Energy-based Periodicity Mining with Deep Features for Action Repetition Counting in Unconstrained Videos”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (2021). ISSN: 15582205. DOI: 10.1109/TCSVT.2021.3055220. arXiv: 2003.06838.

[78] Feng Zhang, Xiatian Zhu, and Mao Ye. “Fast human pose estimation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2019-June. IEEE Computer Society, June 2019, pp. 3512–3521. ISBN: 9781728132938. DOI: 10.1109/CVPR.2019.00363.

[79] Huaidong Zhang et al. “Context-aware and scale-insensitive temporal repetition counting”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2020, pp. 667–675. DOI: 10.1109/CVPR42600.2020.00075. arXiv: 2005.08465. URL: https://github.com/Xiaodomgdomg/.

[80] Weiyu Zhang, Menglong Zhu, and Konstantinos G. Derpanis. “From Actemes to Action: A Strongly-Supervised Representation for Detailed Action Understanding”. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 2248–2255. DOI: 10.1109/ICCV.2013.280.