IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 4, 2020, accepted August 12, 2020, date of publication August 17, 2020, date of current version August 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017076

Human Action Recognition Algorithm Based on Multi-feature Map Fusion

HAOFEI WANG AND JUNFENG LI
Department of Control Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310000, China
Corresponding author: Junfeng Li (ljf2003@zstu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61374022.

ABSTRACT The emergence of the convolutional neural network greatly improves the accuracy of human action recognition. However, with the deepening of the network, fewer and fewer features are extracted, and in some datasets, due to the shooting angle, the size of the target to be recognized is different. To solve this problem, on the basis of resnext human action recognition method, we propose an improved resnext human action recognition method based on multi-feature map fusion. First, the video is uniformly sampled to generate training samples, and we generate samples with different frames as the input to the network. Second, we add n layers of up-sampling layers after layer 1 of resnext, to enlarge the feature maps and extract multiple feature maps, so that the extracted feature maps are clearer, and small targets can be better recognized. Finally, for the n results obtained, we use the weighted geometric means combination forecasting method based on L_1 norm to fuse and obtain the final result. In the process of experiment, using UCF-101 and HMDB-51 for verification, the accuracy of our model is 90.3% on UCF-101, which is higher than most of the state-of-art algorithms.

INDEX TERMS Human action recognition, resnext-101 network, up-sampling method, weight fusion.

I. INTRODUCTION

Due to the potential applications of human action recognition in video surveillance, behavior analysis, video retrieval, and other fields, human action recognition has become a very important field in computer vision research [1]. Human action recognition refers to the video sequence of human action, through the detection, classification, and tracking of moving targets, analysis and recognition of human action, and description in natural language [2]. In the real world, human action recognition plays a basic role in video analysis.

Early human action recognition was based on hand-crafted features. These features are more dependent on databases, they performed well on some databases, however, these features are not necessarily applicable to other databases. Moreover, hand-crafted features take a long time, which is not conducive to feature extraction of large databases.

With the rise of deep learning, the learning of automatic feature engineering has solved the shortcomings of hand-crafted feature engineering and made significant progress in the field of human action recognition. However, because of the long information period, a large amount of redundant information, complex backgrounds, and the diversity of viewpoints [3], there are still many problems to be solved in human action recognition methods based on deep learning.

In the context of big data, human action recognition has broader application scenarios, including video recommendation, monitoring analysis, human-computer interaction, etc. However, although the current algorithm has achieved good performance, it still needs to be improved in accuracy and running speed. Therefore, in order to ensure accuracy and improve the running speed, the human action recognition algorithm is still the focus and difficulty of current research.

II. RELATED WORK

A. FEATURE-BASED APPROACHES

Some classic image feature extraction methods are generalized to video [4], traditional human action recognition feature extraction methods include SIFT (Scale-invariant feature transform) relying on prior knowledge and HOG (Histogram of Oriented Gradient); SIFT improved SURF (Speeded-Up Robust Features) [5] algorithm and 3D-SIFT [6] algorithm; HOG3D [7] algorithm, etc.

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues.

The detection operator of the traditional feature extraction algorithm is artificially designed and obtained by a large amount of prior knowledge. Therefore, the traditional algorithm is time-consuming and the workload is heavy. With the advent of deep learning, more and more studies are affected by the significant achievements of convolutional neural networks in static image recognition. In action recognition, the use of deep models to train end-to-end networks has clearly exceeded hand-crafted features [8].

B. CNN BASED APPROACHES

Human action recognition methods based on convolutional neural network architecture are roughly divided into 2D convolutional neural networks and 3D convolutional neural networks [9]. The 2D convolutional neural network achieves good performance in static image recognition [10], [11]. It is easy to apply a 2D convolutional neural network to video representation to extract features, but ignore the relationship between video frames. For this reason, Ji *et al.* [12] proposed a 3D convolutional neural network method. A 3D convolutional neural network consists of a 2D spatial dimension and a temporal dimension, 3D convolutional neural networks consider time information, but have too many parameters. Compared to 2D convolutional neural networks, 3D convolutional neural networks are more difficult to learn [13].

2D two-stream network architecture [14]. This method divides the video into two parts: the temporal domain and the spatial domain. The obtained RGB images and optical flow frames are input into the network to perform feature learning in temporal and spatial domains. Finally, the prediction is performed only by averaging the classification scores, and only the convolution operation between adjacent frames cannot obtain long-term information. Based on the problems existing in the 2D two-stream network architecture, it was improved and a 3D two-stream network architecture was proposed. The I3D proposed by Carreira and Zisserman [15] increases the length of the input video clip to obtain a longer range of information, but it is computationally intensive and cannot handle a longer range of video. Based on I3D, Wang [16] proposed a non-local neural network, which uses the spatiotemporal non-local relationship in the video. Xu *et al.* [17] and Qiu *et al.* [18] proposed a coding method to obtain the video-level representation but ignored the connection between frames. TSN proposed by Wang *et al.* [19] adopts the sparse and global sampling method to sample a fixed number of frames to cover the time sequence structure of a long-time range so that the entire length of a video is not considered and the fusion is carried out at the end. Different from the above-mentioned spatio-temporal features, [20] encodes spatio-temporal features by imposing a weight sharing constraint on the learnable parameters so that practice and spatial features can benefit from each other through collaborative learning. The above-mentioned spatio-temporal fusion methods are time-consuming and expensive for network training. In order to solve this problem, Zhou *et al.* [21] proposed a new method to embed the spatiotemporal fusion strategy into a pre-defined probability space so that any multiple fusion strategies can be evaluated at the network level without having to train them separately, which greatly improves the strategy for spatiotemporal fusion Analysis efficiency.

Inspired by FPN [22], we propose a multi-scale fusion method for human action recognition; besides, by observing the datasets, we found that the background information of some actions is complex, and the targets we want to recognize are small relative to the entire background, so we use the up-sampling method to enlarge the feature maps to make small targets clearer and easier to detect. Moreover, deep learning is learned by learning the extracted features, the more detailed and clear the extracted features, the better the learning effect. Therefore, on the basis of resnext, we propose the method of Human Action Recognition Algorithm Based on Multi-feature Map Fusion, adding n layers of up-sampling layers after layer1 to train separately, which aims to enlarge the feature maps to make the extracted features clearer. In order to solve the problem of obtaining the final result only by the average classification scores in the 2D two-stream network architecture [14], we propose to use the weighted geometric means combination forecasting method based on L_1 norm to fuse the obtained n results.

III. RESNEXT-101

We first introduce the architecture of resnext. The traditional way to improve the accuracy of the model is to deepen or widen the network, but as the number of hyperparameters increases (such as the number of channels, filter size, etc.), the difficulty of network design and the computational overhead also increases. Therefore, the proposed resnext structure can improve the accuracy without increasing the complexity of the parameters, while reducing the number of hyperparameters.

Xie *et al.* [23], proposed the network resnext, adopting the idea of VGG stack and Inception's split-transform-merge at the same time, but with strong expansibility, which can be considered as increasing the accuracy while not changing or reducing the complexity of the model. There is a noun called cardinality, with the mean of the size of the set of transformations. Experiments demonstrate that increasing cardinality is a more effective way of gaining accuracy than going deeper or wider [23]. Fig. 1 shows a block of resnext.

Based on the block of resnext, the internal structure of resnext-101 is shown in Table 1.

Shallow features contain detailed information, deep features contain semantic information, semantic information can help to accurately detect the target. However, as the network [23] deepens, more useful features are filtered out. And in the dataset, the size of the target to be recognized is different. Therefore, we add n up-sampling layers after layer 1 of resnext-101 to amplify the extracted features, so that the network extracts more detailed features and small targets are better recognized.

Our work is different from the previous approaches in two main aspects [24]: (1) Based on resnext-101, we add

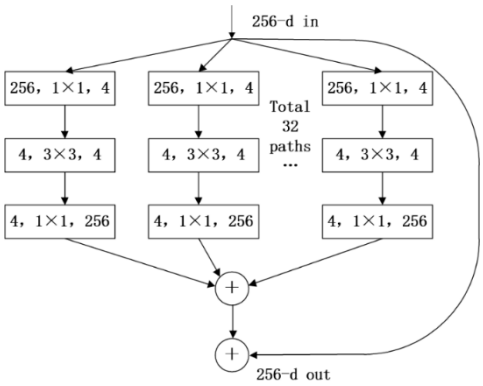

FIGURE 1. A block of resnext with cardinality = 32.

TABLE 1. Units for magnetic properties.

Stage	Output	Resnext-101 [a]
Conv1	112×112	7×7, 64, stride=2
		3×3, max pool, stride=2
Conv2	56×56	1×1,128 3×3,128,C=32 ×3
		1×1,256
Conv3	28×28	1×1,256 3×3,256,C=32 ×4
		1×1,512
Conv4	14×14	1×1,512 3×3,512,C=32 ×23
		1×1,1024
Conv5	7×7	1×1,1024 3×3,1024,C=32 ×3
		1×1,2048
	1×1	Global average pool Fc, softmax

several up-sampling layers [25] to extract more feature maps; (2) Several groups of results obtained are fused using the weighted geometric means combination forecasting method based on L_1 norm to get the final result.

IV. HUMAN ACTION RECOGNITION ALGORITHM BASED ON MULTI-FEATURE MAP FUSION

A. NETWORK ARCHITECTURE

The input image of the neural network input layer is convolved with the convolution kernel to obtain the feature map. A feature map is a description of the characteristics of the image, the more features extracted and the more detailed, the identification effect is better. Therefore, the more feature maps, the more representative the extracted features will be, and the better the recognition effect will be. The shallow features show more detailed information, while the deep features contain more semantic information, which can help accurately detect the target. However, after multi-layer convolution, many features have been filtered out, so on the basis of the resnext-101, we proposed a new architecture named Human Action Recognition Algorithm Based on Multi-feature Map Fusion. N layers of up-sampling are added after layer 1 of resnext-101 for prediction.

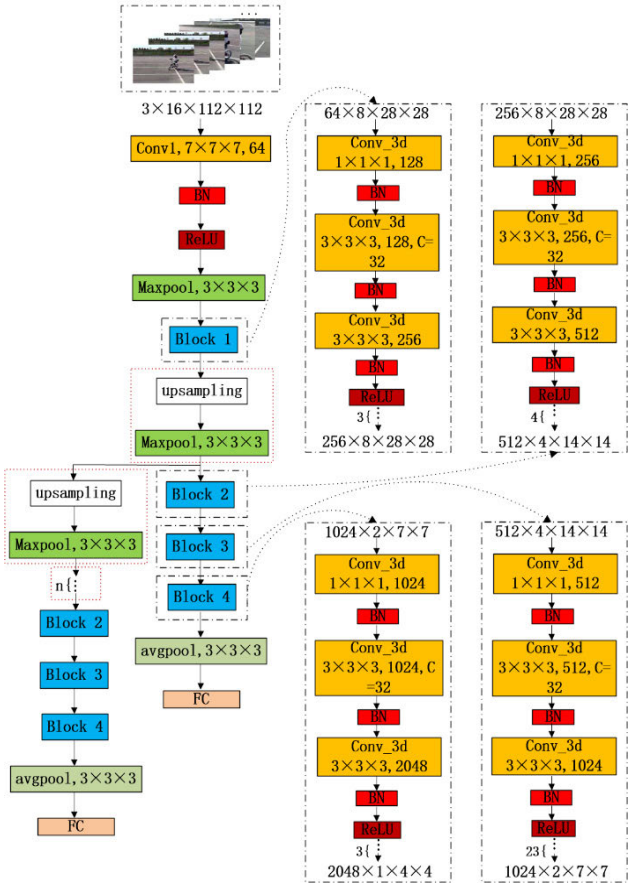

FIGURE 2. The architecture of our network.

Fig. 2 is the architecture of our network. First, the sampling method is uniform sampling, and the default size of each sample is 3 channels × 16 frames × 112 pixels × 112 pixels. Second, we use stochastic gradient descent to train the network and get n prediction results. We calculate the weights of the training results through the above model and fuse the test results according to the weights to get the final results.

B. UP-SAMPLING METHOD

On the basis of resnext-101, we separately add n up-sampling layers to extract more features. To make this more concrete, we now discuss several ways of up-sampling methods:

1) NEAREST NEIGHBOR INTERPOLATION

The simplest interpolation method, we obtain the coordinate (srcX, srcY) of the source image corresponding to (dstX, dstY) by (1), and fill in the corresponding pixel value.

$$\begin{cases} srcX = dstX \times (scrWidth/dstWidth) \\ srcY = dstY \times (scrHeight/dstHeight) \end{cases} \quad (1)$$

(srcWidth, srcHeight) indicates the width and height of the source image, (dstWidth, dstHeight) indicates the width and height of the image after interpolation.

2) BILINEAR INTERPOLATION

This method is to calculate the pixel value of point P(x, y) according to the pixel values of the nearest four points of point P(x, y). The core idea is to perform linear interpolation in two directions respectively. Q_{11}, Q_{12}, Q_{21}, Q_{22} pixel values are known, we first use (2) to calculate the pixel values of R_1, and R_2. Then, we calculate the pixel value of P using (3). We substitute (2) into (3) to get (4).

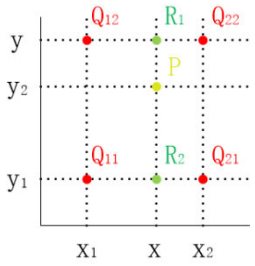

FIGURE 3. Bilinear interpolation.

x direction:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$
$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad (2)$$

y direction:

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \quad (3)$$

So the pixel value at point P is:

$$\begin{aligned} f(x, y) \approx & \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y_2 - y) \\ & + \frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y_2 - y) \\ & + \frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y - y_1) \\ & + \frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y - y_1) \end{aligned} \quad (4)$$

3) TRILINEAR INTERPOLATION

Trilinear interpolation operation in n = 1, three-dimensional D = 3 parameter space, so that 8 points adjacent to the point to be interpolated are needed.

On a periodic cube grid, let x_d, y_d, z_d be the differences between each of x, y, z, and the smaller coordinate related, that is:

$$x_d = \frac{x - x_0}{x_1 - x_0}$$
$$y_d = \frac{y - y_0}{y_1 - y_0}$$
$$z_d = \frac{z - z_0}{z_1 - z_0} \quad (5)$$

where x_0 is the point below x and x_1 is the point above x, y_0, y_1, z_0, z_1 are the same. $f_{000}, f_{001}, f_{010}, f_{011}, f_{100}, f_{101}, f_{110}, f_{111}$ pixel values are known, first, we calculate the pixel values of R_1, R_2, R_3, R_4 using (6). Then we use (7) to calculate the pixel values of r_1, r_2. Finally, we calculate the pixel value of using (8).

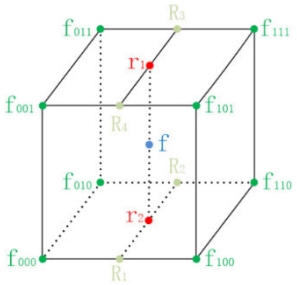

FIGURE 4. Trilinear interpolation.

First, interpolate in the x direction:

$$\begin{aligned} R_1 &= f_{000}(1 - x_d) + f_{100} x_d \\ R_2 &= f_{010}(1 - x_d) + f_{110} x_d \\ R_3 &= f_{011}(1 - x_d) + f_{111} x_d \\ R_4 &= f_{001}(1 - x_d) + f_{101} x_d \end{aligned} \quad (6)$$

Then interpolate in the y direction:

$$\begin{aligned} r_1 &= R_4(1 - y_d) + R_3 y_d \\ r_2 &= R_1(1 - y_d) + R_2 y_d \end{aligned} \quad (7)$$

Finally, interpolate in the z direction:

$$f = r_2(1 - z_d) + r_1 z_d \quad (8)$$

The nearest neighbor interpolation method will cause a discontinuity in the grayscale of the generated image. When the feature map is enlarged, this method directly uses the nearest pixel to generate a new pixel, so where the grayscale changes, there is obvious jagged; The calculation of the bilinear interpolation method is complicated, and the amount of calculation is large, but the calculation result of the four pixels used by this method greatly eliminates the phenomenon of jaggedness and has no disadvantages of grayscale discontinuity. However, the bilinear interpolation has the characteristic of low-pass filtering, so that high-frequency components are damaged, so the edges will become blurred; The trilinear interpolation method overcomes the shortcomings of the above two methods, with high calculation accuracy and better effect. Therefore, we choose the trilinear interpolation as the up-sampling method.

C. FUSE METHOD

We train these networks separately to obtain different training results, the weights of the training results are obtained by the method of [26], when evaluating the network, the weights

obtained by [26] are used to fuse the obtained results to obtain the final result. The combined prediction model of [26] can be expressed as:

$$min\ F(L) = \sum_{t=1}^{N} |\sum_{t=1}^{n} l_i e_{it}|$$
$$e_{it} = lnx_t - lnx_{it}$$

$$\text{s.t} \begin{cases} \sum_{i=1}^{m} l_i = 1 \\ l_i \geq 0, \quad \dot{l} = 1, 2, \ldots, \text{n} \end{cases} \tag{9}$$

Among them, F(L) is the logarithmic error based on the L1 norm between the combined prediction method of the weighted geometric average and the actual value of the index sequence, e_{it} represents the logarithmic error between the predicted value x_{it} and the actual value x_t at time t of the i-th prediction method. The smaller F(L) is, the closer the combined prediction method of weighted geometric mean is to the actual value of the index series, thus the more accurate and effective it will be.

V. EXPERIMENT

A. DATASET

UCF-101 is one of the databases with the largest number of action categories and samples, which contains 13320 videos and 101 categories. The database samples are taken from various sports samples collected from the BBC/ESPN and downloaded from the Internet. Fig. 5 shows several clips of UCF-101.

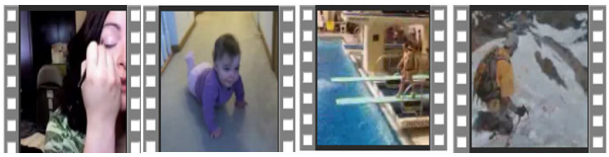

FIGURE 5. Several clips of UCF-101.

HMDB-51 contains 6849 videos and 51 categories. Each category contains at least 101 videos. Most of the videos are from movies, and some are from public databases such as YouTube. Fig. 6 shows several clips of HMDB-51.

FIGURE 6. Several clips of HMDB-51.

We use split1 of UCF-101 and HMDB-51 for training and validation. When testing, the dataset is the same as the validation set.

B. IMPLEMENTATION

In the experiment, we take n = 1,2, that is, add one and two up-sampling layers respectively for the experiment.

Training: We use SGD with momentum to train the networks. At the same time, in order to increase the data, we randomly generate samples in the video of training data. Fig. 7 shows the method of generating training samples.

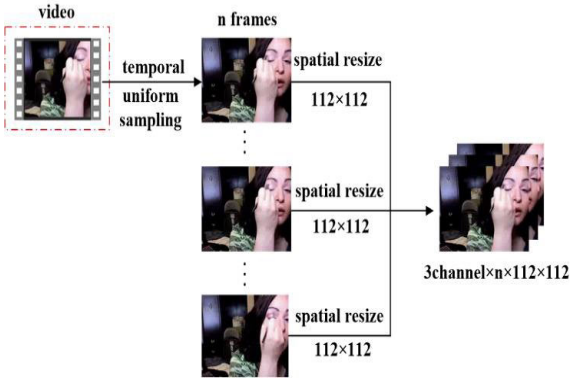

FIGURE 7. Generate training samples.

Firstly, we choose a temporal location in the video, generate the training samples by uniform sampling, and then produce a 16-frames clip around it. If the video is less than 16 frames, we loop it multiple times. Then, we randomly pick a spatial location from four angles or the center and select the spatial scale of a sample for multi-scale cropping.

The aspect ratio of the sample is 1, and the sample is spatio-temporally cropped. Finally, we get that the size of each sample is 3channels × 16frames × 112pixels × 112pixels. All the resulting samples retain the same class labels from their original videos.

In the training, we initialize the parameters of the network with resnext-101 pre-trained on Kinetics and fine-tune the last two layers using SGD optimizer with momentum 0.9. We start with the learning rate of 0.05 and divide it by 10 after the validation loss saturates. To prevent overfitting, we also add dropout with 0.5. The loss function we use is the cross-entropy loss function, $\hat{y}$ represents the predicted value, y represents the actual value.

$$L(\hat{y}, y) = \begin{cases} -\log(\hat{y}) & y = 1 \\ -\log(1 - \hat{y}) & \text{otherwise} \end{cases} \tag{10}$$

Recognition: We use a sliding window to produce the input clips. Then, we input the clip into the network and evaluate the class score of the clip, which is the averaged of all the clips. We use the method of up-sampling to generate two feature maps with different scales using the above method to train and recognition and get different class scores. Finally, we fuse the different class scores through [23] to get the final class score.

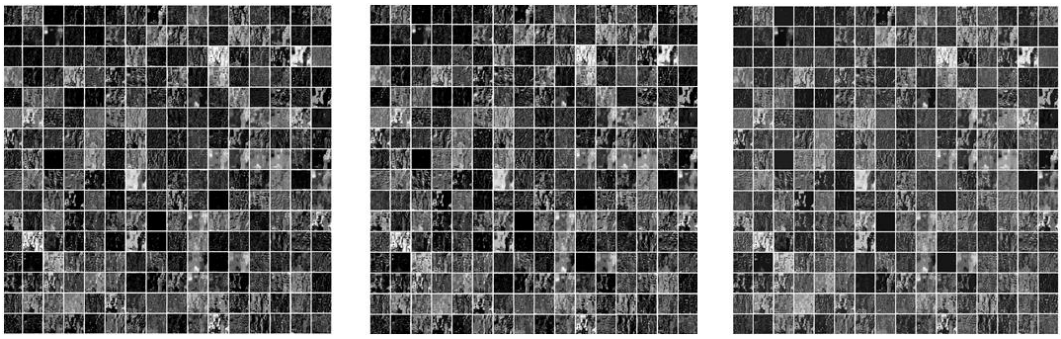

FIGURE 8. The feature maps of three up-sampling methods.

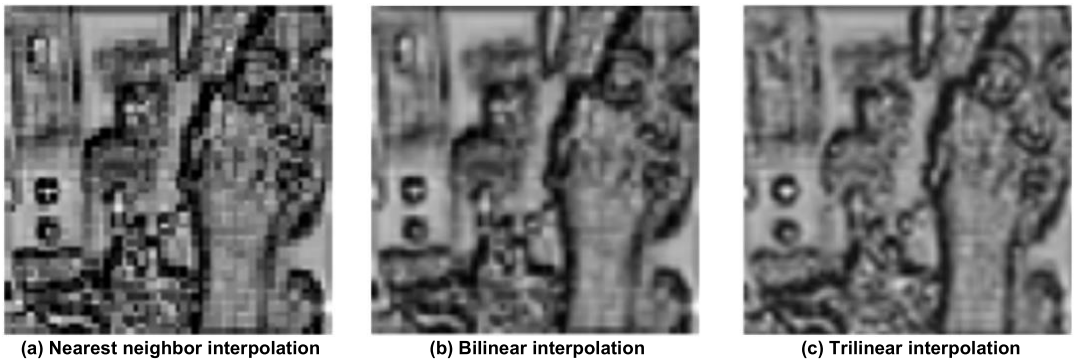

FIGURE 9. The one feature map of these feature maps.

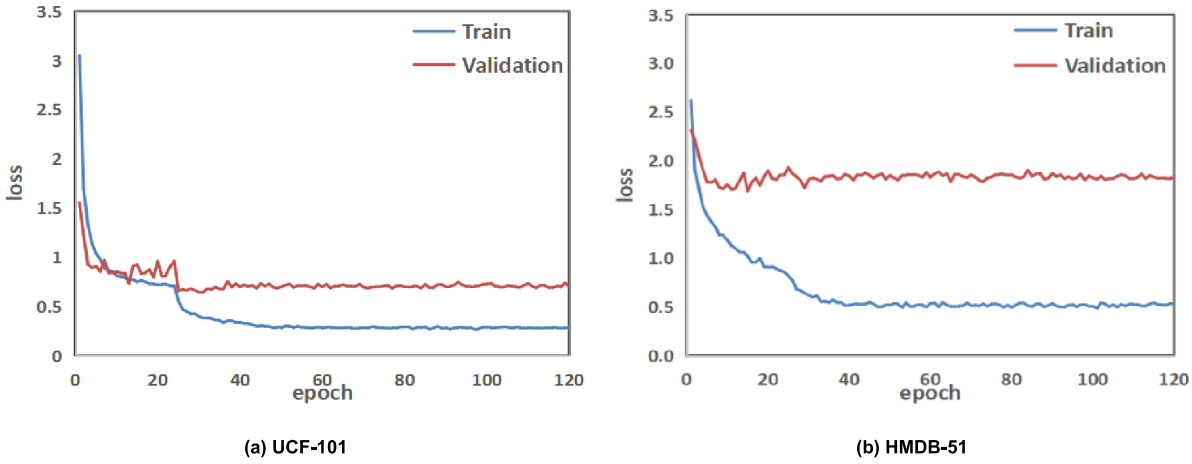

FIGURE 10. Training and validation losses (16 frames).

C. FEATURE MAPS

We use three different up-sampling methods to experiment separately. Fig. 8 shows the feature maps of different up-sampling methods.

We separately enlarge the same one feature map of these three feature maps, as shown in Fig. 9 we can see that the feature map obtained by the nearest neighbor interpolation method is fuzzy and has obvious jagged; The feature map obtained by the bilinear interpolation method is not obvious jagged, but the edge is blurred; The feature map obtained by the trilinear interpolation method is not obvious jagged, and the edge is clearer. These results prove that our choice in IV is correct.

D. RESULTS

We separately sample a 16-frames clip and a 32-frames clip for training. Fig. 10 and 11 separately show the training and validation losses of different sample-durations. As can be seen in the figures, the validation losses are slightly higher than training losses on UCF-101, which indicates that the

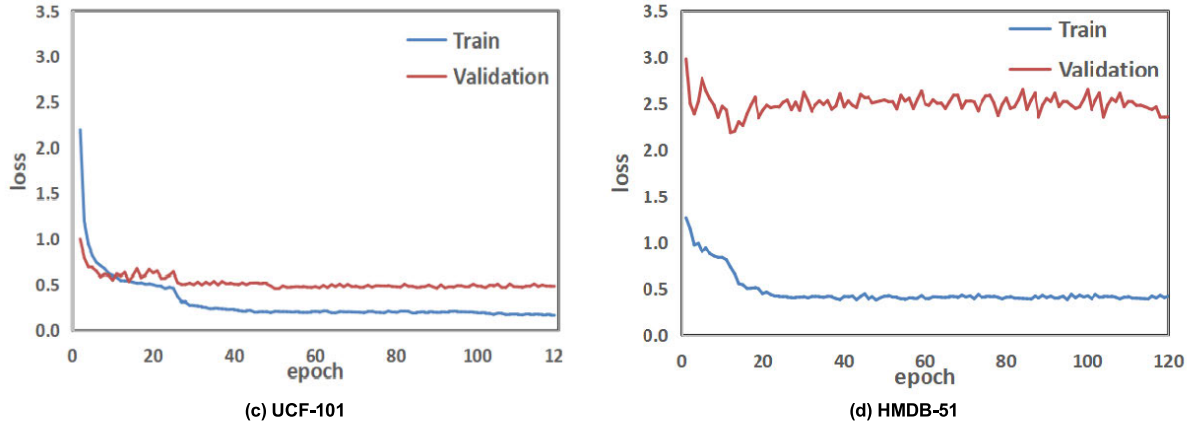

FIGURE 11. Training and validation losses (32 frames).

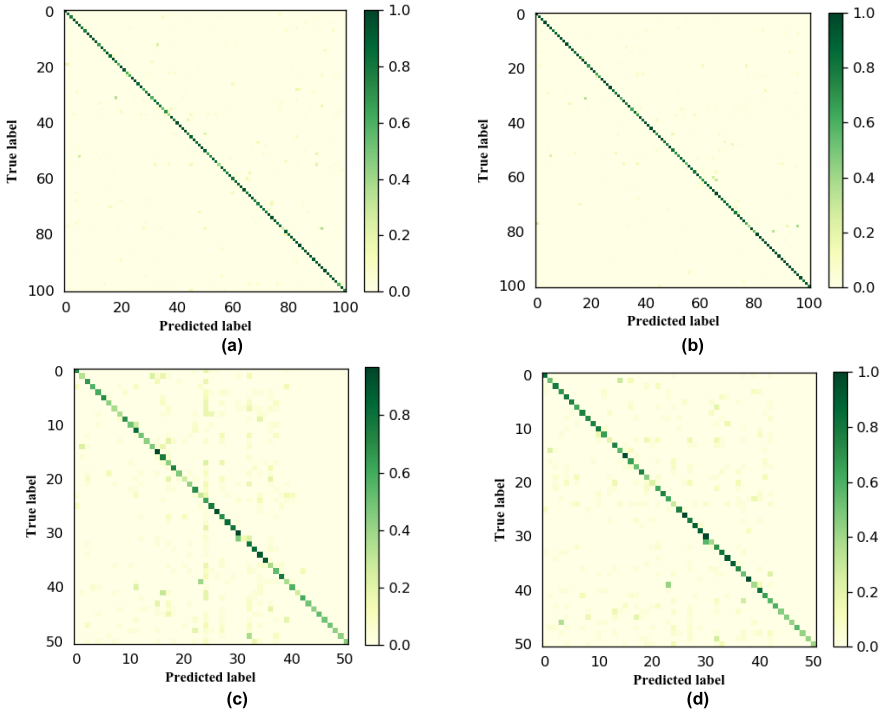

FIGURE 12. The confusion matrix. (a) and (c) are respectively ucf-101 and hmdb-51 of 16 frames. (b) and (d) are respectively ucf-101 and hmdb-51 of 32 frames.

network performs well on UCF-101. However, the validation losses quickly converge and are higher than the training losses on HMDB-51, which indicates that the performance of the network on HMDB-51 is not as good as UCF-101.

TABLE 2. The accuracies of different sample-duration.

Dataset	Clip	Video	Clip	Video
	16f×112×112		32f×112×112	
UCF-101	85.9	89.2	87.5	90.3
HMDB-51	57.9	57.9	57.8	90.3

For evaluating the network, we measure clip and video accuracies. We take the maximum score to each clip score. Then we get the video score by averaging clip scores. Table 2 shows the accuracies of different sample-durations. With the number of sample-duration increases, the clip-level accuracy and video-level accuracy also improve.

We then compare our architecture with state-of-art algorithms and the results are presented in Table 3.

TABLE 3. Comparison with state-of-the-art algorithms.

Method	UCF-101	HMDB-51
iDT [4]	85.9	51.9
Res3D [27]	85.8	54.9
ConvNet [28]	65.4	-
C3D [29]	82.3	-
P3D [30]	88.6	-
Two-stream CNN [14]	88.0	59.4
Two-stream+LSTM [31]	88.6	-
F_{st} *CN* [32]	84.5	49
Proposed	90.3	58.4

FIGURE 13. In UCF-101 the most confused classes.

FIGURE 14. In HMDB-51 the most confused classes.

We can see that the proposed architecture achieved higher accuracies compared with other state-of-art algorithms.

Moreover, our architecture improved 4.4% and 2.3% compared with the most effective iDT [4] with the hand-crafted feature, and the two-stream [14] with a deeper feature on the UCF-101.

Fig. 12 shows the confusion matrices of different sample-durations of UCF-101 and HMDB-51 respectively.

In UCF-101, most of the classes perform well, even some classes, such as ApplyEyeMakeUp, ApplyLipstick, Archery, BabyCrawling, Rowing separately reach the accuracies of 99.8%, 99.7%, 99.9%, 99.9%, 99.8%. However, as shown in Fig. 12 (a) and (b), the most confused classes are: Shotput with VolleyballSpiking (39.1%), Skiing with Surfing (35.0%), ShavingBeard with ApplyEyeMakeUp (32.6%), RockClimbingIndoor with RopeClimbing (32.7%), PlayingFlute with PlayingViolin (29.2%). Fig. 13 shows the most confused classes of UCF-101. We can see that Shotput and VolleyballSpiking are similar movements and have many people in the scenes. Skiing and Surfing are from the same sports and have the same background. ShavingBeard and ApplyEyeMakeUp are in a similar room of a house and have similar movements. RockClimbingIndoor and RopeClimbing are from the same type of sports. PlayingFlute and PlayingViolin are playing Musical Instruments, they're in similar positions and moving their hands.

In HMDB-51, these classes do not perform as well as UCF-101 does. As shown in Fig. 12 (c) and (d), the most confused classes are: ride_horse with ride_bike (56.6%), smile with laugh (43.3%), talk with chew (36.7%), cartwheel with flic_flac (30.0%), walk with run (26.7%). Fig. 14 shows the most confused classes of HMDB-51. Ride_horse and ride_bike both have the riding movements and a similar scene. Smile and laugh are similar facial expressions. Talk and chew have similar mouth movements. Cartwheel and flic_flac are two similar actions. Walk and run have similar leg movements. However, there are some classes confused with other classes. Fall_floor is confused with kick_ball, jump, punch, stand. Shoot_bow is confused with shoot_gun, jump, punch.

VI. CONCLUSION

At present, human action recognition is the focus and difficulty of research and has a very wide application prospect, mainly used in monitoring, human-computer interaction, and other scenarios. Due to the complexity and diversity of human action, research on human action recognition has great challenges. This paper presented a solution to improve the performance of human action recognition. we proposed the architecture based on Multi-feature Map Fusion, which uses multiple up-sampling layers to enlarge feature maps, so that smaller targets can be better detected, at the same time, the information of the features extracted by the network is more and clearer. In our architecture, for the up-sampling method, the nearest neighbor interpolation method, the bilinear interpolation method, and the trilinear interpolation method have been studies. Experiments show that the effect of the feature map obtained by the trilinear interpolation method is better than the other two methods. Simultaneously, we used the clip with different sample-durations for training. The results indicate that with the number of sample-duration increases, the accuracies also improve. Finally, for the results obtained by the network, we did not use the method of averaging scores as mentioned in [14] to fuse the results. We proposed to use the weighted geometric means combination forecasting method based on L_1 norm to fuse the obtained n results. The proposed architecture achieved 90.3% and 58.4% on UCF-101 and HMDB-51, which illustrates that the architecture is effective and comparable.

REFERENCES

[1] A. Diba, V. Sharma, and L. Van Gool, "Deep temporal linear encoding networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jul. 2017, pp. 2329–2338.

[2] Q. Fu, "Analysis of human behavior recognition," *China Comput. Commun., Theory Ed.*, vol. 2017, no. 24, pp. 146–147, 2017. [Online]. Available: http://www.cnki.com.cn/Article/CJFDTotal-XXDL201724059.htm

[3] T. Yu, H. Gu, L. Wang, S. Xiang, and C. Pan, "Cascaded temporal spatial features for video action recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 1552–1556.

[4] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 3551–3558.

[5] G. Willems, T. Tuytelaars, and L. V. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, Oct. 2008, pp. 650–663.

[6] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th Int. Conf. Multimedia*, Augsburg, Germany, 2007, pp. 357–360, doi: 10.1145/1291233.1291311.

[7] A. Klaeser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Leeds, U.K., Sep. 2008, pp. 275:1–275:10.

[8] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng, "MiCT: Mixed 3D/2D convolutional tube for human action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 449–458.

[9] D. He, F. Li, Q. Zhao, X. Long, Y. Fu, and S. Wen, "Exploiting spatial-temporal modelling and multi-modal fusion for human action recognition," Jun. 2018, *arXiv:1806.10319*. [Online]. Available: http://arxiv.org/abs/1806.10319

[10] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Columbus, OH, USA, Jun. 2014, pp. 806–813.

[11] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017, doi: 10.1016/j.neucom.2016.12.038.

[12] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[13] V.-M. Khong and T.-H. Tran, "Improving human action recognition with two-stream 3D convolutional neural network," in *Proc. 1st Int. Conf. Multimedia Anal. Pattern Recognit. (MAPR)*, Ho Chi Minh City, Vietnam, Apr. 2018, pp. 1–6.

[14] K. Simontan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2014, pp. 568–576. [Online]. Available: https://dl.acm.org/doi/10.5555/2968826.2968890

[15] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6299–6308.

[16] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jun. 2018, pp. 7794–7803.

[17] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1798–1807.

[18] Z. Qiu, T. Yao, and T. Mei, "Deep quantization: Encoding convolutional activations with deep generative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6759–6768.

[19] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2740–2755, Nov. 2019.

[20] C. Li, Q. Zhong, D. Xie, and S. Pu, "Collaborative spatiotemporal feature learning for video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 7872–7881.

[21] Y. Zhou, X. Sun, C. Luo, Z.-J. Zha, and W. Zeng, "Spatiotemporal fusion in 3D CNNs: A probabilistic view," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 9829–9838.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2117–2125.

[23] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1492–1500.

[24] M. Zolfaghari, K. Singh, and T. Brox, "ECO: Efficient convolutional network for online video understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 695–712.

[25] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[26] H. Y. Chen, "Weighted geometric means combination forecasting method based on L_1 norm," *J. Anhui Univ., Natural Sci.*, vol. 28, no. 4, pp. 5–10, Jul. 2004, doi: 10.2116/analsci.20.717.

[27] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "ConvNet architecture search for spatiotemporal feature learning," Aug. 2017, *arXiv:1708.05038*. [Online]. Available: http://arxiv.org/abs/1708.05038

[28] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1725–1732.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4489–4497.

[30] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5533–5541.

[31] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4694–4702.
[32] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Boston, MA, USA, Dec. 2015, pp. 4597–4605.

HAOFEI WANG received the bachelor's degree in electrical engineering and automation from Zhejiang Sci-Tech University, in 2018, where she is currently pursuing the master's degree with the Department of Control Science and Engineering. Her current research interests include pattern recognition and intelligent systems.

JUNFENG LI received the B.S. degree in electrical engineering from Zhengzhou University, China, in 2002, the M.S. degree in mechanical engineering from Zhejiang Sci-Tech University, Hangzhou, China, in 2005, and the Ph.D. degree in mechanical engineering from Donghua University, Shanghai, China, in 2010.

From 2005 to 2011, he was a Lecturer with the Department of Automation, Zhejiang Sci-Tech University. Since 2011, he has been an Assistant Professor with the Department of Electrical Engineering, Zhejiang Sci-Tech University. His research interests include intelligent information processing, machine learning, and defect detection.