Computers, Materials & Continua
DOI: 10.32604/cmc.2022.031664
Article

Tech Science Press

Motion Enhanced Model Based on High-Level Spatial Features

Yang Wu[1], Lei Guo[1], Xiaodong Dai[1], Bin Zhang[1], Dong-Won Park[2] and Ming Ma[1,*]

[1]College of Computer Science and Engineering, Inner Mongolia University, Hohhot, 010021, China
[2]Department of Information and Communications, PaiChai University, Daejeon, 35345, Korea
*Corresponding Author: Ming Ma. Email: csmaming@imu.edu.cn
Received: 23 April 2022; Accepted: 29 May 2022

Abstract: Action recognition has become a current research hotspot in computer vision. Compared to other deep learning methods, Two-stream convolutional network structure achieves better performance in action recognition, which divides the network into spatial and temporal streams, using video frame images as well as dense optical streams in the network, respectively, to obtain the category labels. However, the two-stream network has some drawbacks, i.e., using dense optical flow as the input of the temporal stream, which is computationally expensive and extremely time-consuming for the current extraction algorithm and cannot meet the requirements of real-time tasks. In this paper, instead of the dense optical flow, the Motion Vectors (MVs) are used and extracted from the compressed domain as temporal features, which greatly reduces the extraction time. However, the motion pattern that MVs contain is coarser, which leads to low accuracy. In this paper, we propose two strategies to improve the accuracy: firstly, an accumulated strategy is used to enhance the motion information and continuity of MVs; secondly, knowledge distillation is used to fuse the spatial information into the temporal stream so that more information (e.g., motion details, colors, etc.) is obtainable. Experimental results show that the accuracy of MV can be greatly improved by the strategies proposed in this paper and the final recognition for human actions accuracy is guaranteed without using optical flow.

Keywords: Action recognition; motion vectors; two-stream; knowledge distillation; accumulate strategy

1 Introduction

Multi-media data, including video, music, text, and so on, has become the main research topic because of its increasing number on the Internet. In particular, videos have become a new way of communication among internet users, which can contain more information and rich contents. The massive amount of video also provides important data to support video-related tasks, such as video search, video surveillance, etc. Action recognition, therefore, has become a hotspot in the current computer vision field, of which the purpose is to enable the computer to automatically recognize

human actions in the real environment. Therefore, the key to human action recognition is how to establish the mapping between video content and action category description.

The early action recognition approaches are based on hand-crafted features, such as Histogram of Oriented Gradients (HoG) [1], Histogram of Optical Flow (HoF) [2], Motion Boundary Histogram (MBH) [3], etc, and encoded these descriptors with Bag of Visual Words (BoVW) for classification.

Recently, the Convolutional Neural Networks (CNNs) have been proved to be an effective method to classify images [4], detect objects in images [5,6], perform semantic segmentation [7], or re-identify vehicles [8]. The success of these tasks has prompted to use CNNs in action recognition tasks in recent works [9–12]. More recently, one successful example of this task is two-stream ConvNets [10]. The approach learns spatial and temporal information by feeding two different inputs (i.e., RGB frames and optical flow). Because temporal information is directly fed into the network for training independently, the network can learn more effective temporal features than early approaches. However, the two-stream method needs optical flow as the input of the network, which is computationally intensive and time-consuming.

To solve this problem, we use Motion Vectors (MVs) and RGB frames to achieve action recognition. MVs are mainly used for video coding tasks, which represent the position movement of the macroblocks as Fig. 1 shown.

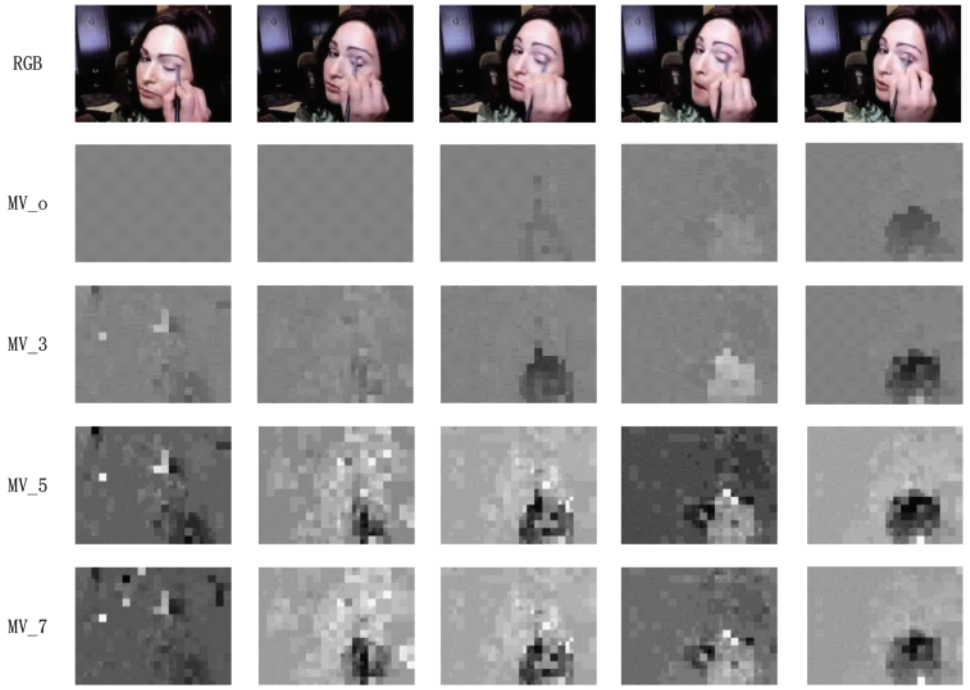

Figure 1: MV examples for different enhancement algorithms

Therefore, similar to optical flows, MVs contain motion information of objects but this information is inaccurate to unveil the temporal relationship. The reason is that MVs are not to represent the movement of objects, but only to represent macroblocks' position movement between two frames for a compressed video. Moreover, the generation of MVs relies on video compressed coding algorithms, which generate different MVs when using different algorithms (such as H.264 or HEVC). In this paper,

a new strategy is proposed for MVs to obtain more motion information to improve the accuracy of action recognition.

It can enhance motion information for two reasons:

- After enhancement, the information of consecutive frames is accumulated, hence temporality is enhanced.
- I-frame decompressed directly without reference to other frames, hence there are no MVs in I-frame. The MV of the I-frame can be 'generated' indirectly by enhancement so that the MV information is continuous.

Furthermore, to further improve the accuracy, we employ a strategy to transfer the spatial information of RGB to the MV stream so that the MV stream learns spatial information. Inspired by these papers [13–15], we train the teacher network using the RGB frame of the video as input to the network. The information before the softmax layer in the teacher network is used as privileged information to guide the training of the network with MV as input, and we use a cross-entropy loss function between the video label and the softmax layer. Thus, the network learns the spatial features of the teacher network as well as information about the temporal features of the MV itself. This strategy allows the network to improve its accuracy compared to training the original MV alone.

2 Related Work

Action recognition tasks have been a research hotspot in the field of computer vision [16–21]. Traditional approaches focus on first using local feature descriptors, such as HoG [1], HoF [2], and MBH [3], among others, and then use BoVW to encode these descriptors and use them for classification. The authors in [22] combined two features which are Scale Invariant Feature Transform (SIFT) and Space-Time Interest Point features (STIP) respectively to characterize the video. Traditional methods have been predominant for quite some time, and these features are extremely effective for classification tasks, but there are some problems as well. Traditional features are time-consuming, and feature extraction of the data requires knowledge of the relevant field of expertise. Therefore, the above reasons make it difficult to meet the needs of realistic tasks. Compared to traditional methods, this paper uses deep learning methods to extract effective features from videos in the action recognition task.

Recently, deep neural networks have become a mainstream approach for image analysis tasks [23]. Therefore, some scholars have started to use deep neural networks in action recognition tasks. For image and video tasks, CNNs [24] are more widely used, which can extract powerful features from image datasets and learn knowledge about images without human intervention. The common practice in the video task is to consider a video clip as a set of frames, and then the classification results can be obtained by entering a frame or a random video clip in the CNNs. The authors in [9] used CNNs and proposed several network architectures using different fusion strategies. They consider each video as a packet of fixed-sized short clips consisting of several adjacent frames. The authors in [10] proposed Two-Stream Convolutional Neural Networks for action recognition, where one stream uses RGB frames as inputs and the other stream uses optical flow. In addition, the RGB stream captures spatial information such as the appearance or color of an object, while the optical flow of multiple frames captures temporal information. The innovation of the two-stream technology is to use optical flow as the second feature and fuse it with RGB stream at the end of the network, thereby greatly improving recognition performance. For these reasons, the optical flow has become one of the normal features in action recognition tasks. Additionally, The authors in [25] proposed a novel sparse classification model for conducting action analysis, which extracts the deep CNN features from the input samples

and uses sparse coding. The authors in [26] proposed a novel hidden two-stream collaborative learning network that masks the steps of extracting the optical flow in the network and greatly speeds up the action recognition. The authors in [27] introduced a method mainly based on the BoVW, which is divided into a feature extractor and a classifier. Therefore, it is not an end-to-end model and the results are not effective on the corresponding datasets. The authors in [28] are inspired by the two streams and proposed a new architecture consisting of three streams to obtain multi-modality information. Besides, 3D CNNs [29] are used in some papers, and experiments have shown that 3D CNNs can learn spatiotemporal features better than 2D convolution. Furthermore, Carreira et al. [11] proposed I3D which used RGB and Flow streams pre-trained on the Kinetics dataset [30]. It achieved the state of the art on the HMDB51 [31] and UCF101 [32] datasets.

Irrespective of the differences between the above methods, hand-crafted optical flow [33,34] is extracted before training in all above methods. This method is a bottleneck because the optical flow is computationally expensive and cannot be evaluated by most of the current algorithms [33,34]. In contrast, instead of optical flow, MVs are used in our method, thus avoiding the optical flow extraction work which is time-consumed. The effectiveness of MVs for motion recognition tasks has been demonstrated [35–37]. It needs to be noted, however, that the purpose of MVs is not to unveil the accurate motion information and the temporal relationship between frames, but to exploit temporal redundancy between adjacent frames to reduce the bit rate of the video compression. Therefore, MVs only contain coarse motion information. If the optical flow is replaced by MVs directly, the accuracy will drop sharply, which is reflected in [35,36] and the experiments in this paper. To solve this problem, the MVs are enhanced in this paper, and experiments show that our method can improve accuracy effectively. To further improve the accuracy of MVs in action recognition tasks, this paper addresses this issue based on the idea of knowledge distillation. Knowledge distillation was first proposed in [13], which is mainly used to transfer knowledge from a complex model to a simple model, namely, taking complex class probabilities as a "soft target" of a smaller model. Inspired by the knowledge distillation technique [14,35], we try transferring information from spatial streams to temporal streams. Our method takes MVs as input, eventually allowing the temporal streams to learn spatiotemporal features to improve their accuracy. Compared to the literature [35], which also uses the knowledge distillation technique, but it is only used to match class probabilities. In contrast, our approach focuses on matching high-level features of MV and RGB. Moreover, all of the above methods use optical flow features either throughout the experiment or only during training, which requires the extraction of computationally intensive optical flows. In contrast to the above methods, our method does not require the extraction of optical flow during the entire experiment.

3 Methods

3.1 Enhance Motion Information by Accumulation

The first strategy, in this paper, is increasing the information of MVs. MVs which consisted of macro-block are mainly used for video motion compensation. Besides, MV is used to mark the movement of macroblocks. In addition to MV, video compression requires residuals, which can be used to encode non-key frames. The non-key frames in the video are divided into P and B frames according to different reference modes. P frames only need to refer to the preceding frames, while B frames need a bidirectional reference. The enhancement algorithm in this paper is to make the MV encoding multiple frames in the current MV, making the feature information richer than the original MV. Moreover, the corresponding MV can be generated by this method for the frames without MV (such as I frame), so that the motion information is more continuous than the original MV, as shown in

Fig. 1. In addition to the above advantages, the enhanced single-frame MV already has a character of temporality that the original MV does not have because the information on multiple frames is encoded in a single frame, whereas the original MV starts the information interaction of multiple MV frames only after the input to the network and during the convolution process. This strategy is earlier and enhances the frame-to-frame information interaction after the input to the convolution than before. The enhancement process in this paper is divided into the following steps:

(1) Partial decoding of video and extraction of MV information for saving.
(2) Converting the saved MV information to images of the MV. The algorithm for the conversion is as follows:
   a. Obtaining the forward maximum offset (*Maxs*), i.e., $dx > 0$ or $dy > 0$, and the reverse maximum offset (*Mins*), i.e., $dx < 0$ or $dy < 0$, in the current dataset, as shown in Eqs. (1) and (2).

$$Maxs = Max\{dx_1, \ldots, dx_n, dy_1, \ldots, dy_n\} \tag{1}$$

$$Mins = Min\{dx_1, \ldots dx_n, dy_1, \ldots, dy_n\} \tag{2}$$

   b. Processing for each dx (dy) yields Sx (Sy), where we use k to represent different orientations, and k can be taken as x or y, as shown in Eqs. (3) and (4).

$$S_k = 128 + 127 * (d_k/Maxs)\ d_k > 0 \tag{3}$$

$$S_k = 127 * (1 - d_k/Mins)\ d_k < 0 \tag{4}$$

As above, the offset center 0 is mapped to 128, in other words, the pixel value 128 means that the macro blocks have no displacement. The pixel values for the leftward and upward displacement conversions are less than 128, while the corresponding rightward and downward pixel values are greater than 128.

(3) Before enhancement, it is necessary to take the appropriate time slice (3, 5, and 7 were attempted in this paper) and superpose the same position of the MV within the time slice. The MV at the same location for three frames as Fig. 2 shown.

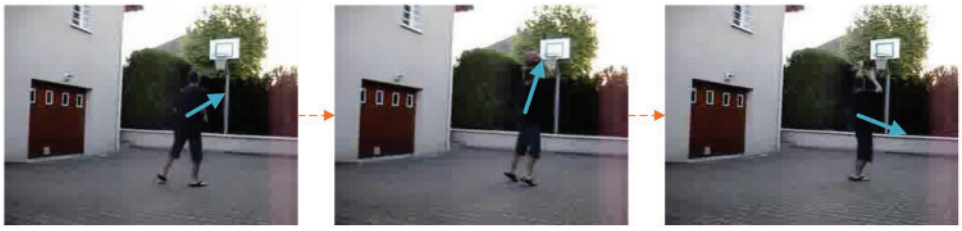

Figure 2: The same location on MVs of 3 frames. The blue arrow represents the motion vector of the current position

To avoid pixel accumulation exceeding the range of $0 \sim 255$, it is necessary to normalize the superimposed value and then enlarge it to the pixel range (N means Normalization, in this paper, Min-Max Normalization is used). T represents the length of time and MV_t represents the value of MV at the current time t, as shown in Eq. (5):

$$MV_t = N\left(\sum_{i=0}^{T-1} MV_{t+i} - T * 128\right) * 255 \tag{5}$$

This method can effectively enhance the characteristics of MV, and the results of the algorithm are shown in the following experimental part.

3.2 Learning Strategy Based on Spatial-Temporal Information

The second augmentation strategy in this paper is the learning strategy based on spatial-temporal information (LSSI). Part of the reason for this strategy comes from the coarse representation of the motion position and orientation information of the object in the MV, which is related to the spatial information of the object in the RGB frame (due to the coarse marking of the position of the moving subject). Secondly, MV and optical flow are similar in that they both have timing information and multiple MV input network, which allows the network to learn the temporality of the motion. Analogous to the literature [15], if the MV network is trained, the training data are represented using the following ternary form as Eq. (6):

$$\{(x_1,\ x_1*,\ y_1),\ (x_2,\ x_2*,\ y_2),\ \ldots,\ (x_n,\ x_n*,\ y_n)\} \tag{6}$$

Here $(x_i,\ y_i)$ denotes a pair of training samples (i.e., the MVs of n frames and the video labels of these MVs), and the privileged information (x^*) refers to the features of the multi-frame RGB (i.e., the input of the softmax layer in the RGB network). The input to the softmax layer of the RGB network is treated as x^* primarily because this layer has high-dimensional features of the video. Compared to the features of the previous layer, this layer more accurately represents the information in the video, which includes information about the object such as its position.

Fig. 3 shows the structure of our network. The different colors of the network in the figure represent whether the weights are frozen or not. The gray blocks represent two loss functions which are Cross-Entropy Loss and Mean Square Error (MSE). The network is divided into two parts (i.e., the feature extraction layer consisting of multiple convolutional layers and the FC layer). The yellow part represents a trained RGB stream with frozen parameters. The blue part represents the stream that uses MV as input.

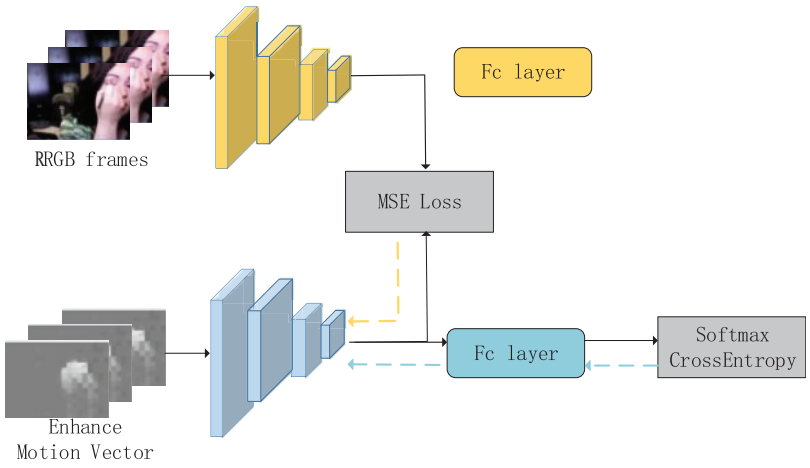

Figure 3: The structure of our network

The solid line represents the forward propagation, while the dashed line represents the update of the parameter, as shown in Fig. 3. The blue dashed line represents the back propagation of the cross-entropy loss between the softmax result of the MV stream and the real label, and the yellow dashed line represents the back propagation of the loss between the RGB stream and the high-dimensional features of the MV stream. Thus, parameter updates at the FC layer depend on the cross-entropy loss function, while parameter updates at the front half of the network depend on the combination of these two losses.

To enable MV networks to effectively learn the spatial information of RGB, the following loss function is used in this paper as Eq. (7):

$$Loss_{MS} = \lambda MSE(MS_{hf},\ RS_{hf}) + CrossEntropy(MS_{soft\max},\ y) \tag{7}$$

The first half of loss is the loss function of the high-dimensional information (i.e., MS_{hf}, RS_{hf}) of the MV and RGB networks, which takes the form of MSE, while the second half uses a cross-entropy loss function to reduce the loss between the output of the MV stream and the true label y. The use of λ as a weight between the two different loss functions indicates their relationship.

This strategy is designed so that the network can learn both spatial and temporal information at the same time without losing too much information about the motion. According to these two strategies, the accuracy of the MV network can be effectively improved.

4 Experiments

In this section, we first describe the details of our experiments. After that, we report our results and analysis procedures. Further, the principles of our proposed method can be easily applied to other tasks, such as video classification, object detection, and action localization, mainly because action recognition is more fundamental than other tasks.

4.1 Datasets and Device

We evaluate our approach on two more widely used datasets, namely, HMDB51 [31] and UCF101 [32]. UCF101 contains a total of 13,320 videos in 101 human action categories, of which each category is made up of videos, which include 25 people doing 4–7 sets of the corresponding action, each with a resolution of 320∗240. HMDB51 contains 6766 videos in 51 human action categories with a resolution of 320∗240 from YouTube, Google videos, etc., and the size of the dataset size is 2G. Additionally, UCF101 and HMDB51 contain clipped videos, and each video corresponds to an action label. Besides, during our entire experiments, we trained our network in the device, which contains NVIDIA Tesla P40 and Core (TM) i7-4790 CPU.

4.2 Experiments Details

For RGB images, which can be read directly from the video, we extracted video frames at a rate of 103 images per second, compressed them using JPEG, and saved them as JPG files. In our experiments, we adjust the minimum edge size of the image to 256 pixels. Video frames were extracted using the OpenCV-python library. For MV image, we can't get it directly by complete decoding, but by partial decoding, we save it as JSON file, and then process it to get MV image, the process is in the first part of methods. According to [14,38], the size of the clip used in the experiment was 16 consecutive frames, and was randomly cropped 112∗112 and flipped randomly so that a reasonable running time could be guaranteed. Before inputting the network, the data needs to be normalized, RGB minus the mean

of ActivityNet, while the mean of MVs is calculated by traversing the entire dataset. In the testing phase, we use center crop to cut the frames into the size allowed by the network. The final category for each video is the category of the video with the highest score after averaging the scores for each video clip. In our experiments, we use the ResNext 3D structure with a depth of 101 layers, mainly due to its performance on the Kinetics, UCF101, and HMDB51 datasets [38]. Our RGB stream is pre-trained on the Kinetics dataset and fine-tuned using SGD. Learning rate starts from 0.001. MV stream is trained from scratch with 0.1 of the learning rate.

4.3 Results and Analysis

We analyze the effect of the two approaches on the classification results. In Section 1, we show the results of four different MV enhancement strategies and give possible reasons why the method is effective. In Section 4.3, we analyze the results of our method (LSSI). We also show the accuracy when using different λ, as well as the loss variation curve, and finally, a comparative analysis with the currently popular algorithms.

4.3.1 Impact of Accumulating MVs

We first evaluate four strategies, namely, raw MV, 3-frames enhanced MV, 5-frames enhanced MV, and 7-frames enhanced MV on UCF101 and HMDB51. Tab. 1 shows the accuracy of the four strategies.

Table 1: Accuracy on UCF101 and HMDB51 datasets using different strategies

Datasets	Raw MV, %	3-frames, %	5-frames, %	7-frames, %
UCF101	72.40	73.51	72.66	72.72
HMDB51	24.77	26.60	17.32	17.25

We observe that compared to the original MVs on the action recognition task, the accuracy is largely improved using our approach, which can improve at about 1%. The main reason is that the original MV information is sparse, and its sparsity lies in the sparse spatial information on one MV frame and the discontinuous temporal information between consecutive frames, due to I-frames without MV information. Besides, the highest performance among the three methods of enhancement is the 3-frame enhancement method. We argue that MVs contain only coarse motion information and noise. The use of 5 and 7 frames enhance the motion information while also accumulating noise, which reduces the accuracy of classification.

4.3.2 Speed Evaluation

In this subsection, we compare the speed of model processing and the speed that optical flow and MV are extracted, respectively. In our implementation, we use the CPU to extract the MV while the GPU is used to complete the feed forward process of CNN.

Tab. 2 shows the speed of various models on different datasets (this paper uses fps as the speed metric. (S) and (D) represent single GPU and double GPU, respectively), where MV_CNN represents the forward propagation time of the network using the 3-frame enhanced MV as input. Compared to the time of the LSSI strategy, there are not much differences between the two methods, mainly because these backbone models are the same and the input size of the network is the same. Using

different numbers of GPUs can have a significant impact on speed, and the results are different for the two datasets, mainly due to the complexity of the video.

Table 2: Comparison of speed on UCF101 and HMDB51 datasets using different strategies

Datasets	MV_CNN(S)(FPS)	LSSI(S)(FPS)	MV_CNN(D)(FPS)	LSSI(D)(FPS)
UCF101	3939.8	3939.8	4328.9	4224.6
HMDB51	4212.5	4007.0	2079.6	2079.6

Tab. 3 compares the speed (fps) of optical flow (extracted using the TV_L1 algorithm) and that of MV extraction (decompressed directly from videos) on the same device (Intel (R) Xeon (R) Gold 5218 CPU). The comparison shows that although the extraction speed varies on the UCF101 and HMDB51 datasets (due to the complexity of the images in the video), the extraction speed of the optical flow is much lower than that of the MV, which is at least 210 times faster. The reason for the difference in speed is the different extraction processes and purposes. The optical flow mainly represents the motion patterns between images, matching and searching at the pixel level, so its motion features are significantly clear and accurate, but the extraction speed is too slow for the complexity of the calculation, which cannot meet the requirements of real-time tasks. Compared to optical flow, MV is mainly used for video compression, so the motion pattern is rougher, but only partial decompression of the video is needed. Hence, the extraction speed is much faster than the optical flow.

Table 3: Comparison of speed on extraction process of optical flow and MV

Datasets	Optical flow (FPS)	Motion vector (FPS)	Ratio
UCF101	5.78	1214.82	210.10
HMDB51	7.18	1952.15	271.74

4.3.3 Evaluation on LSSI

In this section, we begin evaluating our second strategy (LSSI) that transfers spatial information of RGB to the MV training process to improve the accuracy of MVs on action recognition. It is well known that the learning of neural networks relies on the back propagation of gradients, and the calculation of gradients requires a loss function. Therefore, our fusion strategy is to modify the original loss function for MV independent training, i.e., to combine a loss function that measures the difference between RGB and MVs high-level features. In this way, the network can learn both the spatial features of RGB and the temporal features of MVs. In the experiment, the value of a represents the relationship between the cross-entropy loss function and the MSE loss function. We experiment with different values of an including {30, 50, 70}, shown in Fig. 4.

It can be observed from Fig. 4 that the performance of MV flow is better when the value of λ is taken as 30. Additionally, we plot the curves of the $\lambda \times MSE$ using different λ as well as the cross-entropy loss function in Fig. 5. To observe the effect of λ on the experiments, and since λ only affects the value of MSE, the cross-entropy losses for different λ are averaged and plotted in Fig. 5.

The best performance is obtained when the value of $\lambda \times MSE$ is approximate to the value of the cross-entropy loss function ($\lambda = 30$) at the beginning of training. Therefore, the accuracy of MV can be effectively improved after using the LSSI method proposed in this paper.

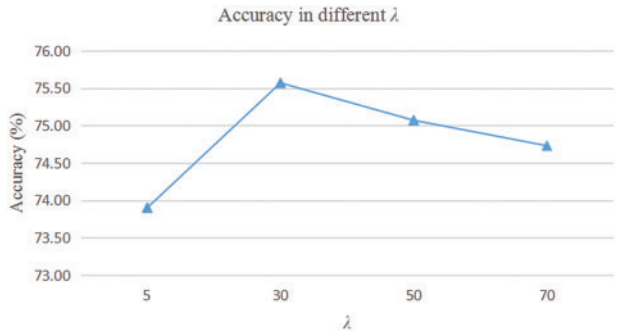

Figure 4: Accuracy in different λ on UCF101

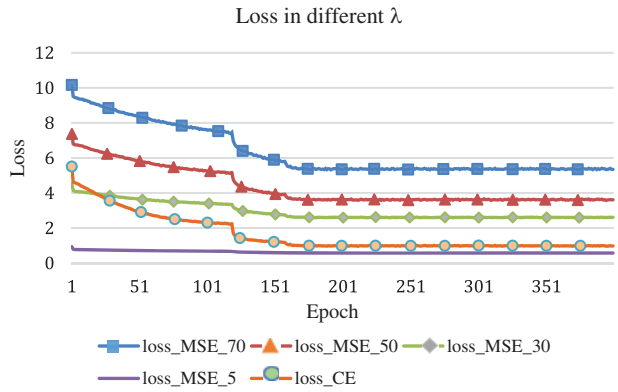

Figure 5: Evolution of losses while training LSSI with different λ on UCF101

4.3.4 Comparison with the State of the Art

In this section, we compare the accuracy of using the two methods in this paper on the UCF101 and the HMDB51 dataset with the accuracy of the state-of-the-art approach in Tab. 4.

Table 4: Accuracy on UCF101 and HMDB51

Method	Streams	UCF101,%	HMDB51,%
Two-stream ConvNets [10]	RGB	72.6	40.5
LRCN [39]	RGB	71.1	—
C3D(1 net) [29]	RGB	82.3	—
I3D [11]	RGB	84.5	49.8
TSN [12]	RGB	85.7	51.0
Two-stream ConvNets [10]	RGB + Flow	88.0	59.4
I3D [11]	RGB + Flow	93.4	66.4
TSN [12]	RGB + Flow	94.2	69.4
EMV + RGB-CNN [35]	RGB + MV	86.4	—
CoViAR [36]	RGB(I) + MV	90.4	60.3
Ours	RGB + MV	94.7	64.4

The uppermost algorithms in Tab. 4 do not require optical flow as input, while the middle level is algorithms that are based on a Two-stream approach and use dense optical streams as input features for the temporal stream. The algorithm in the bottom layer is also based on the Two-stream model but uses compressed domain features, i.e., MVs, as input for the temporal stream. It can be concluded that the higher accuracy of classification is obtained in the Two-stream method compared with only RGB. Although the accuracy of the Two-stream method with RGB and MVs is a little low than that using optical flow, applying the same structure MV extraction is faster than it. Therefore, our method does not use dense optical flow so that the time-consuming work about extracting optical flow is not required. Besides, our method achieves better performance than other methods [35,36] for least 4.1%.

4.3.5 Visualization of Feature Map

In this section, for the purpose of further demonstrating the effectiveness of our strategies, we visualize the feature maps of two strategies (3-frame enhancement and LSSI) in Fig. 6.

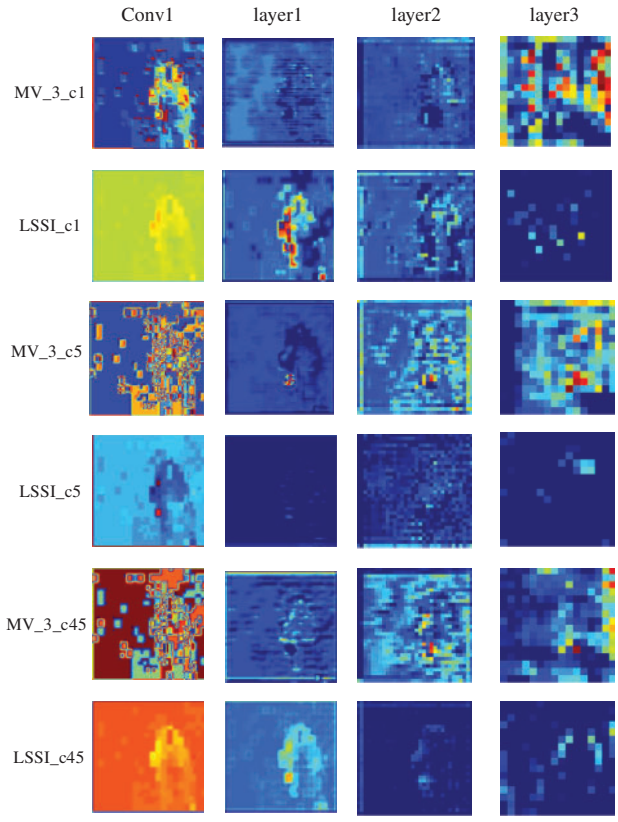

Figure 6: Different layer feature maps for MV streams using different strategies

The above figure shows the features of each layer of the model, in which different columns represent different layers of the network. The different rows represent different channels and feature maps obtained after training with different strategies, the odd number of rows represents the 3-frames enhanced MV proposed in this paper, while the even number of rows represents the LSSI strategy proposed in this paper. As Fig. 6 shows, the LSSI method reduces noise better than the 3-frame enhancement method and provides better motion information, such as position and edge information. For example, the last two lines represent the feature map of the 45th channel of the output, in which

the feature map obtained using the 3-frame enhancement method is more noisy although information about motion target is enhanced, and the subsequent feature maps are still noisy. The final high-level feature map obtains semantic information, while the noise is preserved together. For the feature map with LSSI strategy, the values do not change drastically, and the information about the moving object is highlighted. In layer2, the contour of the moving body is marked, as well as in layer3, the information on the edges is enlarged, and the surrounding noise is reduced. Therefore, the LSSI strategy is more effective.

5 Conclusion

In this paper, we propose two strategies to enable MVs to obtain more motion and spatial information. The experiments are conducted to show the effectiveness of our two strategies for human action recognition on UCF101 and HMDB51. Our first strategy accumulate multiple MVs. Then, the new MVs can be obtained after normalization, which makes the motion information of MVs more continuous. Experimental results show that compared to the original MVs on the action recognition task, the accuracy is improved using our first strategy. Further, the second strategy in this paper transfers spatial information of RGB to MV stream, which enables the MV stream to learn spatial and temporal information at the same time (i.e., LSSI). We experiment with different hyperparameters and illustrate the accuracy of the LSSI in the experimental results in both UCF101 and HMDB51 datasets. The results show that our method is faster than that using dense optical flow because the time-consuming work about extracting optical flow is not required. Besides, our method achieves better performance than other methods using MVs for least 4.1%. Moreover, visualization results show that the LSSI method can efficiently reduce noise and provide better motion information, such as position and edge information.

Acknowledgement: This work is supported by the Inner Mongolia Natural Science Foundation of China under Grant No. 2021MS06016 and the CERNET Innovation Project (NGII20190625).

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, San Diego, USA, pp. 886–893, 2005.

[2] I. Laptev, M. Marszalek, C. Schmid and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, USA, pp. 1–9, 2008.

[3] N. Dalal, B. Triggs and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. of European Conf. on Computer Vision (ECCV)*, Graz, Austria, pp. 428–441, 2006.

[4] P. Wang, K. Han, X. S. Wei, L. Zhang and L., Wang, "Contrastive learning based hybrid networks for long-tailed image classification," in *Proc of 2021 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, pp. 943–952, 2021.

[5] L. Chen, T. Yang, X. Zhang, W. Zhang and J. Sun, "Points as queries: Weakly semi-supervised object detection by points," in *Proc of 2021 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, pp. 8823–8832, 2021.

[6] D. Zhang, J. Hu, F. Li, X. Ding, A. K. Sangaiah *et al.,* "Small object detection via precise region-based fully convolutional networks," *Computers, Materials & Continua*, vol. 69, no. 2, pp. 1503–1517, 2021.
[7] N. Araslanov and S. Roth, "Self-supervised augmentation consistency for adapting semantic segmentation," in *Proc of 2021 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, virtual, pp. 15384–15394, 2021.
[8] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.,* "A multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3560, 2021.
[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar *et al.,* "Large scale video classification with convolutional neural networks," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, USA, pp. 1725–1732, 2014.
[10] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, Montreal, Canada, pp. 568–576, 2014.
[11] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. of 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, USA, pp. 4724–4733, 2017.
[12] L. Wang, Y. J. Xiong, Z. Wang, Y. Qiao, D. H. Lin *et al.,* "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. of European Conf. on Computer Vision (ECCV)*, Amsterdam, Netherlands, pp. 20–36, 2016.
[13] G. Hinton, O. Vinyals and J. Dean, "Distilling the knowledge in a neural network," in *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, Montreal, Canada, pp. 1–9, 2014.
[14] N. Crasto, P. Weinzaepfel, K. Alahari and C. Schmid, "MARS: Motion-augmented RGB stream for action recognition," in *Proc. of 2019 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, USA, pp. 7882–7891, 2019.
[15] V. Vapnik and R. Izmailov, "Learning using privileged information: Similarity control and knowledge transfer," *The Journal of Machine Learning Research (JMLR)*, vol. 16, pp. 2023–2049, 2015.
[16] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. of 2013 IEEE Int. Conf. on Computer Vision (ICCV)*, Sydney, Australia, pp. 3551–3558, 2014.
[17] L. Wang, Y. Qiao and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA, pp. 2674–2681, 2013.
[18] Y. H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, pp. 4694–4702, 2015.
[19] L. Wang, Y. Qiao and X. Tang, "Action recognition with trajectory-pooled deep convolutional descriptors," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, pp. 4305–4314, 2015.
[20] C. Gan, N. Wang, Y. Yang, D. Y. Yeung and A. G. Hauptmann, "Devnet: A deep event network for multimedia event detection and evidence recounting," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, pp. 2568–2577, 2015.
[21] A. M. Jamel and B. Akay, "Human activity recognition based on parallel approximation kernel k-means algorithm," *Computer Systems Science and Engineering*, vol. 35, no. 6, pp. 441–456, 2020.
[22] N. B. Aoun, M. Mejdoub and C. B. Amar, "Bag of sub-graphs for video event recognition," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Firenze, Italy, pp. 1566–1570, 2014.
[23] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, Lake Tahoe, Nevada, USA, pp. 1106–1114, 2012.
[24] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, pp. 1–14, 1995.

[25] B. Gu, W. Xiong and Z. Bai, "Human action recognition based on supervised class-specific dictionary learning with deep convolutional neural network features," *Computers, Materials & Continua*, vol. 63, no. 1, pp. 243–262, 2020.
[26] S. Zhou, L. Chen and V. Sugumaran, "Hidden two-stream collaborative learning network for action recognition," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1545–1561, 2020.
[27] S. Wang, Y. Yang, R. Wei and Q. Wu, "3-dimensional bag of visual words framework on action recognition," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1081–1091, 2020.
[28] C. Zhu, Y. K. Wang, D. B. Pu, M. Qi, H. Sun *et al.,* "Multi-modality video representation for action recognition," *Journal on Big Data*, vol. 2, no. 3, pp. 95–104, 2020.
[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. of 2015 IEEE Int. Conf. on Computer Vision (ICCV)*, Santiago, Chile, pp. 4489–4497, 2015.
[30] W. Kay, J. Carreira, K. Simonyan, B. Zhang and A. Zisserman, "The kinetics human action video dataset," arXiv preprint, arXiv:1705.06950, 2017.
[31] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, Barcelona, Spain, pp. 2556–2563, 2011.
[32] K. Soomro, A. R. Zamir and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," arXiv preprint, arXiv:1212.0402, 2012.
[33] D. Sun, X. Yang, M. Y. Liu and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. of 2018 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, pp. 8934–8943, 2018.
[34] C. Zach, T. Pock and H. Bischof, "A duality based approach for realtime TV-l1 optical flow," in *Proc. of 29th DAGM Symposium*, Heidelberg, Germany, pp. 214–223, 2007.
[35] B. Zhang, L. Wang, Z. Wang, Y. Qiao and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proc. of 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, pp. 2718–2726, 2016.
[36] C. Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola *et al.,* "Compressed video action recognition," in *Proc. of 2018 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, pp. 6026–6035, 2018.
[37] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding, and classification for action recognition," in *Proc. of 2014 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, USA, pp. 2593–2600, 2014.
[38] K. Hara, H. Kataoka and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?," in *Proc. of 2018 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, pp. 6546–6555, 2018.
[39] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan *et al.,* "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. of 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, pp. 2625–2634, 2015.