Real-Time Action Recognition based on Enhanced Motion Vector Temporal Segment Network

Xue Bai, Enqing Chen and Haron Chweya Tinega

January 16, 2019

Real-Time Action Recognition based on Enhanced Motion Vector Temporal Segment Network

Xue Bai[a, b], Enqing Chen*[a, b] , Haron Chweya Tinega[b]
[a]Industrial Technology Research Institute, Zhengzhou University, Zhengzhou, CHINA
[b]School of Information Engineering, Zhengzhou University, Zhengzhou, CHINA

ABSTRACT

At present, the method based on two-stream network has achieved good recognition performance in action recognition, however, its real-time performance is obstructed due to the high computational cost of optical flow. Temporal Segment Network (TSN), a successful example based on the two-stream network, achieves high recognition performance but cannot be processed in real time. In this paper, the motion vector TSN (MV-TSN) is proposed by introducing the motion vector into temporal segment networks, which greatly speeds up the processing speed of TSN. In order to solve the problem of performance degradation caused by the motion vectors lacking fine structure information, we propose a knowledge transfer strategy, which initializes the MV-TSN with the fine knowledge learned by optical flow. The experimental results show that the proposed method achieves a comparable recognition performance to the previous state-of-the-art approaches on UCF-101 and HMDB-51, and the processing speed is 206.2 fps, which is 13 times of the original TSN.

Keywords: Action Recognition, Temporal Segment Network, Motion Vector, Real-time Processing, Knowledge Transfer Strategy

1. INTRODUCTION

Action recognition has important applications in behavior analysis, video retrieval, video surveillance, human-computer intelligent interaction and other fields, which has attracted wide attention from academia and business circles and made great progress. Early works of[9-12] mainly carried out action recognition through extraction of hand-crafted features, feature coding, and classification steps. However with successful applications of Convolutional Networks (ConvNets)[1] in image recognition, face recognition and other fields, researchers are shifting their focus to the use of deep ConvNets in action recognition[2, 3,13-18]. Particularly, the two-stream ConvNets[2] have achieved remarkable results when applied in action recognition. This is because of their ability to break down a video into a spatiotemporal object from which spatial and temporal information can be extracted, processed separately and rather fused to improve action recognition. Whereas the spatial stream detects the appearance, the temporal stream detects motion in videos, therefore the spatial stream inputs an RGB video frame, while the temporal stream inputs continuous optical flow frames. However, the calculation of optical flow is very time-consuming, which limits the speed of the whole action recognition process. Therefore, optical flow is the main obstacle to prevent the two-stream network to be real-time.

In this paper, temporal segment network (TSN)[3] is chosen as the basic architecture of the system. TSN is an improved method based on the two-stream network. It divides a video into K segments, then randomly samples each segment to get the corresponding snippet. Then the short snippets are sent into the two-stream network to get the initial category score, which is rather fused to get the final result. This method[3] can utilize the long-range time information in the video, which is one of the most successful network models in action recognition based on deep learning. However, since TSN method is based on two-stream network, it needs to calculate optical flow which is computationally expensive. Although TSN achieves better action recognition performance, the poor real-time performance restricts its use in practice. For instance, when K40 GPU is used, it will take about 60 ms[4] to calculate an optical flow frame, which cannot meet the requirements of real-time processing.

In order to solve the real-time problem of the two-stream based method and get better recognition performance, a real-time motion vector TSN (MV-TSN) model is proposed in this paper. Inspired by Zhang et al.[5, 25], this model introduces the motion vectors into TSN[3], which cannot only realize real-time video processing but also ensure the state-of-the-art performance in video action recognition. Compared with the original two-stream network[2] used in Zhang's method[5], the TSN can effectively solve the long-term problem by using the longer-range information in the video, so that the recognition performance is much better than the original two-stream network. To compensate for the high computational cost of the TSN in real-life scenarios, this paper uses the motion vector to replace optical flow in TSN. Since the motion vectors can be obtained directly from the decoding process of standard video compression files, and no additional computational process is needed, this greatly reduces the time of video processing.

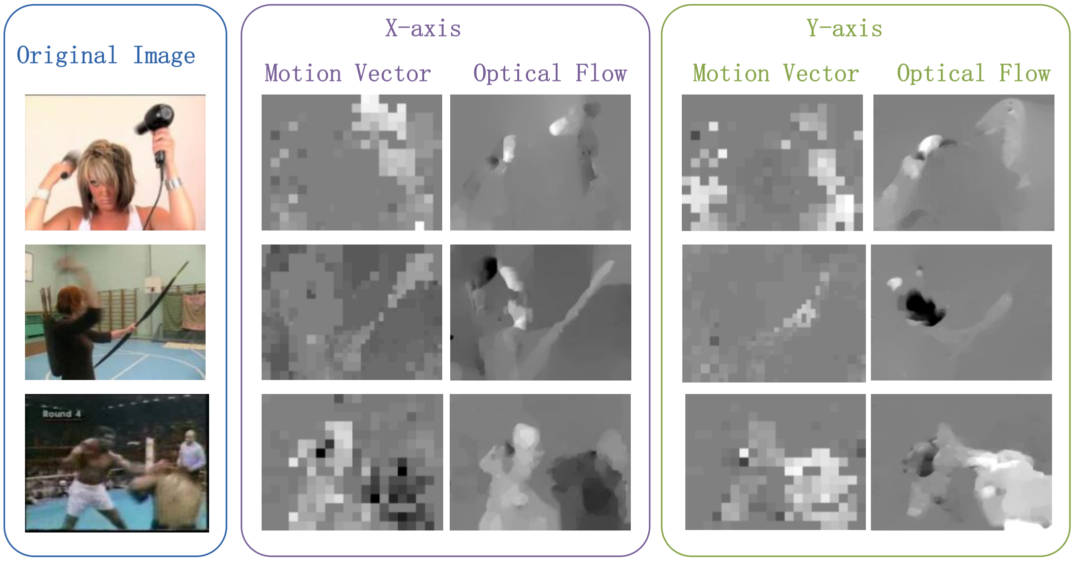

Figure 1. Comparisons between motion vector and optical flow images.

The motion vector is widely used in various video compression standards. It can be obtained directly from the video decoding process without other computational processes. The motion vector represents the relative movement of macroblocks between the current frame and its reference frame. Previous studies[5, 6] show that the motion vector contains motion information and can be used for action recognition. However, since the motion information generated by motion vectors is block-level and the optical flow contains the motion information at the pixel level, the structure of the motion vector is rougher than that of the optical flow. If the optical flow is replaced by the motion vector directly, the performance will be seriously degraded.

To improve the recognition accuracy of the motion vector TSN (MV-TSN), this paper transfers the fine knowledge learned by optical flow TSN (OF-TSN) to MV-TSN through knowledge transfer strategy. The reason for this is that the motion vector and optical flow are intrinsically correlated and are similar in describing local motion. Optical flow contains fine pixel-level motion information, while the motion vector contains block-level motion information, so the latter is relatively rough and inaccurate, as shown in Figure 1. In this paper, we use the fine knowledge of OF-TSN to strengthen MV-TSN, which is called optical flow enhanced MV-TSN. The experimental results show that the proposed method cannot only realize real-time action recognition, but also achieve state-of-the-art performance. Compared with the original TSN, the proposed method can process 206.2 video frames per second, which is 13 times that of the original TSN[3]. Moreover, compared with Zhang's method[5], the recognition accuracy is increased by 5.9% and 9.3% on UCF101[7] and HMDB51[8], respectively.

The main contributions of this paper are as follows. a) Inspired by Zhang et al.[5], the motion vectors are introduced into the temporal segment network for the first time, and a real-time action recognition system with higher recognition accuracy is realized. b) Transfer the fine knowledge learned by OF-TSN to MV-TSN, which further improves the recognition accuracy.

2. RELATED WORK

Action recognition has made significant progress in the past few decades, and a large number of methods have emerged. These methods can be roughly divided into two categories: the method based on handcrafted features and the method based on deep learning. Furthermore, methods based on deep learning can be further divided into three categories:

methods based on 3D convolution network, methods based on recurrent neural network and methods based on the two-stream network.

Early work mainly focused on some methods based on hand-crafting features. Laptev et al.[9] proposes a spatiotemporal interest points (STIPs) method, which extends the Harris corner detector to three-dimensional to capture motion. Similarly, SIFT and HOG are extended to SIFT-3D[10] and HOG3D[11] respectively for action recognition. Wang et al.[12]put forward the improved dense trajectories (iDT), which is the best hand-crafting feature at present. However, these hand-crafted features are mainly aimed at small databases with relatively few types of actions and therefore perform poorly on large datasets. Moreover, because these features are designed artificially and depend on the application itself, the universal applicability and transferability is relatively poor.

In recent years, with the rapid development of deep learning in speech, image recognition, and other fields, more and more researchers have been attracted to the research of action recognition based on deep learning. At present, the methods of action recognition based on deep learning are mainly divided into three categories: the 3D convolution network, Recurrent neural network, and the two-stream architecture.

3D convolution network: Tran et al.[13] extended the convolution kernel to the time domain and constructed the network using 3D Pooling and 3D convolution, which made convolution is performed simultaneously in both spatial and temporal domains. Qiu et al. [14] proposed Pseudo-3D Residual Net (P3D ResNet). They reformed the 3D convolution by replacing $3\times3\times3$ convolutions with $1\times3\times3$ convolution filters and $3\times1\times1$convolutions filters. And these two kinds of convolution filters are used to obtain the characteristics of spatial and temporal dimensions, respectively. Temporal 3D ConvNet (T3D)[15] uses a 3D DesnseNet-based architecture and introduces a "time transition layer" (TTL) to simulate variable time convolution kernel depth.

Recurrent neural network: Donahue et al.[16] proposed a Long-term Recurrent Convolutional Neural Network (LRCN) and this is a deep hierarchical end-to-end model. In their work, the spatial features of video are extracted through a CNN network and then sent to LSTM network to extract temporal features. Veeriah et al.[17] proposed Differential LSTM that tracks the derivative of memory state by adding a new gating to the LSTM to discover patterns in salient motion patterns.

The two-stream architecture: Simonyan et al.[2] proposed a the two-stream network, consisting of spatial and temporal networks. While the spatial network takes a single RGB image as input, the and temporal network uses a stack of optical flow frames as inputs. The classification results from the spatial and the temporal network are rather fused for action recognition. Wang et al.[18] proposed Trajectory-Pooled Deep-Convolutional Descriptors (TDDs), which combines trajectory features and the two-stream network. It is a successful example of combining shallow local features with deep learning. Wang et al.[10] proposed a temporal segment network and further improved the recognition performance by using multi-modality input. At present, the method based on the two-stream network is the most efficient method to apply deep learning to action recognition. However, the main drawback of this kind of method is that the computational cost of optical flow is too high to realize real-time processing. For this reason, Zhang et al.[5] introduced the motion vectors into the original two-stream network[2] and realized a real-time action recognition system. However, because the original two-stream network cannot use the long-range temporal information in video, the recognition accuracy in Zhang's approach[5] is relatively low. In order to realize a real-time action recognition system with higher recognition performance, we introduce the motion vector into TSN[3], and further improves the recognition accuracy through knowledge transfer strategy.

3. TECHNICAL APPROACH

In this section, the proposed real-time action recognition system will be introduced in detail. We start by reviewing the knowledge of temporal segment network and motion vector. Then the real-time action recognition framework MV-TSN is introduced. Next, MV-TSN is strengthened by knowledge transfer to achieve better performance. Finally, the data augmentation strategy used in this paper is presented.

3.1 Temporal Segment Network

Temporal Segment Network (TSN)[3] is an improved network structure based on the two-stream network that was originally proposed by Simonyan and Zisserman[2]. The two-stream network approach was designed to extract the spatial and temporal information from a video. Each stream was implemented by an independent deep ConvNet, in which the spatial stream takes a single RGB image as input and the temporal stream operates on several stacked optical flow frames. The two streams are trained separately, and the final classification results are obtained by fusing the scores of two streams. But the obvious problem with the original two-stream network[2] is that it is impossible to model long-range temporal structure. The main reason is that the original two-stream network[2] is designed to operate only on a single frame (spatial network) or stacked frames (temporal network). It only pays attention to short-term action changes and cannot capture long-range temporal information of video. But in action recognition scenarios, long-range information in videos are more useful as it helps to differentiate similar actions such as dunking and shooting in a basketball game which if analyzed only on several stacked video frames could lead to misjudgments[19,20]. Therefore, TSN realizes the long-range temporal structure modeling of video sequences by using the video segmentation processing.

Firstly, a video is divided into three equal segments[3], which is expressed as $\{y_1, y_2, y_3\}$. Then, the corresponding snippet $\{x_1, x_2, x_3\}$ is obtained from each segment by random sampling. Then, the short snippets are sent to two-stream network to get the initial predicted score. Next, the initial scores of the three short snippets are fused by averaging to obtain category consensus among snippets. Finally, based on this consensus, the softmax function is used to predict the probability that the whole video belongs to each behavior category. The snippets are modeled as follows:

$$f(x_1, x_2, x_3) = S(G(F(x_1;W), F(x_2;W), F(x_3;W))) . \qquad (1)$$

Where $F(x_1;W)$ denotes a function of ConvNet with a parameter W and returns its initial prediction score of the short snippet. G denotes the category consensus function, which averages all initial scores, and S denotes the softmax function.

3.2 Motion Vector

The motion vector represents the relative movement of macroblocks between the current frame and the reference frame and is often used in various video compression standards. They can be obtained directly from the process of video decompression without additional computations. The motion vector acquires the temporal relationship between adjacent frames by the movement of macroblocks between frames, and it contains some motion information that can be used for action recognition. Since the motion vector has been computed and coded in standard compressed video, they can be obtained at a very low computational cost.

There still exist great challenges in training a high-performance MV-TSN network, mainly due to the following three reasons: 1) Not every video frame contains the motion vectors. This is because there are three types of video frames in video compression: I frame, the P frame and B frame. I frame is an intra-coded frame that does not refer to other video frames, so I frame does not contain motion information. The P frame is called predictive coding frame, which needs to refer to the previous I frame or the P frame to encode. B frame is a bidirectional predictive coding frame. Therefore, both the P frame and B frame contain time information. If empty I frames are used for action recognition, the performance may be degraded. So this paper uses the motion vector of the previous frame to replace the I frame to reduce the impact of empty I frame. 2) Compared with optical flow, the structure of the motion vector is rough and may contain inaccurate motion information. As shown in Figure 1, the optical flow is at pixel level, providing fine motion information. The motion vectors are based on macroblocks and only generate block-level motion information. Therefore, the motion vectors may lose some fine details, which will seriously affect the performance of MV-TSN. 3) The motion vectors contain more noise than optical flow. The existence of noise information and inaccurate motion information mainly because video compression algorithms need to balance the compression rate and encoding speed. Therefore, the motion vectors can only provide motion information containing noise, which hinders the final performance of the network.

3.3 Motion Vector TSN

The proposed real-time action recognition framework motion vector TSN (MV-TSN) is shown in Figure 2. It consists of video decoder and temporal segmentation network architecture. As for video decoder, compressed videos are taken as input, and the motion vector images are extracted directly from the decoding process. For the temporal segmentation network architecture, the extracted motion vectors and RGB images are sent to the TSN architecture to get the final prediction results. The main difference between the proposed method and TSN[3] is that we do not need to compute optical flow, thus avoiding the most time-consuming part of TSN. The difference between our method and Zhang's[5] is that we use TSN to solve the problem that long-range temporal information cannot be used in the original two-stream network, which further improves the recognition accuracy.

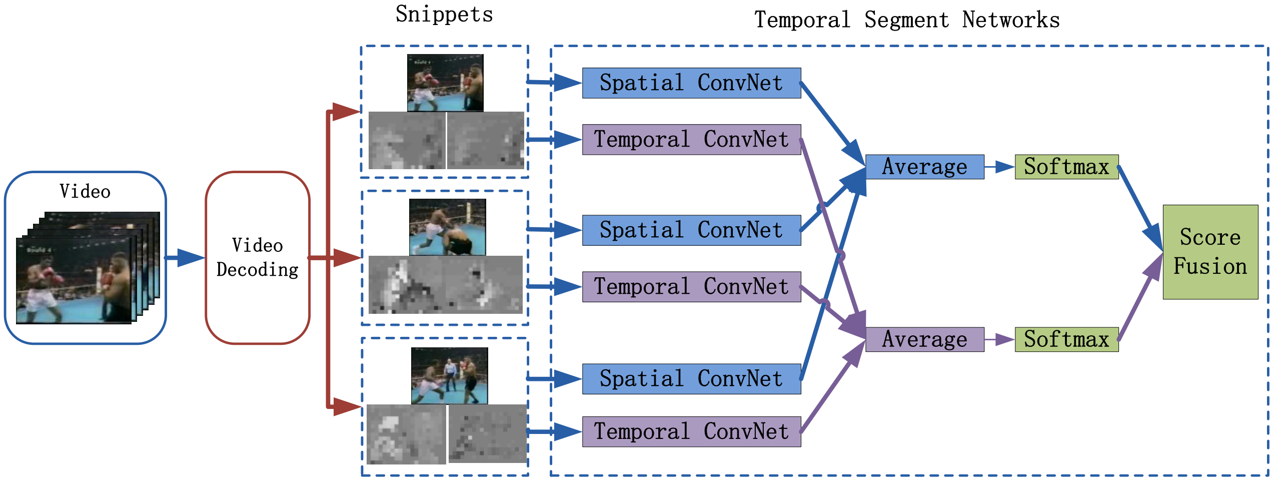

Figure 2. The proposed real-time action recognition framework MV-TSN, which consists of video decoder and TSN architecture.

3.4 Enhanced Motion Vector TSN

The paper aims at establishing an action recognition framework that is real-time and has higher performance. However, as can be seen from subsection 3.2, the motion vector lacks precise motion information and contains noise, which makes it more difficult to train high-performance MV-TSN. If the optical flow is replaced by the motion vector directly, the recognition accuracy of temporal network will be greatly degraded, which will be reduced by18.5% on UCF-101 split1. To solve this problem, this paper proposes a knowledge transfer strategy to transfer the fine knowledge learned by OF-TSN to MV-TSN to obtain enhanced MV-TSN (EMV-TSN), as shown in Figure 3. The reason for this is that optical flow and the motion vectors contain similar motion information. In the training phase, the motion vectors are initialized with the fine knowledge acquired by OF-TSN. However, in the testing phase, optical flow is not needed to be calculated, so the proposed knowledge transfer strategy will not affect the test speed of the action recognition system.

Previous work[2, 3] has shown that initializing network by using the model pre-trained on ImageNet[20] can accelerate the convergence of the network and improve the recognition accuracy. In addition, the motion vectors and optical flow are intrinsically related, and they contain similar motion information. Inspired by these two facts, this paper finds a more suitable pre-training model for the motion vectors, that is, to initialize MV-TSN with the knowledge learned by OF-TSN, which is expressed as follows:

$$\mathrm{S}_p^t = T_p^t, t = 1, 2, \cdots, n.. \tag{2}$$

$T_p = \{T_p^1, T_p^2, \cdots, T_p^n\}$ denotes the parameters learned by OF-TSN, where n denotes the total number of layers and T_p^n denotes the parameters of the n-th layer of OF-TSN. $S_p = \{S_p^1, S_p^2, \cdots, S_p^n\}$ denotes the parameters of MV-TSN and S_p^n denotes the parameters of the n-th layer of MV-TSN. MV-TSN and OF-TSN have the same network structure. After initializing the parameters learned by OF-TSN, the MV-TSN is fine-tuned with the motion vector image until convergence.

3.5 Data Augmentation Strategy

Over-fitting may occur when the number of samples in the dataset is insufficient. Data Augmentation strategy can increase the diversity of samples, prevent over-fitting and increase the robustness of the model. In the training phase, scale jittering[22], corner cropping and horizontal flipping are used to expand datasets. Scale jittering fixes the size of the input RGB or the motion vector image to 256×340, then randomly selects the width and height of the clipping area from $\{256, 224, 192, 168\}$. Finally, the clipping areas are resized to 224×224 for training. For corner cropping, we extract regions from the center or edge of the image to avoid the only use of the center of the image. It is worth noting that the random horizontal flip is used in all training steps.

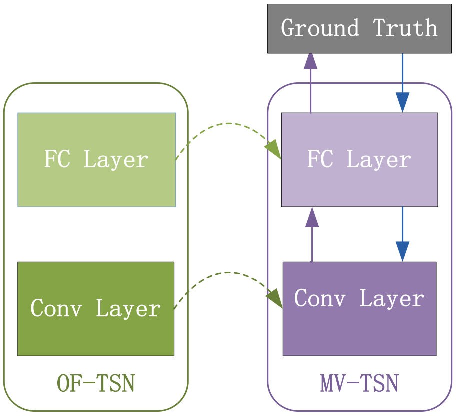

Figure 3. Transfer the fine knowledge learned by OF-TSN to the MV-TSN.

4. EXPERIMENTS

In this section, we first introduce the datasets used and the implementation details of the network in our paper. Then, the performance and the speed of the MV-TSN and EMV-TSN is evaluated on the UCF101 Split1. Finally, the proposed method is compared with the existing approaches.

4.1 Datasets

The proposed real-time action recognition system is evaluated on UCF101[7] and HMDB51[8] datasets, which are quite challenging in the field of action recognition. UCF101 contains 13320 clips which are segmented from 101 action categories. Each video contains 100-300 frames, lasting about 3-10 seconds. The fully annotated videos in UCF101 are collected from YouTube. The HMDB51 dataset contains of 6766 videos from 51 action categories, which collected from various sources, mainly from movies, and a small part from Google and YouTube videos. For both datasets, standard training/testing splits are adopted, and the evaluation protocols provided are followed. Firstly, the recognition performance and speed of MV-TSN and EMV-TSN are evaluated on split1 of UCF101 data set. Finally, the results of ours are compared with the state-of-the-art, and the time efficiency and average accuracy on three splits of UCF101 and HMDB-51 are presented respectively.

4.2 Implementation Details

During training, in order to increase the robustness of the model, three data enhancement strategies of scale jitter, horizontal flip and angle cropping are used in this paper. The details are described in section 3.5. The network parameters are learnt using mini-batch stochastic gradient descent, where the mini-batch size is set as 256 frames, the weight attenuation and momentum are set to 0.0005 and 0.9 respectively, and the L2-norm of the gradient is limited to 40. For spatial networks, the learning rate is initially set to 10^{-3}, then the rate is divided 10 per 1500 iterations. The maximum number of iterations is set to 3500. For the temporal network, the initial learning rate is set to 0.005, which is divided 10 after 10,000 and 16,000 iterations respectively, and training stopped after 18,000 iterations. Because the HMDB51 dataset is relatively small, and the video used for training is only 3.6K. To reduce over-fitting, the multi-task learning strategy [2] is used to train the temporal model on HMDB51 dataset . Following [2], the ConvNet architecture is modified. Two SoftmaxWithLoss layers are added after the last fully connected layer, one for calculating the loss on HMDB51, the other for calculating the loss on UCF101. The total losses are calculated by the sum of losses on two tasks.In addition, since there are many noises in the motion vectors of the original videos in HMDB51 dataset, this paper follows the method in [6] to re-encode the videos and extracts the motion vectors. Furthermore, OF-TSN is trained on TV-L1 optical flow[23], and all experiments are based on the Caffe[24] platform.

In the testing stage, the settings of the test protocols are the same as those in Simonyan's paper[2] that sampling 25 RGB frames or stacked motion vector frames from the test video at regular intervals. Here, each stacked motion vector frame contains 2L frames (including x, y directions) continuous motion vector images, where L = 5 frames. Then, the center and four corners of each frame are cropped and flipped to obtain 10 images as the ConvNet input. The spatial network and the temporal network are fused by weighted average. The weight ratio of the spatial network and the temporal network is set to 1:1.5. Finally, all the experiments are carried out on a CPU (E5-2620 v4) and a K80 GPU.

4.3 Evaluation of MV-TSN and EMV-TSN

In this section, the performance of MV-TSN and EMV-TSN on UCF101(split1) is analyzed experimentally. The results are summarized in Table 1.

Table 1. Comparisons of the performance of MV-TSN and EMV-TSN on UCF101 (split1)

Method	Temporal	Spatial	Fusion
OF-TSN[3]	87.7%	86.0%	93.5%
MV-TSN	69.2%	86.0%	90.9%
EMV-TSN	81.4%	86.0%	92.7%

According to the recognition rate of time network in the first column of Table 1, the performance of MV-TSN temporal network decreases significantly compared with that of OF-TSN temporal network, approximately 18.5%, which indicates that replacing Optical Flow with the motion vector will lead to the loss of fine motion in action video. Moreover, by comparing the performance of the MV-TSN temporal network and the EMV-TSN temporal network, the performance of the EMV-TSN temporal network is improvement over the MV-TSN trained from the beginning by 12.2%. The results indicate that our knowledge migration strategy is effective, which provides a good training initial point for the MV-TSN temporal network. From the fusion results of the third column spatial and temporal networks in Table 1, EMV-TSN is still superior to MV-TSN. In addition, compared with the OF-TSN, the performance of EMV-TSN is only reduced by 0.8%, but the running speed improves significantly. The evaluation of speed will be introduced in the next section.

4.4 Speed Evaluation

In this part, we evaluate the processing speeds of OF-TSN and EMV-TSN on UCF101 (split1). It is worth noting that the process of extracting the motion vector is completed by the CPU, but the feed forward process of the CNN is completed by the GPU. The processing speed of our system is measured by the number of frames per second processed. We compare the speed of extracting MV/OF, the speed of TSN feed forward process and the total speed of OF-TSN and EMV-TSN respectively, the results are shown in Table 2. It should be noted that the system will process 2L = 10 frames MV/OF and 1 frame RGB at the same time, where the center and four corners of each frame are cropped and flipped to obtain 10 images.

Table 2. Comparisons of the speed of OF-TSN and EMV-TSN on UCF101 (split1).

Method	Extract MV/OF(fps)	Feed forward calculation (fps)	Total(fps)
OF-TSN[3]	16(GPU)	578(GPU)	15.6
EMV-TSN	308(CPU)	624(GPU)	206.2

As shown in Table 2, the speed of extracting motion vectors from CPU is almost 20 times faster than that of extracting optical flow Brox's flow[4] from GPU. However, the feed forward speed of OF-TSN and EMV-TSN is similar. It can be seen that the calculation of optical flow is the main obstacle to the realization of real-time processing for TSN and other two-stream architectures. Moreover, the total processing speed of our proposed EMV-TSN is 206.2 fps, which is about eight times faster than real-time processing (25 fps). By comparison, the overall processing speed of OF-TSN is only 15.6, which cannot achieve real-time processing.

4.5 Comparison with state-of-the-art

In this section, the proposed method is compared against several state-of-the-art methods. The average recognition rate and the total processing speed of the system on three splits of UCF-101 and HMDB-51 are reported in Table 3 and 4, respectively. As can be seen in Table 3 and 4, though the accuracy of our method on UCF-101 and HMDB-51 slightly decreased over TSN[3] by 1.7% and 3.9%, respectively, the speed has increased by 190.7 fps, which is about 13 times faster than that of TSN, avoiding the calculation of optical flow in TSN and realizing real-time processing. compared with Zhang's method[5], we take advantage of information in a longer-range time, which significantly improves the recognition performance on UCF-101 and HMDB-51 by 5.9% and 9.3%, respectively. Both the processing speed and recognition accuracy of our method outperform MV + FV[6] and iDT + FV[12].

Table 3. Performance comparisons with the state-of-the-art methods on UCF101 (3 splits).

Method	**Accuracy**	**FPS**
C3D(1 net) (GPU)[13]	82.3%	313.9
C3D(3 net) (GPU)[13]	85.2%	-
MV+FV(CPU)[6]	78.5%	132.8
IDT+FV(CPU)[12]	85.9%	2.1
Two-stream(GPU)[2]	88.0%	14.3
EMV+RGB-CNN[5]	86.4%	390.7
TSN(GPU)(2 modalities)[3]	94.0%	15.6
EMV-TSN	92.3%	206.2

Table 4. Performance comparisons with the state-of-the-art methods on HMDB51 (3 splits).

Method	**Accuracy**	**FPS**
MV+VLAD(CPU)[6]	46.7%	227.8
MV+FV(CPU)[6]	46.3%	101.0
IDT+FV(CPU)[12]	57.2%	2.1
Two-stream(GPU)[2]	59.4%	14.3
DTMV+RGB-CNN[5]	55.3%	390.7
TSN(GPU)(2modalities)[3]	68.5%	15.6
EMV-TSN	64.6%	206.2

5. CONCLUSIONS

We propose the MV-TSN by introducing the motion vector into Temporal Segment Network. The MV-TSN avoids expensive calculation of optical flow, greatly speeds up the processing speed of the system, and realizes real-time action recognition. However, since the motion vector only contains block-level motion information, the lack of fine structure makes the recognition performance seriously degraded. With regard to this, a knowledge transfer strategy is proposed to enhance the performance of MV-TSN by using the fine knowledge learned from optical flow. We implement a real-time recognition system with higher recognition performance. The processing speed of the proposed method on UCF-101 and HMDB-51 is 13 times faster than that of the original Temporal Segmentation Network, and the accuracy is increased over Zhang's method[5] by 5.9% and 9.3 %, respectively.

ACKNOWLEDGMENTS

This work is supported in part by the State Key Program of NSFC under Grant 61331021 and the Program of NSFC under Grant U1804152.

REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.

[2] K. Simonyan, A. Zisserman. Two-stream convolutional networks for action recognition in videos[C]//Advances in neural information processing systems. 2014: 568-576.

[3] L. Wang, Y. Xiong, Z. Wang, et al. Temporal segment networks: Towards good practices for deep action recognition[C]//European Conference on Computer Vision. Springer, Cham, 2016: 20-36.

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In ECCV'14, pages 25–36, 2004.

[5] Zhang B , Wang L , Wang Z , et al. Real-Time Action Recognition with Deeply-Transferred Motion Vector CNNs[J]. IEEE Transactions on Image Processing, 2018:1-1.

[6] V. Kantorov and I. Laptev. Efficient feature extraction,encoding, and classification for action recognition. In CVPR'14, pages 2593–2600, 2014.

[7] K. Soomro, A. R. Zamir, M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild[J]. arXiv preprint arXiv:1212.0402, 2012.

[8] H. Kuehne, H. Jhuang, E. Garrote, et al. HMDB: a large video database for human motion recognition[C]//Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011: 2556-2563.

[9] I. Laptev. On space-time interest points. International journal of computer vision, 64(2-3):107–123, 2005. 2

[10] M.-y. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. 2009. 2

[11] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In BMVC, pages 275–1, 2008. 2

[12] H. Wang and C. Schmid. Action recognition with improved trajectories. In Proc. Conference on Computer Vision and Pattern Recognition (CVPR), pages 3551–3558, 2013. 2, 8

[13] D. Tran, L. Bourdev, R. Fergus, et al. Learning spatiotemporal features with 3d convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2015: 4489-4497.

[14] Z. Qiu, T. Yao, T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks[C]//2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017: 5534-5542.

[15] A. Diba, M. Fayyaz, V. Sharma, et al. Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification[J]. arXiv preprint arXiv:1711.08200, 2017.

[16] J. Donahue, L. Anne Hendricks, S. Guadarrama, et al. Long-term recurrent convolutional networks for visual recognition and description[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 2625-2634.

[17] V. Veeriah, N. Zhuang, G. J. Qi. Differential recurrent neural networks for action recognition[C]//Proceedings of the IEEE international conference on computer vision. 2015: 4041-4049.

[18] L. Wang, Y. Qiao, X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 4305-4314.

[19] J. C. Niebles, C. W. Chen, Fei-Fei L. Modeling temporal structure of decomposable motion segments for activity classification[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2010: 392-405.

[20] L. Wang, Y. Qiao, X. Tang. Latent hierarchical model of temporal structure for complex activity classification[J]. IEEE Transactions on Image Processing, 2014, 23(2): 810-822.

[21] J. Deng, W. Dong, R. Socher, et al. Imagenet: A large-scale hierarchical image database[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. Ieee, 2009: 248-255.

[22] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015) 1-14.

[23] C. Zach, T. Pock, H. Bischof. A duality based approach for realtime TV-L 1 optical flow[C]//Joint Pattern Recognition Symposium. Springer, Berlin, Heidelberg, 2007: 214-223.

[24] Y. Jia, E. Shelhamer, J. Donahue, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 675-678.

[25] Zhang B , Wang L , Wang Z , et al. Real-Time Action Recognition with Enhanced Motion Vector CNNs[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2016.