LETTER

A Monkey Swing Counting Algorithm Based on Object Detection

Hao CHEN[†], ***Nonmember***, **Zhe-Ming LU**[†a)], ***Member***, ***and*** **Jie LIU**[††,†††], ***Nonmember***

SUMMARY This Letter focuses on deep learning-based monkeys' head swing counting problem. Nowadays, there are very few papers on monkey detection, and even fewer papers on monkeys' head swing counting. This research tries to fill in the gap and try to calculate the head swing frequency of monkeys through deep learning, where we further extend the traditional target detection algorithm. After analyzing object detection results, we localize the monkey's actions over a period. This Letter analyzes the task of counting monkeys' head swings, and proposes the standard that accurately describes a monkey's head swing. Under the guidance of this standard, the monkeys' head swing counting accuracy in 50 test videos reaches 94.23%.

key words: *monkey detection, object detection, head swing counting, YOLO*

1. Introduction

In the field of biomedicine [1], monkeys are important experimental objects, and we need to observe the action of monkey before and after taking drugs in order to judge the effect of them. As for monkeys, the frequency of head swing is one of the most important factors. Therefore, calculating the number of times monkeys swing their heads is of great significance to judge the abnormal behavior of monkeys after taking drugs. Accurate head swing counting is an important indicator to verify the experimental results. In the past, the number of monkeys' head swings was usually calculated manually. This method is very accurate, but once the video length is very long or there are many videos, it needs to consume almost the same time as the video duration. Moreover, manual counting consumes a lot of work force. In view of this contradiction, this paper hopes to propose a method based on deep learning, which can automatically detect monkeys in videos, locate them, and then calculate the number of times monkeys swing their heads in videos [2].

In recent years, as one of the three major tasks of computer vision, object detection has developed rapidly, and a large number of excellent works have emerged, such as R-CNN [3], SSD [4], Faster R-CNN [5] and YOLO [6]. R-CNN uses two parts, one extracting about 2000 regions through the RPN module, the other judging whether these areas contain the target through a classifier. YOLO and SSD output all information through the network, including the target frame and the probability that the target belongs to different categories. The detection result of R-CNN algorithm is more accurate, but the training process leads to higher training complexity. The YOLO algorithm is simple enough and has a low training cost, which is favored by the industry, but the detection accuracy is lower than R-CNN. Considering the application requirements of this project, we mainly use YOLO as the detection algorithm.

In this Letter, the monkey swing counting algorithm is implemented by extending YOLO. Considering that this project only calculates the number of times monkeys shake their heads, we trained the model with a set of monkey pictures with labeled heads. First, the monkey head is detected by YOLO, and the coordinates of the bounding box are obtained [6]. Then, according to the boundary box of the monkey head, we get the position of the monkey head in the picture. Through continuous experiments, we get the factors that affect the number of swings. Finally, we obtained the basis to judge the monkey's head swing and the parameters to describe the monkey's head swing accurately. The main contributions of our work are as follows: First, we combine target detection with biological action recognition to realize the monkey's head swing counting. Second, we explore the movements of the monkeys in their head swing to put forward the behavioral standards for accurately describing the monkeys' head swing. Third, our work can be extended to other biometric action recognition fields.

The remainder of this Letter is organized as follows. Section 2 describes the relevant algorithm of the monkey swing counting. Section 3 mainly introduces the monkey swing counting method. Section 4 shows experimental results. Section 5 concludes the whole Letter.

2. Related Work

2.1 Monkey Detection Algorithm

Since 2020, many teams have been devoted to working on monkey detection [1]. The strategy was only employed in agriculture at the time since the academic intersection was not as great. Even while the team has started using deep learning to identify monkeys, its use is still restricted to just that and does not delve deeply into the target's action description. The recognition of human key points has developed

Manuscript received August 15, 2023.
Manuscript revised October 29, 2023.
Manuscript publicized December 7, 2023.

[†]The authors are with the School of Aeronautics and Astronautics, Zhejiang University, No.38, Zheda Road, Hangzhou 310027, P.R. China.

[††]The author is with Centre for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Science, Shanghai 200000, P.R. China.

[†††]The author is with Ruijin Hospital, Shanghai Jiao Tong University School of Medicine, Shanghai 200000, P.R. China.

a) E-mail: zheminglu@zju.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2023EDL8055

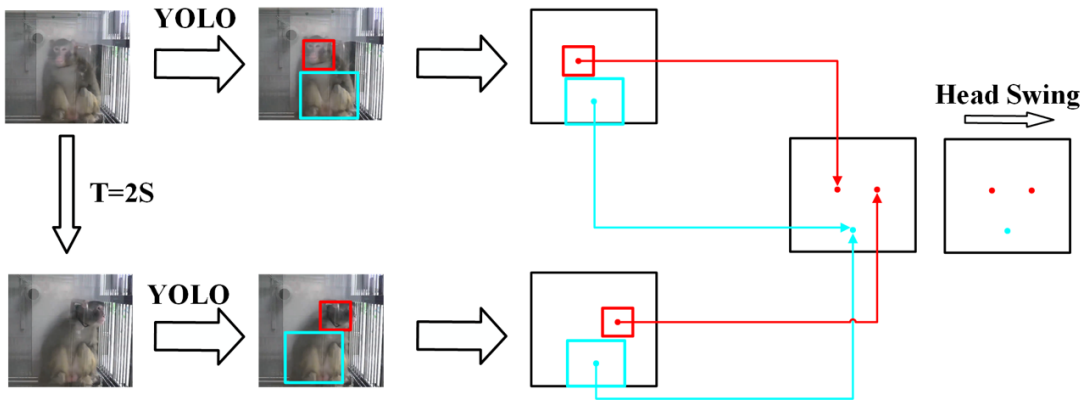

Fig. 1 The block diagram of our scheme. We compare the changes of the monkey's head position before and after 50 frames to determine whether the monkey swings its head.

along with deep learning, and some researchers have even adopted key point detection to monkeys [7]–[11]. Through the detection of key points of monkeys, we have been able to capture the movements of monkeys, but there is still no precise quantitative indicator. It's worthy noting that [16] identifies 14 different actions of monkey, which achieves a great performance monkey tracking. Differently, our work focuses on the monkey head swing and further extends to monkey swing counting in a more difficult scenery. In this Letter, we combine the monkey detection and morphology, and give the basis for accurately describing the monkey head swing.

2.2 Object Detection Algorithms

As one of the three major tasks in deep learning, object detection has always attracted a lot of attention. With the advent of the ResNet [12], deep learning has developed rapidly. We first consider the model R-CNN [3] that performs well in the object detection. This model still affects two important tasks in deep learning, object detection and segmentation. However, the model needs to be trained twice, which greatly increases the complexity of the model. To further simplify the training complexity of the model, SSD [4] and YOLO [6] have come out one after another. They get all the information including the target bounding box and the probability that the target belongs to different classes through one model. Due to its high accuracy and efficiency, the YOLO model [6], [13], [14] was soon widely recognized by the industry, and multiple versions were derived. In recent years, in order to describe the human pose, key point detection has become more and more popular [15]. However, the related technologies are not mature enough to deal with the complex scenarios in this project. Therefore, we finally adopt YOLO as the core algorithm.

3. Method

3.1 Monkey Detection

This Letter uses YOLO to detect the monkey's head as shown in Fig. 1. YOLO is a classic one-stage target detection algorithm. It is very different from R-CNN. R-CNN requires two steps in training. The model gets a lot of proposals in the first step and gets the detection results in the second step. YOLO, on the other hand, uses category and bounding box as a whole for regression. Therefore, YOLO is simpler than R-CNN.

YOLO adopts darknet53 as a backbone to extract the features of the input. The DarkNet53 is based on ResNet, which can maintain a stable gradient on the basis that the number of model layers can be continuously deepened. In order to reduce the spatial dimension of data features and speed up the training process, YOLO uses ResNet as the bottleneck to reduce the number of channels and reduce the burden of the model. Since large-scale features are more conducive to target detection tasks, YOLO is a multi-scale model for multi-channel detection. The low-level features complement the information lost during training. The multi-scale model also enriches the feature fusion of the model in training.

The main contribution of YOLO is the loss function. The model considers the classification loss and regression loss together. YOLO uses the L2 regression function to calculate the loss function of the BBox and uses the cross entropy to calculate the classification loss, including foreground and background. The accuracy of the target detection algorithm determines the accuracy of the monkey head swing counting.

3.2 Monkey Swing Recognition

The ideal situation of this project is to transplant the standard of manual counting directly into the algorithm. However, human subjective judgment is often difficult to form an effective algorithm criterion. This adds a lot of difficulty to the algorithm. This project uses a number of different dimensions to determine whether the monkey is swinging its head. The most direct criterion used by human when judging whether a monkey swings its head is the speed of swinging its head. Only when the speed is fast enough can we consider it as

an effective head swing. At the same time, the head swing occurs in a small period, and only the time when the action occurs is short enough can we consider it as an effective head swing. In addition, in order to exclude the interference of small perturbations, we also introduce the amplitude as a criterion for the model. The block diagram of our scheme is given in Fig. 1. The main idea is to compare the changes of the monkey's head position before and after 50 frames to determine if the monkey swings its head.

Therefore, we use speed, time, and distance as the criteria. Only within a certain period of time, a swing with a fast enough speed and a large enough swing can be considered as an effective swing.

4. Experiments

4.1 Dataset and Criterion

We collected monkey activity videos taken within one month as the dataset. We split the videos into frames and screened out frames with different actions as much as possible, forming a dataset of 50,000 pictures, and randomly selecting 5,000 of them as test data. We preprocess the dataset according to YOLO. The video collected in the project is in a monkey house without light (to avoid light interference to monkeys). In a dark environment, the monkey's skin and background color are very similar, which increases the detection difficulty.

Our test data includes two sets of videos. One of them was collected from a video of monkey activities during the day. We used the time of monkey's daily activities to shoot a 1-minute video every 10 minutes, and 50 videos at equal intervals. Another set was collected within three days. By taking a 1-minute video every 10 minutes, we took 320 videos at equal intervals. The test set with 50 videos contains frequent monkey activities, while the test set with 320 videos contains a large number of videos of monkeys in the silent phase. The training and testing datasets comprise data collected from three monkeys. The training set consists of 50,000 images captured from one monkey, while the test set comprises 50 videos from another monkey and 320 videos from a different monkey.

To enhance the accuracy of our results, we fine-tune several indicators as hyper-parameters. These hyper-parameters are derived from pixel-based measurements in the images. Specifically, we utilize 'Head Speed' and 'Body Speed' to represent the velocity of head and body swings within a specified time interval. Additionally, we employ 'Distance' and 'Time' as thresholds to determine the pixel distance and duration of a swing, respectively, for accurate identification of monkey swinging instances.

We have designed an algorithm to compute the counting accuracy of the algorithm. The actual number of head swings is m, and the count of algorithm is n. If $|m-n| \leq 2$, it is regarded as an accurate count, and if it exceeds 2, it is regarded as an error count. For videos with head swings within 12 times, $|m-n|-2 \leq 10$ (the number of fault tolerances is 10), the number of error counts divided by 10 is the error rate. For videos with more than 12 head swings, $|m-n|-2 > 10$, the number of error counts divided by the total number of fault tolerances is the error rate.

$$\text{score} = \begin{cases} 1 & ,|m-n| \leq 2 \\ 1 - \frac{(|m-n|-2)}{10} & ,|m-n|-2 \leq 10 \\ 1 - \frac{(|m-n|-2)}{|m-2|} & ,|m-n|-2 > 10 \end{cases} \quad (1)$$

Table 1 Comparison of the mAP between different object detection algorithms in monkey detection.

Algorithms	Backbone	input size	Training data	AP50	AP50:95
Faster R-CNN	Resnet50	640	Trainval set	95.3	76.4
SSD	VGG16	640	Trainval set	92.2	71.3
YOLOv3	Darknet53	640	Train set	97.8	75.2
YOLOv4	CSPdarknet53	640	Train set	99.3	80.5
YOLOv5s	Focus+CSP*5	640	Train set	99.6	81.1
YOLOv5m	Focus+CSP*5	640	Train set	99.6	82.4
YOLOv5l	Focus+CSP*5	640	Train set	99.6	81.5
YOLOv5s6	Focus+CSP*6	640	Train set	99.7	80.8
YOLOv5m6	Focus+CSP*6	640	Train set	99.7	81.7
YOLOv5l6	Focus+CSP*6	640	Train set	99.7	82.6

Table 2 Counting accuracy for both two test datasets. '50' and '320' of test denote the test datasets with 50 videos and 320 videos.

Method	Test	Head speed	Body speed	Distance	Time	Result
Ours	50	50	8	50	2	94.23%
	320	50	8	50	2	84.92%
SiamRPN	50	50	8	50	2	89.43%
	320	50	8	50	2	78.64%

4.2 Accuracy of Object Detection Algorithm

We tested the performance of three algorithms for monkey detection, including SSD, faster R-CNN, and YOLO. Considering the popularity of the YOLO family in the field of object detection, we tested the performance of YOLOv3, YOLOv4, and YOLOv5.

From the detection results, the latest YOLOv5 algorithm has the highest detection accuracy for the monkey head and body, and there is no obvious difference within the YOLOv5 series. After comprehensive analysis, we choose lightweight YOLOv5s6 as the main algorithm of this model.

4.3 Results of the Swing Counting Algorithm

As shown in Table 2, our model achieves an accuracy of 94.23% and 84.92% on 50 videos and 320 videos, respectively. Notably, our model outperforms the classical video tracking method, SiamRPN [18], by a margin of 4.80% in 50 videos and 6.28% in 320 videos. Furthermore, we conducted an extensive analysis of algorithmic counting accuracy across various scenes, aiming to establish a standardized criterion for monkey swing based on speed, amplitude, and distance across three dimensions.

Speed is an important index to evaluate whether monkeys swing their heads. Only when the head speed is fast enough can it be considered as an effective head swing. The speed is expressed by dividing the distance between ten frames before and after monkey swing by the number

Table 3 The effect of monkey head swing speed on counting accuracy

Test dataset	Head Speed	Result
50 videos	20	85.43
	30	88.72
	40	90.35
	50	**91.03**
	60	90.88
	70	90.56
	80	89.47
	90	87.52
	100	83.23
320 videos	30	75.63
	40	78.5
	50	**80.32**
	60	79.85
	70	78.34

Table 4 The effect of monkey body speed on counting accuracy, when head speed = 50

Test dataset	Body speed	Head speed	Result
50 videos	6	50	90.34
	7	50	92.77
	8	50	**93.91**
	9	50	93.17
	10	50	91.89
320 videos	6	50	81.85
	7	50	83.48
	8	50	**84.07**
	9	50	82.56
	10	50	80.99

Table 5 The effect of monkey head swing distance on counting accuracy, when head speed = 50, body speed = 8

Test dataset	Distance	Result
50 videos	25	93.98
	50	**94.17**
	75	92.70
320 videos	25	84.32
	50	**84.77**
	75	84.69

of frames, and the speed = distance/10. Considering the influence of monkey walking on head swing count, we also take body speed as an indicator. The results in Tables 3 and 4 show that when the monkey moves 50 pixels within 10 frames, it is an important basis for judging whether the monkey swings its head. Considering body movement, the maximum distance allowed for body movement is 8 pixels.

In order to further conform to the judgment of manual counting, we also take time and distance as the criteria to judge whether the head swings. Only in 2s, if the head swing amplitude exceeds 50 pixels, can it be considered as an effective head swing, and the experimental results in Tables 5 and 6 also support this.

4.4 Ablation Study about Score Computation

To further assess the efficacy of our approach, we have employed various score computation methods to evaluate our model. As illustrated in Table 7, we compare three computation methods across two datasets. The term 'Plain Accuracy'

Table 6 The effect of monkey head swing time on counting accuracy, when head speed = 50, body speed = 8, distance = 50

Test dataset	Time	Result
50 videos	1	94.21
	2	**94.23**
320 videos	1	84.87
	2	**84.92**

Table 7 Counting accuracy with two datasets in three settings

Allowed Error	Test dataset	Result
2	50 videos	94.23%
	320 videos	84.92%
0	50 videos	85.72%
	320 videos	77.81%
Plain Accuracy	50 videos	87.32%
	320 videos	79.67%

refers to the computation of the test using simple regression error. When the allowed error is set to 0, the counting accuracy significantly decreases to 85.93% and 77.81% in 50 videos and 320 videos, respectively. Conversely, when employing the plain accuracy computation method, the experimental results yield 87.32% and 79.67% in 50 videos and 320 videos, respectively. These findings clearly demonstrate that our method consistently achieves competitive results across different settings.

4.5 Compared with Skeleton-Based Methods

We have incorporated DeepLabCut [17] as a part of our methodology; however, the obtained experimental results were relatively inadequate. The outcome of our experiments is closely linked to the accuracy of the detection algorithm employed. The detection of monkey poses poses a significant challenge due to their abundance and complexity, necessitating the algorithm's robustness in capturing these poses. Our experimental findings indicate that DeepLabCut performs well in scenes with a distinct contrast between the foreground and background. However, given that both the monkey and background in our datasets exhibit low levels of illumination, skeleton-based methods exhibit poor performance. Furthermore, our observations reveal that the YOLO algorithm demonstrates superior robustness in detecting monkeys compared to the novel method employed.

4.6 Visualization

As shown in Fig. 2, we visualize the monkey swing process in two distinct test datasets, comprising 50 videos and 320 videos, respectively. Based on our findings, we draw the conclusion that our algorithm adeptly identifies the monkey's head and body, enabling precise counting of the number of monkey swings.

5. Conclusion

This Letter is an application-oriented paper on the counting of monkey head swings. We try to design an algorithm based

(a) 50 test videos

(b) 320 test videos

Fig. 2 The visualization of two test datasets with two different monkeys. (a) and (b) denote one swing process of monkey in 50 test videos and 320 test videos.

on object detection to detect monkey head swings without human participation. Ultimately, we achieved 94% accuracy on 50 videos. There is still a certain error in the monkey positioning using target detection. So we try to locate the monkey in a better way. For example, the key point detection can directly obtain the coordinates of the monkey. However, considering the accuracy requirements of this algorithm, the algorithm with the best stability may have the best detection effect.

Acknowledgments

This work is supported by the Shanghai Municipal Science and Technology Major Project, Grant No. 2018SHZDZX05.

References

[1] P. Kumar and M. Shingala, "Native monkey detection using deep convolution neural network," In: A. Hassanien, R. Bhatnagar, A. Darwish (eds), Advanced Machine Learning Technologies and Applications (AMLTA 2020), Advances in Intelligent Systems and Computing, vol.1141, pp.373–383, Springer, Singapore, 2020.

[2] H. Zhu, H. Wei, B. Li, X. Yuan, and N. Kehtarnavaz, "A review of video object detection: datasets, metrics and methods," Applied Sciences, vol.10, no.21, Article no.7834, 2020.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.580–587, 2014.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C. Berg, "SSD: Single Shot MultiBox Detector," ECCV 2016, Part I, vol.9905, pp.21–37, in B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., LNCS, Cham: Springer, 2016.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol.39, no.6, pp.1137–1149, 2017.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," IEEE CVPR, Las Vegas, NV, USA, pp.779–788, 2016.

[7] S. Agezo and G.J. Berman, "Tracking together: estimating social poses," Nature Methods, vol.19, pp.410–411, 2022.

[8] J.M. Graving, D. Chae, H. Naik, L. Li, B. Koger, B.R. Costelloe, and I.D. Couzin. DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning, Elife, 8:e47994, 2019.

[9] T.D. Pereira, D.E. Aldarondo, L. Willmore, M. Kislin, S.S.-H. Wang, M. Murthy, and J.W. Shaevitz, "Fast animal pose estimation using deep neural networks," Nature Methods, vol.16, pp.117–125, 2019.

[10] A. Mathis, P. Mamidanna, K.M. Cury, T. Abe, V.N. Murthy, M.W. Mathis, and M. Bethge, "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," Nature Neuroscience, vol.21, pp.1281–1289, 2018.

[11] P.C. Bala, B.R. Eisenreich, S.B.M. Yoo, B.Y. Hayden, H.S. Park, and J. Zimmermann, "Automated markerless pose estimation in freely moving macaques with OpenMonkeyStudio," Nature Communications, vol.11, Article no.4560, 2020.

[12] K. He, X. Zhang, S. Ren, and J. Sun "Deep residual learning for image recognition," IEEE CVPR, Las Vegas, NV, USA, pp.770–778, 2016.

[13] C. Hao and Z.-M. Lu, "Contraband detection based on deep learning," Journal of Information Hiding and Multimedia Signal Processing, vol.13. no.3, pp.165–177, 2022.

[14] H. Chen and Z.-M. Lu. "Dynamic Smoke Detection by Eliminating Static Targets in Video," International Journal of Innovative Computing, Information and Control, vol.19, no.2, 2023, doi: 10.24507/ijicic.19.02.355.

[15] A.S. Eltanany, M.S. Elwan, and A.S. Amein, "Key point detection techniques," International Conference on Advanced Intelligent Systems and Informatics, Springer, Cham, pp.901–911, 2019.

[16] Li, Chuxi, et al. "Deep learning-based activity recognition and fine motor identification using 2D skeletons of cynomolgus monkeys," Zoological Research, vol.44, no.5, p.967, 2023.

[17] A. Mathis, P. Mamidanna, K.M. Cury, T. Abe, V.N. Murthy, M.W. Mathis, and M. Bethge, "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," Nature neuroscience, vol.21, no.9, pp.1281–1289, 2018.

[18] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.