www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 10 Issue 11 November 2021, Page No. 25413-25419
ISSN: 2319-7242 DOI: 10.18535/ijecs/v10i11.4631

Video-Based Rope Skipping Repetition Counting with ResNet Model

Xinxin Li[1], Jiawen Wang[2]

[1]School of Computer Science and Technology, Shandong University of Technology, China
[2]Cangzhou Technical College, China

Abstract
Video Repetition Counting is one of the important research areas in computer vision. It focuses on estimating the number of repeating actions. In this paper, we propose a method for video-based rope skipping repetition counting that combines the ResNet Model and a counting algorithm. Each frame in the given video is first classified into two categories: upward and downward, describing its current motion status. Then the classification sequence of the video is processed by a statistical counting algorithm to obtain the final repetition number. The experiments on real-world videos show the efficiency of our model.

Keywords: *Repetition Couting, ResNet Model, Counting Algorithm*

Introduction
Physical exercise plays an important role in human health. It can improve the growth and development of the human body. Repetition counting is one of the important research areas in computer vision. It can be applied to repetitive activities such as physical exercises, music performances, and fitness. Rope skipping is a sport that everyone can participate in, which shows less damage to the human body. Video-based Repetition Counting can greatly reduce the complexity of manual statistics.

Video-based methods are the most common methods for repetition counting. Cutler tracked the object and applied Fourier time-frequency analysis to detect the periodic action by calculating the self-similarity of the object over time [1]. Dwibedi presented a RepNet model, using self-similarity as an intermediate representation, and generalized it to unseen videos. The model was trained on a large number of unlabeled synthetic videos which contains video clips sampled at different periods and lengths [2]. Levy and Wolf trained a CNN model to evaluate the period length of each block and used the entropy of the model output to determine the beginning and end of the repetition [3]. The disadvantage of this method is that the duration length of the repetition and its frequencies cannot be changed.

In this paper, we proposed a method for video-based rope skipping repetition counting based on ResNet model and a statistical counting algorithm. Our method can process rope skipping repetition counting in different environments. The details of our method are described in section 3, and section 4 shows the experimental results.

Related Work
From the viewpoint of the data source, repetition counting methods can be divided into sensor-based and video-based methods. Currently, video-based repetition counting methods include signal processing methods, period detection methods, and frame boundary detection methods.

One of the earliest methods is to convert motion into a one-dimensional signal, and then use signal processing methods, such as Fourier transform, peak detection, singular value decomposition, and other methods to extract frequency information. This method assumes that the movement is periodic and stationary, and is not suitable for many non-stationary situations. Cutler tracked the object, calculated the self-similarity of the object over time, detected the periodic operation using Fourier time-frequency analysis, and used the 2D

grid structure of the self-similarity matrix to analyze the periodicity [1]. Thangali took the intensity pattern of the linear sampling path along with the time and space of the video frame sequence to estimate the period of motion [4]. The sampling path is composed of the first frame and the last frame of intensity motion. The peak of intensity motion can be estimated by the least square method. Pogalin presented a framework to divide motion into 10 categories and proposed an algorithm to detect these categories [5]. The algorithm can track the independent features of the object and perform repetition counting through probabilistic PCA and spectrum analysis. Azy adopted the maximum likelihood estimation method to estimate the period and used the correlation of image segmentation to segment the object [6]. Correlation is used to represent the best position of the object in each frame, and the segmentation tree is used to describe image segmentation. Runia et al. designed three types of motion and three motion continuities from the 3D divergence, gradient, and curl operators, and then generated 18 basic situations in 2D perception [7]. This paper introduced a different flow-based representation, used the wavelet transform to solve non-stationary scenes, and selected the most discriminative signal based on the quality evaluation.

Another approach is to view period detection as finding the similarity between two video clips, e.g. similarity matrix. Panagiotakis proposed an unsupervised method to find the similarity of two video clips, using a similarity matrix to calculate the Euclidean distance of any frame in the input video, and to detect periodic segments in the video and its period [8]. Dwibedi used self-similarity as an intermediate representation and generalized it to unseen videos [2]. This paper presented a RepNet model, trained on a large number of unlabeled synthetic videos. Synthetic videos are video clips sampled at different periods and lengths, based on a category-agnostic prediction method.

Recognizing different video frames can also be used for repetition counting. Levy and Wolf analyzed sequential blocks of different lengths, trained a CNN model to evaluate the period length of each block, and used the entropy of model output to determine the beginning and end of repetition [3]. This method can be applied in real situations. Its disadvantage is that the period length of the repetition and the frequencies cannot be changed. For the hand-tremor problem, Pintea proposed two methods to detect frequency [9]. The first method used the Lagrangian method to detect the hand's position in each frame, and the second method adopted the Euler method to locate the hand's position and then detect the frequency along the trajectory of the hand. To count the repetition of motions, Zhang proposed a coarse-to-fine search method to search the repetition length, avoiding large-scale search complexity [10]. A bidirectional length estimation method was proposed to predict the position of the repetition period using a context-sensitive regression method.

3D human pose estimation can be used to calculate joint coordinates. Alatiah presented an algorithm to identify and track athletes, and then used a deep learning method to detect motion types and repetition counting [11]. Khurana proposed GymCam system, which can use cameras to automatically detect, recognize, and track the movements of multiple people at the same time [12]. The system didn't estimate the key joints of the body, but introduced a classifier to segment and identify the motion categories and a multi-layer perceptron regression to predict the number of repetitions. Yu proposed a multi-task system that combined human pose estimation, action recognition, and repetition counting [13]. This system used a deep learning model, which took joint positions and human motion features from the heat map as input. Ferreira extracted skeleton features and deep semantic features from the 2D human pose estimation network and employed an MLP classifier and encoder network to predict the key moments of the frame [14]. To solve the problem of unbalanced data sets, key pose labels are generated based on skeleton features.

Besides video-based methods, there are many sensor-based methods. Skawinski used a wearable 3D accelerator to obtain monitoring data and proposed a 5-layer CNN model to divide the motions into different categories [15]. Then PCA and vertex detection algorithms were used to count the number of repetitions. Soro introduced a CNN network to identify the motion types, used a neural network method to predict the starting position of action, and then calculated the total number of repetitions [16]

Our Method

Our method first converts the video into a sequence of frames, and each frame is processed by a dense optical flow algorithm to extract the motion information. Then, a ResNet model is used to predict the motion categories (upward or downward) of each frame. Finally, a statistical counting method is used to calculate the repetition numbers. The method proposed in this paper is shown in figure 1.

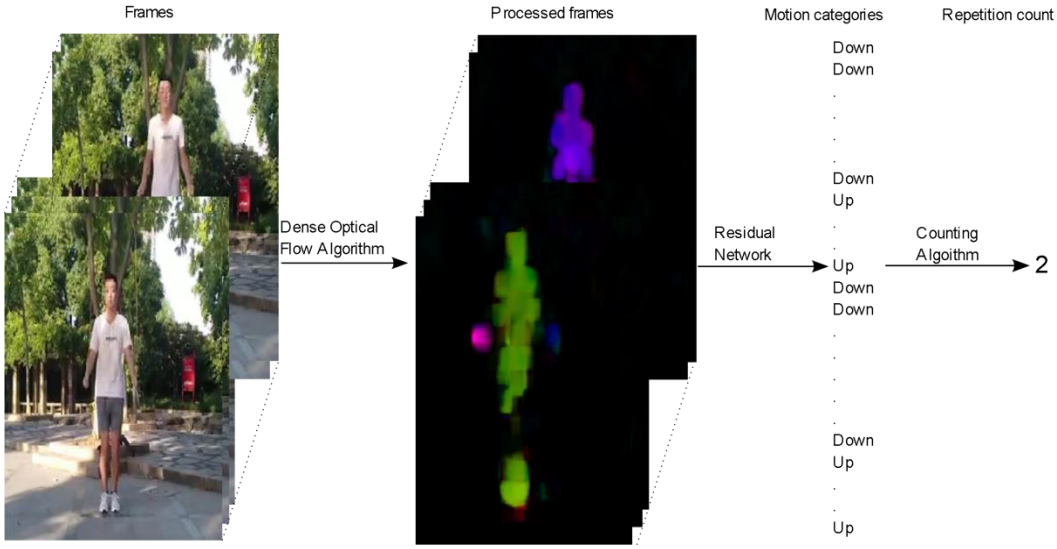

Figure 1. Our Method

Farneback Dense Optical Flow Algorithm

The optical flow method calculates the motion of the objects between adjacent frames by using the changes of the pixels in the image sequence and the correlation between adjacent frames [17]. The dense optical flow algorithm proposed by Gunner Farneback performs better than the sparse optical flow algorithm [18]. Farneback algorithm is a gradient-based method that assumes that the image gradient is constant and that the local optical flow is constant. The local optical flow is constant, that is, for any $y \in N(x), d = \partial X / \partial t$ doesn't change.

Farneback algorithm takes the neighbors of a pixel (usually a square area of size of 2n+1 with the pixel as the center) and uses the values and coordinates of these pixels to estimate the coefficients with a weighted least square method. These pixels closer to the center have a greater correlation with the center pixel, and the farther points provide less information. The weights of pixels outside the neighborhood can be set as 0.

If the pixel is moved d, the entire polynomial should change. The original location is

$$f_1(x) = x^T A_1 x + b_1^T x + c_1 \quad (1)$$

After the pixel is moved, the location is

$$\begin{aligned} f_2(x) &= f_1(x-d) \\ &= (x-d)^T A_1 (x-d) + b_1^T (x-d) + c_1 \\ &= x^T A_1 x + (b_1 - 2A_1 d)^T x + d^T A_1 d - b_1^T d + c_1 \\ &= x^T A_2 x + b_2^T x + c_2 \end{aligned} \quad (2)$$

Where,

$$A_2 = A_1 \quad (3)$$

$$b_2 = b_1 - 2A_1 d \quad (4)$$

$$c_2 = d^T A_1 d - b_1^T d + c_1 \quad (5)$$

If A_1 is not singular, then the formula 4 above can be obtained:

$$d = -\frac{1}{2} A_1^{-1} (b_2 - b_1) \quad (6)$$

In our method, we calculate the motion features for each frame of input video using Farneback dense optical flow algorithm.

Res Net Model

We classify each frame into two motion categories: upward, downward. Each rope skipping is composed of a sequence of upward and downward motions. After extracting the motion features, we adopt the residual network as our motion classifier [19]. For traditional convolutional neural networks, with the increase of network levels, the accuracy of the model continues to improve. But when the network level increases to a certain number, the training accuracy decreases, which shows that when the network becomes very deep, the deep network becomes difficult to train.

ResNet model borrows the idea of highway network[20]. With the addition of network layers, the performance of the model becomes better. The basic structure of the residual network is shown in figure 2.

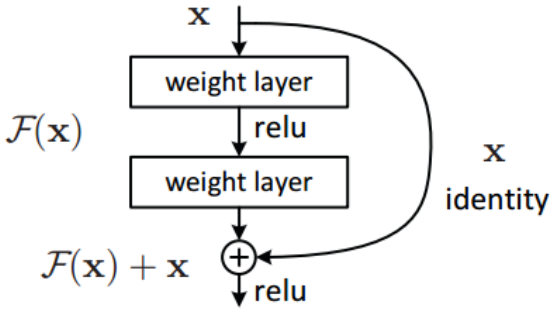

Figure 2: Residual block

Suppose that the input of a neural network is x, and the expected output is H(x). In the training phase, through the shortcut connection method, the input x is directly passed to the output as the initial result, and the output result is H(x)=F(x)+x. From figure 2, we can see that the residual structure can directly cross several layers as the input of a layer.

Counting Algorithm

A complete repetition cycle of rope skipping can be viewed as a sequence of upward and downward motions. Since we obtained the upward and downward motions for all frames, we can calculate the repetition numbers. In this paper, we proposed a new counting algorithm by considering all upward and downward motions.

Algorithm: Counting Algorithm

Input: motion sequence (m_1,m_2,...,m_n)
Output: repetition number k

```
k <- 0
up_counter <- 0
down_counter <- 0
for m in (m1,m2,...,mn)
        if m is 'up':
                if down_counter > 5:
                        k <- k+1
                up_counter <- up_counter+1
                down_counter <- 0
        elif m is 'down':
                down_counter <- down_counter+1
                if down_counter > 3:
                        up_counter <- 0
return k
```

After observing the rope skipping video, we find that the number of downward motions is more than upward motions. Therefore, our algorithm chooses the downward motion as the main reference. When the downward motion occurs more than 3 times, we set the upward count to 0, which means it is in the downward state. If the current motion is upward and the downward motions occur more than 5 times, it indicates that a rope skipping cycle is completed, so the downward count is set to 0.

Experiments

Dataset and Experimental Setting

The videos of rope skipping are collected from the internet. To diversify the dataset, we gathered videos of different people in different environments. When we got the videos, Farneback dense optical flow algorithm was used to process the frames in these videos. The processed frames are annotated into two categories: upward and downward. The distributions of our dataset are shown in table 1.

Table 1: Dataset information

Cagegories / Da	Training	Test
Upward	800	200
Downward	800	200

Both upward and downward categories include 1000 frames separately, in which 800 frames are used as training data and 200 frames as test data. Examples of upward and downward frames are shown in figure 3 below.

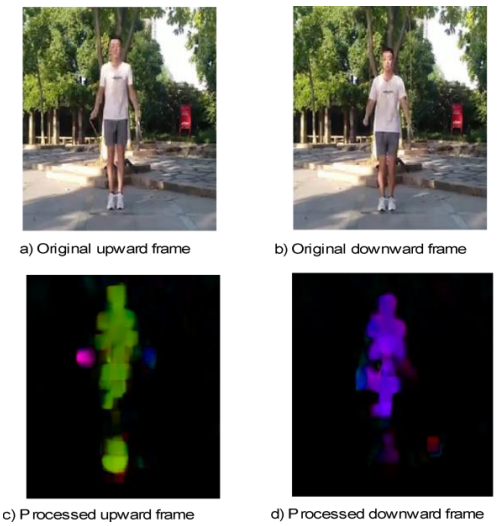

Figure 3: Examples of upward and downward frames

In this paper, we use Mean Absolute Error (MAE) as the evaluation metrics. This metric calculates the difference between the predicated count and the golden count, and then divided by the golden count. The final MAE is the mean of the MAEs on the entire dataset.

Experiments for Frame Classification

In the experiments, we use ResNet18 for frame classification and compare it with traditional convolutional neural networks. Figure 4 shows the performance of these models in different training iterations. It's shown that ResNet18 model is more accurate than CNN model, and ResNet18 achieves its best performance on the 7th iteration.

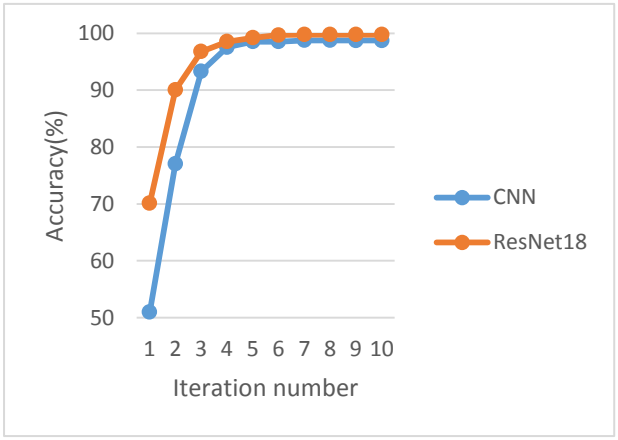

Figure 4: Accuracy on different models

Experiments on videos

When we perform ResNet18 model on videos, there are some images that are not correctly classified. The reason might be that some images are in landing situation, losing both upward and downward features after processed by Farneback algorithm. Therefore, a counting algorithm to process those misclassified categories is needed. We apply our counting algorithm to 10 collected videos with a length of less than 1 minute. We compare MAE on counting algorithms using category sequences, and category sequences generated by ResNet19 model without counting algorithm. The results are compared in Table 2. It's shown that our counting algorithm improves the performance about 18.6%.

Table 2: The error rate of counting algorithm

Algorithms	MAE
ResNet18	19.2%
Counting algorithm (CNN)	1.7%
Counting algorithm (ResNet18)	0.6%

Conclusion

In this paper, we proposed a method for video-based rope skipping repetition counting. The frames in a rope skipping video processed with Farneback Algorithm are classified into a sequence with upward and downward categories. Then a statistical counting algorithm is used to calculate the repetition number. In future, we consider extend this method into more complex cases.

References

[1] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 781–796, 2000, doi: 10.1109/34.868681.

[2] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "Counting out time: Class agnostic video repetition counting in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Jun. 2020, pp. 10387–10396.

[3] O. Levy and L. Wolf, "Live repetition counting," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, Dec. 2015, pp. 3020–3028.

[4] A. Thangali and S. Sclaroff, "Periodic motion detection and estimation via space-time sampling," in *2005 seventh IEEE workshops on applications of computer vision (WACV/MOTION'05) - volume 1*, 2005, vol. 2, pp. 176–182. doi: 10.1109/ACVMOT.2005.91.

[5] 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.

[6] O. Azy and N. Ahuja, "Segmentation of periodically moving objects," United States, 2008. doi: 10.1109/icpr.2008.4760949.

[7] T. F. H. Runia, C. G. M. Snoek, and A. 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.

[8] C. Panagiotakis, G. Karvounas, and A. Argyros, "Unsupervised detection of periodic segments in videos," in *2018 25th IEEE international conference on image processing (ICIP)*, 2018, pp. 923–927. doi: 10.1109/ICIP.2018.8451336.

[9] S. L. Pintea, J. Zheng, X. Li, P. J. M. Bank, J. J. van Hilten, and J. C. van Gemert, "Hand-tremor frequency estimation in videos," Sep. 2018.

[10] H. Zhang, X. Xu, G. Han, and S. He, "Context-aware and scale-insensitive temporal repetition counting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, Jun. 2020, pp. 670–678.

[11] T. Alatiah and C. Chen, "Recognizing exercises and counting repetitions in real time." 2020.

[12] R. Khurana, K. Ahuja, Z. Yu, J. Mankoff, C. Harrison, and M. Goel, "GymCam: Detecting, recognizing and tracking simultaneous exercises in unconstrained scenes," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 4, pp. 1–17, Dec. 2018, doi: 10.1145/3287063.

[13] Q. Yu, H. Wang, F. Laamarti, and A. El Saddik, "Deep learning-enabled multitask system for exercise recognition and counting," *Multimodal Technologies and Interaction*, vol. 5, no. 9, 2021, doi: 10.3390/mti5090055.

[14] B. Ferreira *et al.*, "Deep learning approaches for workout repetition counting and validation," *Pattern Recognition Letters*, vol. 151, pp. 259–266, 2021, doi: https://doi.org/10.1016/j.patrec.2021.09.006.

[15] K. Skawinski, F. Montraveta Roca, R. D. Findling, and S. Sigg, "Workout type recognition and repetition counting with CNNs from 3D acceleration sensed on the chest," in *Advances in computational intelligence*, Cham, 2019, pp. 347–359.

[16] A. Soro, G. Brunner, S. Tanner, and R. Wattenhofer, "Recognition and repetition counting for complex physical exercises with deep learning," *Sensors*, vol. 19, no. 3, 2019, doi: 10.3390/s19030714.

[17] [A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*,

2017, pp. 2720–2729. doi: 10.1109/CVPR.2017.291.

[18] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image analysis*, Berlin, Heidelberg, 2003, pp. 363–370.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Jun. 2016, pp. 770–778.

[20] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training Very Deep Networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2377–2385.