Human Behavior Recognition Algorithm Based on HD-C3D Model

Zhihao Xie[1], Lei Yu[2], Qi Wang[1], and Ziji Ma[1](✉)

[1] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
zijima@hnu.edu.cn

[2] Information Institute of Ministry of Emergency Management of the People's Republic of China, Beijing 100029, China

Abstract. To address the problems of low recognition accuracy and long training time of the original C3D (Convolutional 3D) model, this paper proposes a modified method to improve its framework. Firstly, the Relu activation function in the hidden layer is replaced by the Hardswish function to allow more neurons to participate in parameter updating and to alleviate the problem of slow gradient convergence. Secondly, the dataset was optimised using the background difference method and the image scaling improvement respectively, and the optimised dataset was used for model training. The image scaling improvement combined with the activation function improvement results in a better HDs-C3D (Hardswish Data scaling - Convolutional 3D) model. Its accuracy on the training dataset reached 89.1%; meanwhile, the training time per round was reduced by about 25% when trained in the experimental environment of this paper.

Keywords: C3D model · activation function · background difference method · image scaling improvement

1 Introduction

The recognition of human behavior in video is one of the most popular aspects of video understanding. The scientific results generated by human behavior recognition technology are now being used in various aspects of life.

Nowadays, traditional human behavior recognition algorithms based on feature extraction are gradually replaced by algorithms based on deep learning. Tran et al. [1] proposed a C3D convolutional neural network, which differs from previous 2D convolutional neural networks in that both convolution and pooling operations are implemented using 3D kernels. Ye et al. [2] added DenseNet from the 3D convolutional layer to

This work is supported in part by the National Nature Science Foundation of China under Grant 61971182, in part by Nature Science Foundation of Hunan Province 2021JJ30145, in part by Hunan Province Enterprise Science and technology commis-sioner program 2021GK5021, in part by Guangxi key R&D plan project 2022AB41020

C. Wu et al. (Eds.): MONAMI 2023, LNICST 559, pp. 85–93, 2024.
https://doi.org/10.1007/978-3-031-55471-1_7

improve the feature utilization in the network, and finally verified its effectiveness in recognition. Niu et al. [3] combined the C3D model with an SVM classifier and L2 regularisation to improve the running speed of the algorithm and prevent model over-fitting. Liu et al. [4] proposed the CoDT method for dealing with domain sharing and target-specific feature co-clustering for cross-domain 3D action recognition. Si et al. [5] proposed adversarial self-supervised learning for 3D action recognition. Singh et al. [6] proposed a two-path temporal comparison model to address video behavior recognition with few labels. Yang et al. [7] proposed a temporal pyramid network to capture action instances at different tempos. Hong et al. [8] proposed Video Pose Distillation. (VPD) to improve the performance of few-shot, fine-grained action recognition, retrieval, and detection tasks. Li et al. [9] proposed a Channel Independent Directional Convolution (CIDC) structure that allows the network to focus on the more meaningful and behaviorally relevant parts of the frame. Jiang et al. [10] proposed a new strategy of aggregated decentralized down-sampling to prevent the loss of feature information. Yuan et al. [11] proposed Dynamic Inference Network (DIN) network for group activity recognition, which reduces the computational overhead of the inference module. Sudhakaran et al. [12] proposed Gate Shift Module (GSM) module, which reduces the complexity of the model with the use of this network. Yang et al. [13] proposed the Recurrent Vision Transformer (RViT) framework, which can correctly process video clips of various lengths without consuming large amounts of GPU memory. Li et al. [14] proposed the Dynamic Spatio-Temporal Specialization (DSTS) module, which yields advanced performance in fine-grained action recognition. Liu et al. [15] proposed cascade vision detection. Huang et al. [16] proposed the RoofSplit framework to enable efficient execution of deep learning models at the edge of the network.

In order to improve the recognition accuracy and shorten the training time of the C3D model, the improved method proposed in this paper has the following effects. Replacing the Relu function with the Hardswish function enables more neurons to be updated with parameters, while improving the gradient convergence speed to a certain extent. The use of background difference method can get the foreground action of changes in the dataset, there is easy feature extraction. Improvements to the image scaling operation during model training can shorten the training time as well as improve recognition accuracy.

2 Human Behavior Recognition Based on C3D Model

The convolution operation in the C3D model with a convolution kernel of $3 \times 3 \times 3$ size can get better experimental results. The $3 \times 3 \times 3$ sized convolution kernel in the model is sliding in $1 \times 1 \times 1$ steps throughout the convolution layer, while the 3D pooling kernel is not. In order to preserve the initial timing information at the time of video frame input, the pooling kernels and sliding steps of the first pooling layer are set to $1 \times 2 \times 2$, while the pooling kernels and sliding steps of the remaining pooling layers are set to $2 \times 2 \times 2$ (Fig. 1).

The network contains 8 convolutional layers, 5 pooling layers, 2 fully connected layers and a Softmax classifier. The data size is denoted as $c \times l \times h \times w$, where c is the number of channels of video frames, l is the number of video frames, h is the height

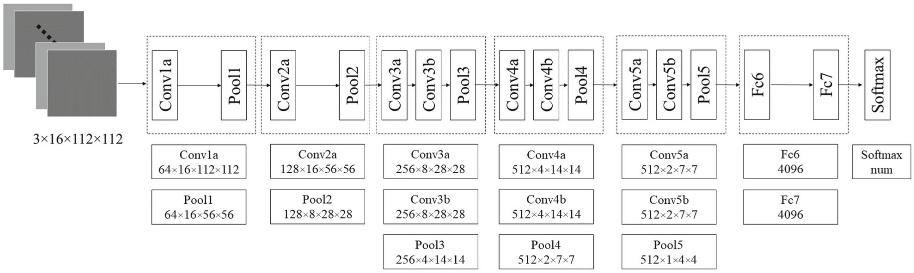

Fig. 1. Structure of C3D convolutional neural network. This structural diagram shows the size changes of the input data after passing through various parts of the model.

of video frames, and w is the width of video frames. The data size of the input network is $3 \times 16 \times 112 \times 112$ in size.

The 3D convolution kernel can be represented by $d \times k \times k$, with d being the time depth and k being the spatial size of the convolution kernel. The image data is processed with a quantity of n, kernel size $3 \times 3 \times 3$, and sliding step and padding of both $1 \times 1 \times 1$, and the output feature map size is $n \times l \times h \times w$. Optimizer uses SGD (Stochastic Gradient Descent) algorithm, Relu as the activation function in the hidden layer, and Dropout regularization to prevent overfitting.

3 Improvements to This Paper

3.1 Optimisation of Activation Functions

The activation function used in the original C3D convolutional neural network is Relu, which has the following drawbacks: (i) As the initial parameters are randomly assigned, when the parameter value is negative, then its output value and gradient are zero, resulting in a break in back propagation using chain derivation, and thus the corresponding parameters cannot continue to be updated down the line (ii) The function has no negative output, which can slow down the convergence speed of the gradient and easily enter local optima.

To relieve the above problem, the Hardswish [17] function is applied to this paper instead of the Relu function, which does not increase the computational complexity significantly. The expression for this function is shown in Eq. (1):

$$Hardwish(x) = \begin{cases} 0 & x \leq -3 \\ \frac{x \cdot (x+3)}{6} & -3 < x < 3 \\ x & x \geq 3 \end{cases} \tag{1}$$

The Hardswish function has the following advantages over the Relu function: (i) The function has an output value and gradient of 0 at $x < -3$, with a wider range of valid values than the Relu function, enabling more neurons to be involved in parameter updating; (ii) The function has negative values in the (−3,0) interval, which can alleviate the problem that the gradient convergence rate becomes slower and easier to enter the local optimum (Fig. 2).

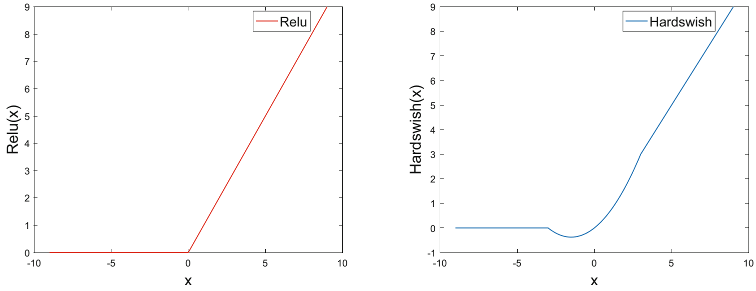

Fig. 2. Relu activation function and Hardswish activation function.

3.2 Background Difference Method Optimisation of Dataset

The background difference method [18] is commonly used in the field of target detection. The basic principle is that a frame is selected as the background frame and the current frame is subtracted from the background frame. The region where the pixel difference between the current frame and the background frame exceeds a certain threshold is used as the foreground to obtain the size contour of the moving body. The process of implementation is as follows (Table 1):

Table 1. Algorithm 1 the pseudocodes of background difference method.

Algorithm 1: Background Difference Method

Input: Current frame F_t , Background frame B, Binarization threshold T.
Output: Background differential frame D_t.

$F_t \leftarrow$ *Grayscale processing*
$B \leftarrow$ *Grayscale processing*
\# Add *Gaussian blur* to F_t ,B
$F_t , B \leftarrow G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$
$D_t = |F_t(x,y) - B(x,y)|$
\# Add *Binarization process* to D_t
if $D_t(x,y) > T$
    $D_t(x,y) = 255$
else
    $D_t(x,y) = 25$
$D_t \leftarrow$ *Morphological Dilation*
return D_t

In this paper, the dataset is processed using the background difference method, and the new data after processing is uniformly cropped to a size of 106 × 106. The network structure is also modified for model training, and the data flow under the new structure is shown below (Fig. 3).

Table 2. Simulation environment

Hardware	Software environment
Operating system: Ubuntu 18.04.6 LTS CPU: Intel Core i7-10700K Disk: 512 GB GPU graphics card: RXT3080 GPU memory: 12G	Python version: 3.8 Pytorch version: 1.8 CUDA Version: 11.4 Torchvision version: 0.9.0 OpenCV version: 4.1.2

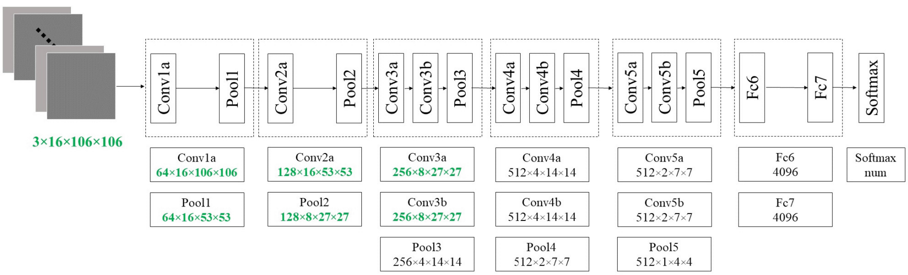

Fig. 3. Model structure and data flow under the background difference method

3.3 Image Scaling Improvements

The UCF-101 dataset is chosen for this experiment, and the video frames undergo two scale transformations before being trained. In this paper, the process is improved by first scaling the 171 × 128 size to 106 × 106 size before scaling to 112 × 112 size. This is shown in Fig. 4 below.

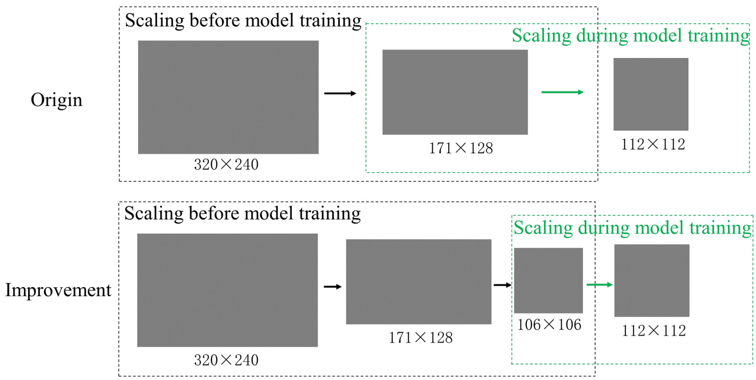

Fig. 4. Comparison of improved image scaling operations

As obtaining a certain size of image, equal scaling produces clearer images than non-equal scaling, resulting in less image data loss. Also, the scaling of the image from

171 × 128 to 112 × 112 size is done during the training of the model. When the image to be scaled is close to the target size, the time taken to run the image scaling function can be reduced. Therefore, this time, the image of 171 × 128 size is first scaled to 106 × 106 size to improve the image quality and shorten the model training time. The reason for using shrink then zoom for the images is that (i) it makes the dataset more lightweight (ii) scaling down and then scaling up allows for pixel filling of the images.

4 Experiments and Analysis of Results

4.1 Improved Model HD-C3D

In this paper, based on the original C3D model, the HD-C3D (Hardswish Data optimization - Convolutional 3D) model is proposed by combining three improvement points, which consists of two parts: the HDb-C3D (Hardswish Data background difference - Convolutional 3D) model and the HDs-C3D model. The HDb-C3D model is derived from the Hardswish activation function combined with the background difference method; the HDs-C3D model is derived from the Hardswish activation function combined with image scaling. We unify the HDb-C3D model and HDs-C3D model into HD-C3D model (Fig. 5).

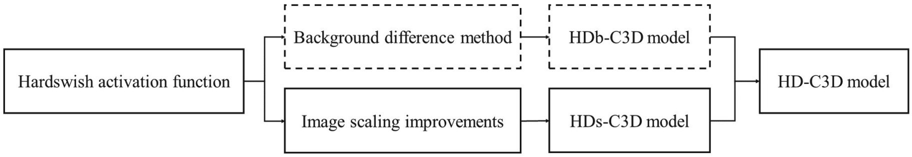

Fig. 5. Structure of the HD-C3D model.

4.2 Simulation Environment

(1) Data set: This paper uses the UCF-101 dataset, which has a total of 13,320 videos all with a resolution of 320 × 240 and a total duration of approximately 27 h. There are 101 categories encompassing the 5 main types of human movement.
(2) Experimental setting: The specific hardware and software environment is shown in Table 2.

4.3 Analysis of Results

In this experiment, the HDb-C3D model and HDs-C3D model are compared with the original C3D model for data comparison. For the experiments, the UCF-101 dataset is used for training, and the number of epochs is 50.

The accuracy curves and loss curves of the model training are shown in Figs. 6 and 7. As the number of training rounds increased, the accuracy of the model increased and the loss value decreased, and levelled off at around 25 rounds. Finally, the accuracy of HDb-C3D model in this training can reach 85.4%, and the accuracy of HDs-C3D model in this training can reach 89.1%, and the accuracy of the original C3D model in this experiment is 50.2%. The accuracy of HDb-C3D model is improved about 35% compared with the original model, and the accuracy of HDs-C3D model is improved about 39% compared with the original model. Meanwhile, the loss reduction of HDb-C3D and HDs-C3D over

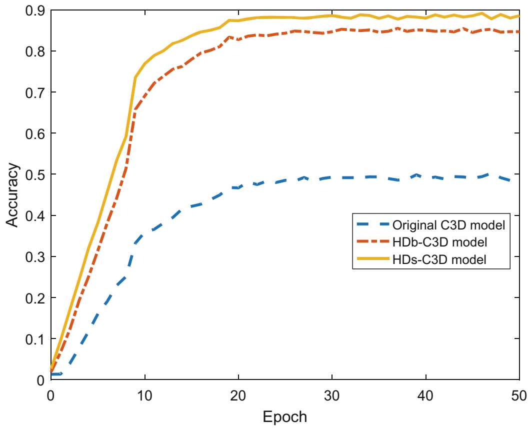

Fig. 6. Variation curve of model training accuracy.

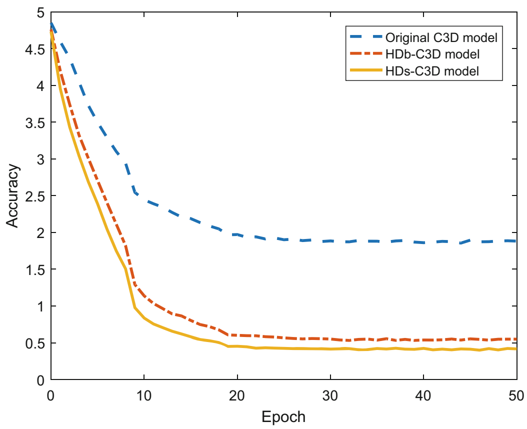

Fig. 7. Variation curve of model training loss.

the original C3D model is 1.3 and 1.4, respectively. It can be seen that both improved models have good results in improving accuracy and reducing loss.

In addition, the new dataset obtained using background difference method reduces the storage space by approximately 3.7 GB compared to the original dataset, while the detection time can be reduced by approximately half compared to the original detection time when using the new video set composed. Without the image scaling improvement, the training dataset takes about 8 min to train once, and with it the training takes about 6 min, a 25% reduction in training time.

5 Summary

In this paper, three C3D model improvement methods are proposed and combined into two improved models. The recognition accuracy of HDb-C3D model and HDs-C3D model are improved by about 35% and 39% respectively compared with the original C3D model; the loss is reduced by 1.3 and 1.4 respectively; meanwhile the training time per round of HDs-C3D model is shortened by about 25%.

Although this paper has improved the original C3D model, it has mostly improved the data pre-processing and the study is not comprehensive enough. For this reason, subsequent work could be directed towards improving the model architecture in the following directions:

(1) Adding an attention mechanism to the network. This mechanism allows the network to pay more attention to the behavioral actions of interest, thus improving recognition accuracy.
(2) The introduction of a residual network. The addition of this network allows the model to learn higher level features without gradient disappearance, thus improving the accuracy of recognition.

References

1. Tran, D., Bourdev, L., Fergus, R., et al.: Learning spatiotemporal features with 3D convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4489–4497 (2015)
2. Ye, Q., Yang, H.: Design of human behavior recognition network based on deep learning. China Sci. Technol. Inf. **10**(628), 91–94 (2020)
3. Niu, Y., Su, W., Yu, C., et al.: Real-time behavior recognition for intelligent surveillance based on TX2 environment. Inf. Technol. Informatization **4**(253), 243–245 (2021)
4. Liu, Q., Wang, Z.: collaborating domain-shared and target-specific feature clustering for cross-domain 3D action recognition. In: In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13664, pp. 137–155. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19772-7_9
5. Si, C., Nie, X., Wang, W., et al.: Adversarial Self-Supervised Learning for Semi-Supervised 3D Action Recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, JM. (eds.) ECCV 2020. LNCS, vol. 12352, pp. 35–51. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58571-6_3

6. Singh, A., Chakraborty, O., Varshney, A., et al.: Semi-supervised action recognition with temporal contrastive learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10389–10399 (2021)
7. Yang, C., Xu, Y., Shi, J., et al.: Temporal pyramid network for action recognition 2020. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 588–597 (2020)
8. Hong, J., Fisher, M., Gharbi, M., et al.: Video Pose Distillation for Few-Shot, Fine-Grained Sports Action Recognition. 2021, in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2021, pp. 9234–9243
9. Li, X., Shuai, B., Tighe, J.: Directional temporal modeling for action recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12351, pp. 275–291. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58539-6_17
10. Jiang, Z., Ma, Z., Wang, Y., et al.: Aggregated decentralized down-sampling-based ResNet for smart healthcare systems. Neural Comput. Appl. **75**, 1–13 (2021)
11. Yuan, H., Ni, D., Wang, M.: Spatio-temporal dynamic inference network for group activity recognition. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7456–7465 (2021)
12. Sudhakaran, S., Escalera, S., Lanz, O.: Gate-shift networks for video action recognition. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1099–1108 (2020)
13. Yang, J., Dong, X., Liu, L., et al.: Recurring the transformer for video action recognition. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14043–14053 (2022)
14. Li, T., Foo, L., Ke, Q., et al.: Dynamic spatio-temporal specialization learning for fine-grained action recognition. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13664, pp. 386–403. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19772-7_23
15. Liu, J., Liu, H., Chakraborty, C., et al.: Cascade learning embedded vision inspection of rail fastener by using a fault detection IoT vehicle. IEEE Internet Things J. **10**(4), 3006–3017 (2021)
16. Huang, Y., Zhang, H., Shao, X., et al.: RoofSplit: an edge computing framework with heterogeneous nodes collaboration considering optimal CNN model splitting. Future Gener. Comput. Syst. **140**, 79–90 (2023)
17. Duta, I.C., Liu, L, Zhu F, et al.: Improved residual networks for image and video recognition. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9415–9422 (2021)
18. Rosin, P., Ellis, T.: Image difference threshold strategies and shadow detection. In: 1995 British Machine Vision Conference, pp. 347–356 (1995)