MEViT: Motion Enhanced Video Transformer for Video Classification

Li Li[1,2] and Liansheng Zhuang[1(✉)]

[1] University of Science and Technology of China, Hefei 230026, China
lili1234@mail.ustc.edu.cn, lszhuang@ustc.edu.cn
[2] Peng Cheng Laboratory, Shenzhen 518000, China

Abstract. Due to the advantages in extracting the long-range dependencies, self-attention based transformers are widely used to model the spatio-temporal features for video classification, which achieves competitive performance compared to 3D CNNs. To reduce the computational complexity, existing methods divide the frames into patches and factorize the spatial and temporal domains. However, most existing methods globally connect the patches at the same position in different frames to extract the temporal features, and ignore the patch motion due to video objects moving, which might hurt the performance of transformers. This paper proposes a novel architecture called Motion Enhanced Video Transformer (MEViT) for video classification, which captures patch motion information via a new module named Motion self-attention. Different from existing self-attention operation on the temporal dimension, motion self-attention globally connects the query patch and the neighborhood patches in other frames along the temporal dimension when modelling the patch temporal dependencies. Furthermore, this paper also discusses how attention blocks are stacked and how to use the spatio-temporal feature to get the classification feature. Experiments on popular public datasets (including Kinetics-400/600 and Something-Something-v2) demonstrate that our MEViT model outperforms existing dominant video transformer models.

Keywords: Video classification · Video transformer · Motion self-attention

1 Introduction

Video understanding has many real-world applications, including behavior analysis, video retrieval, and human-robot interaction. One of the most important tasks in video understanding is video classification, which is to produce a label

This work was supported in part by Next Generation AI Project of China No. 2018AAA0100602, in part to Dr. Liansheng Zhuang by NSFC under contract No. U20B2070 and No. 61976199, and in part to Dr. Houqiang Li by NSFC under contract No. 61836011. Li Li is with the School of data science.

B. Þór Jónsson et al. (Eds.): MMM 2022, LNCS 13142, pp. 419–430, 2022.
https://doi.org/10.1007/978-3-030-98355-0_35

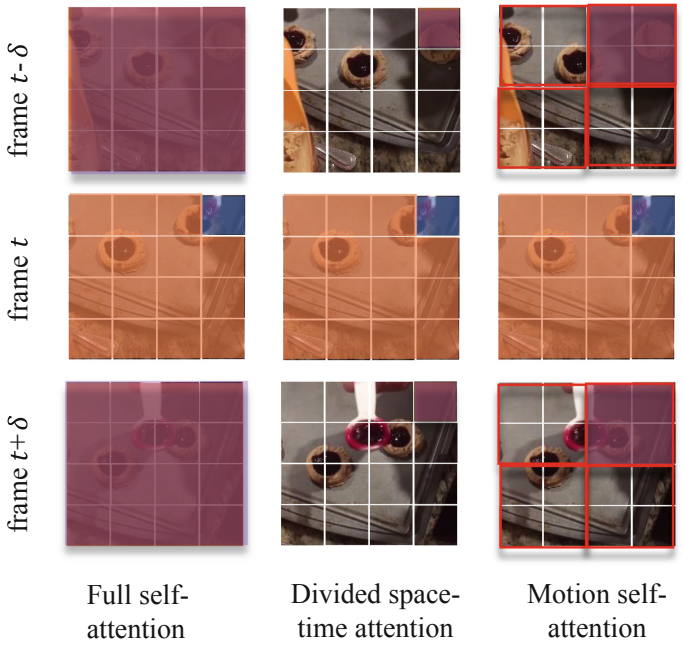

Fig. 1. Visualization of the different self-attention schemes. Each video clip is viewed as a sequence of frame-level patches which contain 16×16 pixels. For illustration, we use blue to represent the query patch, and use non-blue to display the self-attention spatio-temporal neighborhood under each scheme. Patches without color are not used for self-attention calculation of blue patch. (Color figure online)

that is relevant to the video given its frames. There are at least two challenges in video classification to overcome: how to represent the spatio-temporal information in a video and how to use the spatio-temporal information for classification. Spatio-temporal information contains two aspects: spatial information such as objects in the frame and temporal information such as correlations in different frames which is important for video classification. Previous methods used convolutional and recurrent operations to gather the information from the given video. Both convolutional and recursive operations process a local neighborhood, either in space and time; thus long-range correlation can be captured only when these operations are repeated and the signal is gradually propagated through the data.

However, there is an important defect in existing divided space-time attention, which might hurt the performance of transformer models. Current divided space-time attention concatenates the query patch and the patches located at the same position in other frames as shown in Fig. 1 with the assumption that these patches are well-aligned so that they can jointly model the motion information of some part in video. Nevertheless, due to video object moving or video camera moving, there always exist patch motions, which lead to the misalignment between the query patch and those patches from the other frames. The

misalignment may violate the performance of temporal features extracted by self-attention. Obviously, we should consider the neighbor patches from a different frame when doing self-attention, so that we can capture the relative patch motion. Inspired by this insight, this paper proposes a novel model named Motion Enhanced Video Transformer (MEViT), which can capture the motion information. MEViT divides each frame into non-overlapping patches. Several adjacent patches make up one block. Self-attention in time dimension is calculated on the same spatial block in different frames, which is named Motion self-attention. To avoid the computing cost, MEViT does not use space-time attention to jointly learn the spatial information and time information in all layers. Instead, it calculates spatial features first and then the spatio-temporal features as done in [27]. Experiments on public datasets including Kinects-400/600 and Something-something-v2 demonstrate its effectiveness.

The main contributions are summarized as follows:

- This paper proposes a new architecture named Motion Enhanced Video Transform for video classification, which can better extract temporal features for videos.
- Motion self-attention scheme is introduced to model the long-range patch dependencies, which can better capture the patch motion information due to video object moving or video camera moving.
- Extensive experiments on public datasets including Kinects-400/600 and Something-something-v2 show that our proposed MEViT outperforms state-of-the-art video transformers.

2 Related Works

Early works on video classification used hand-crafted features to encode appearance and motion information [15,22]. With the success of CNNs in image classification [14], the model for video classification is dominated by deep learning which can be broadly classified into two categories: 2D-based and 3D-based approaches. The 2D-based approaches [12,16] process each frame independently to extract frame-based features, which are then modeled by some kind of temporal model performed at the end of the network. The 3D-based approaches [8,9] are considered as the current state-of-the-art since they can typically learn stronger temporal models via 3D convolutions. However, they also incur higher computational and memory costs. To alleviate this, some works attempt to improve their efficiency via factorising convolutions across spatial and temporal dimensions or using grouped convolutions [8,21,27].

Recently, transformer-based architectures also showed promising results on large scale image classification [6]. The Vision Transformer (ViT) demonstrated the pure transformer network which is similar to the application in NLP can also obtain state-of-the-art results on ImageNet [5]. ViT has inspired a lot of follow-up work in the field of computer vision. We notice that there are many parallel methods that can extend ViT to other tasks in computer vision [3,4,11,26], and improve its data efficiency [19].

Vision transformer architectures, derived from [6], were extended through time dimension for video classification [1,2]. Because performing full space-time attention is computationally prohibitive, their main focus is on reducing computation cost via temporal and spatial factorization. In TimeSformer [2], the authors apply spatial and temporal attention in an alternating manner reducing the complexity of calculating attention weights. In a similar fashion, ViViT [1] explores several methods of space-time factorization. In addition, they also proposed to adapt the patch embedding to 3D data. Our work proposes a similar approximation to full self-attention which is also efficient. To this end, we restrict full self-attention to Motion self-attention which not only extracts the temporal features of patches from the same spatial location in other frames, but also extracts the neighborhood around that location in other frames. And this paper discusses the stacking mode of attention blocks and how to use spatio-temporal features for video classification.

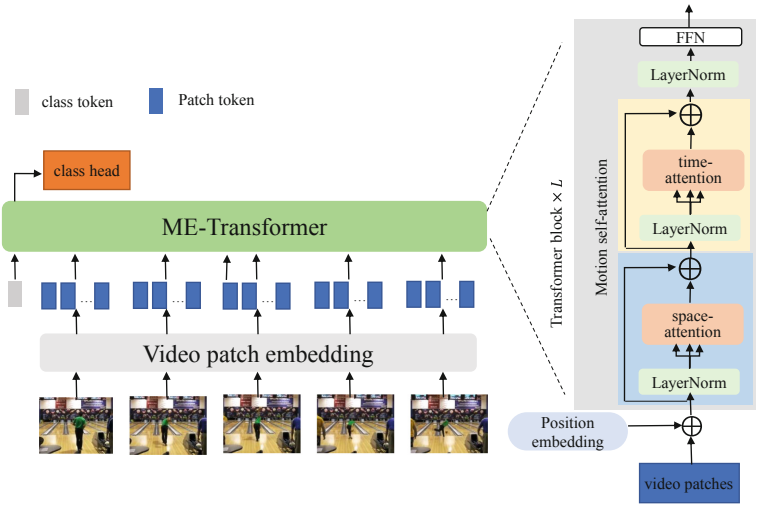

Fig. 2. Diagram of our model. The input clips is linearly projected into the patch embedding, and add the position embedding. The patches and the class token are fed into the transformer.

3 Our Method

The architecture of our model is shown in Fig. 2. It consists of the following modules: video patch embedding module, Motion Enhanced Video Transformer, Motion self-attention module, and the class embedding module.

Video Patch Embedding. The input of the model is a video clip which contains T frames sampling from the video $X \in R^{T \times H \times W \times 3}$, where H and W are

the height and width of a frame. Following the ViT approach, each frame of the input X is parted into N non-overlapping patches with the size $P \times P$. And then, the patch is reshaped into a flatten vector $x_{(p,t)} \in R^{3\times P^2}$, with p denoting the spatial locations and t denoting the index of frames. Then the vector $x_{(p,t)}$ is linearly projected into the an embedding patch token $z^{(0)}_{(p,t)} \in R^d$:

$$z^{(0)}_{(p,t)} = Ex_{(p,t)} + e^{pos}_{(p,t)}, \tag{1}$$

where $e^{pos}_{(p,t)}$ is the positional embedding and E is the trainable matrix. In order to use the transformer for video classification, a learnable vector $z^{(0)}_{(0,0)}$ named class token which represents the embedding of the classification is added in the first position of the sequence of patch tokens. The place of class token added into the transformer influences the accuracy of recognition, and we talk about the class embedding later.

Motion Enhanced Video Transformer (MEViT). Transformer consists of L encoding blocks with A heads. At each block $l \in \{1, ..., L\}$ and each head $a \in \{1, ..., A\}$, each patch token or class token is projected into query, key, and value vector by the preceding block:

$$\begin{aligned} q/k/v^{(l,a)}_{(p,t)} &= W^{(l,a)}_{Q/K/V} LN\left(z^{(l-1)}_{(p,t)}\right), \\ (p,t) \in \{(p,t)|^{p'=1,...,N}_{t'=1,...,T}\} &\cup \{(0,0)\} \end{aligned} \tag{2}$$

where LN is the LayerNorm, and d_h is hidden dim of each head. W_Q, W_K, W_V are learnable weights for embedding vector query, key, and value matrices.

The weights of self-attention are computed via dot-product. The self-attention weights $\alpha^{(l,a)}_{(p,t)} \in R^{NT+1}$ for query patch $q^{(l,a)}_{(p,t)}$ are given by

$$\alpha^{(l,a)}_{(p,t)} = SM\left(\frac{q^{(l,a)^T}_{(p,t)}}{\sqrt{d_h}} \cdot \left[k^{(l,a)}_{(0,0)} \{k^{(l,a)}_{(p',t')}\}^{p'=1,...,N}_{t'=1,...,T}\right]\right), \tag{3}$$

where SM denotes the softmax activation function. The attention weights are used as coefficients in a weighted sum over the values for each attention head:

$$s^{(l,a)}_{(p,t)} = \alpha^{(l,a)}_{(0,0)} v^{(l,a)}_{(0,0)} + \sum_{t'=1}^{T}\sum_{p'=1}^{N} \alpha^{(l,a)}_{(p,t),(p',t')} v^{(l,a)}_{(p',t')}. \tag{4}$$

These outputs from attention heads are concatenated and passed through embedding matrix W_O and the feed-forward network (FFN):

$$\begin{aligned} z'^{(l)}_{(p,t)} &= W_O \begin{bmatrix} s^{(l,1)}_{(p,t)} \\ \vdots \\ s^{(l,A)}_{(p,t)} \end{bmatrix} + z^{(l-1)}_{(p,t)}, \\ z^{(l)}_{(p,t)} &= FFN\left(LN\left(z'^{(l)}_{(p,t)}\right)\right) + z'^{(l)}_{(p,t)}. \end{aligned} \tag{5}$$

The full self-attention (3) is computed by joint space and time dimension which incurs high computational cost. A reduction in computation can be achieved by disentangling the spatial and temporal dimensions. When the attention weight is computed over one dimension, the computational cost is significantly reduced. In the case of space-attention, only $N+1$ query-key comparisons are made, using exclusively keys from the same frame as the query:

$$\alpha^{(l,a)space}_{(p,t)} = SM\left(\frac{q^{(l,a)^T}_{(p,t)}}{\sqrt{d_h}} \cdot \left[k^{(l,a)}_{(0,0)}\{k^{(l,a)}_{(p',t)}\}_{p'=1,\ldots,N}\right]\right). \tag{6}$$

The baseline time-attention proposed by TimeSformer [2] which uses of the patches from the same location as the query patch in the different frames:

$$\alpha^{(l,a)time}_{(p,t)} = SM\left(\frac{q^{(l,a)^T}_{(p,t)}}{\sqrt{d_h}} \cdot \left[k^{(l,a)}_{(0,0)}\{k^{(l,a)}_{(p,t')}\}_{t'=1,\ldots,T}\right]\right) \tag{7}$$

The full self-attention is approximated by divided space-time attention via space-attention (6) and time-attention (7).

Motion Self-attention. To extract the motion information, we need to care about not only the patches from the same location in different frames, but also the neighborhood around the location in other frames. Each frame is parted to non-overlapping blocks and each block contains $M \times M$ patches. The self-attention in time dimension is calculated by including the patch from the same spatial block in different frames:

$$\alpha^{(l,a)time}_{(p,t)} = SM\left(\frac{q^{(l,a)^T}_{(p,t)}}{\sqrt{d_h}} \cdot \left[k^{(l,a)}_{(0,0)}\{k^{(l,a)}_{(p',t')}\} \quad \begin{matrix} p' \in B \\ t' = 1,\ldots,T \end{matrix}\right]\right), \tag{8}$$

where B is the block which query patch p belongs to. The Motion self-attention uses the (8) as the time-attention layer. The model has L Motion self-attention blocks, each block has (8) and (6) orderly.

Classification Embedding. The final clip embedding is obtained from the class token of final block:

$$y = LN\left(z_{(0,0)}^{(L)}\right) \in R^d. \tag{9}$$

The class token has two purposes: guiding the self-attention learning between patches and aggregating overall information to the linear classifier [20]. Recent works have shown that separating two approaches is beneficial to the classification. We will test whether this method influences the accuracy in video classification. In our model, there are two stages: self-attention stage which updates the spatio-temporal feature of patch tokens and class-attention stage which only updates the class token. In class-attention layer, we only update the class token embedding and keep the features of patch token consistent. First, the query vectors for class token and the key/value vectors for patch tokens are calculated, and then the weight of attention and outputs of each class-attention head are calculated:

$$\begin{aligned} q_{(0,0)}^{(l,a)} &= W_Q^{(l,a)} LN\left(z_{(0,0)}^{(l-1)}\right), \\ k/v_{(p,t)}^{(l,a)} &= W_{K/V}^{(l,a)} LN\left(z_{(p,t)}^{(l-1)}\right), \\ \alpha_{(0,0)}^{(l,a)} &= SM\left(\frac{q_{(0,0)}^{(l,a)^T}}{\sqrt{d_h}} \cdot \left[k_{(p',t')}^{(l,a)}\right] \begin{matrix} p' = 1, ..., N \\ t' = 1, ..., T \end{matrix}\right), \\ s_{(0,0)}^{(l,a)} &= \sum_{t'=1}^{T} \sum_{p'=1}^{N} \alpha_{(0,0),(p',t')}^{(l,a)} v_{(p',t')}^{(l,a)}. \end{aligned} \tag{10}$$

Then, we use the (5) to calculate the $z_{(0,0)}^{(l)}$ as the output of class-attention layer.

To summarize, our model has some Motion self-attention blocks(SA), and each Motion self-attention block is composed of space-attention layer and time-attention layer orderly in Fig. 2.

4 Experiment

4.1 Experiment Setup

Datasets. We trained and evaluated the proposed models on the two widely used datasets. The Kinetics [13] dataset contains short clips sampled from YouTube. The version of the datasets used in this paper contains approximately 260k clips for Kinetics-400 and 375k clips for Kinetics-600. The SSv2 [10] dataset consists of about 220k short videos, with a length between 2 and 6 s that picture humans performing pre-defined basic actions with everyday objects. Because the backgrounds and objects of the videos are consistent in different action classes, this dataset often needs stronger temporal modeling (Fig. 3).

Network Architecture. Most of the experiments were performed using the MEViT-B/16 (L = 12, h = 12, d = 768, P = 16). For the space-attention module in the Motion self-attention, we use the pre-trained weights from ImageNet. For the time-attention layers, the block size varies from 1 to 14.

Training and Inference. For training phase, we resize the smaller dimension of each frame to a value $\in [256, 320]$, and take a random crop of size 224 × 224 from the same location for all frames of the same video. In the inference phase, we give the accuracy results for 1 × 3 views (only 1 temporal clip and 3 spatial crops) not the popular approach of using up to 10 temporal clips and 3 spatial crops. The models are implemented by pytorch, and were trained on a DGX-v1 server.

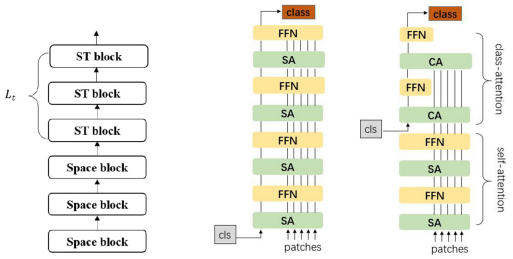

Fig. 3. The number of blocks using time-attention layer and model with class-attention layers. Space block represents the Motion self-attention only having the space-attention, and ST block represents the Motion self-attention block.

4.2 Ablation Studies

This subsection shows our proposed model with Motion self-attention can better learn the spatio-temporal features. And, we explore the importance of class token and class-attention layers. Then, we test the top-heavy transformer which uses only space-attention layers in the early transformer blocks.

Effect of the Motion Self-attention. We conduct the experiment to show the proposed Motion self-attention scheme has better performance. Table 1 shows the accuracy of our model using the Motion self-attention varying size M from 1 to 7. First, performance of our proposed Motion self-attention is superior to the origin divided space-time attention when the block size equal to 1. Second, model with Motion self-attention the block size M = 2 has the best performance. Compared to the origin divided space-time attention when block size is equal to 1, our model (M = 2) gets a bigger receptive field with more patches in the other frames in time dimension. The accuracy drops when the block size is bigger than 2, because the attention weights is calculated by more patches which might be noise.

Effect of Class-Attention Layers. For fair comparison, the total number of layers is fixed to 12. The inserted-layer i is the place where the class token is inserted into our model, i.e., our model has i self-attention blocks and $(12 - i)$ class-attention blocks. From the Table 2, we find that the architecture contains 11 self-attention blocks and 1 class-attention block gets the best performance. There is no benefit in copying the class embedding information of the class-attention block back to the patch embedding of the self-attention blocks in front process. If we keep 12 self-attention blocks, we find our model can achieve better performance by adding only one class-attention block which needs more parameters.

Table 1. Effect of the block size of motion self-attention.

Block size	Top-1	Top-5
1	78.6	93.0
2	**80.2**	**93.6**
3	80.0	93.2
7	77.6	92.8
14	77.9	92.8

Table 2. Effect of class-attention layers.

SA+CA	Inserted-layer	Top-1
9+3	9	79.4
10+2	10	80.3
11+1	11	**80.5**
12+1	12	80.6

Depth of Time-Attention Layer. Some works found that extracting the spatial information and temporal information independently is useful for video classification. We talk about the top-heavy model which keeps only the space-attention layer in the front Motion self-attention blocks, so the model has only L_t blocks with time-attention layer in Fig. 3a. The accuracy of using different numbers of time-attention layers is shown in Table 3. From the result, we can see the model with no time-attention layer has the worst performance, and model with $L_t = 8/12$ time-attention layers has the best performance. It's obvious that our model is significantly superior to the space-only attention model. Space-only attention model focuses on the spatial information and ignores the temporal information. In this situation, video classification task is regarded as object recognition. But, the first four blocks with no time-attention layer get higher accuracy. In the front block, the model calculates the attention weights from the patches in the same frame would be less affected by noise from the other frames.

Table 3. Effect of L_t. L_t denotes the number of block with the time-attention layer in our model architecture.

L_t	0	2	4	8	12
Top-1	75.6	77.5	77.9	**80.3**	80.2

4.3 Comparison with State-of-the-Art

Based on our ablation studies in the previous section, we compare to the current state-of-the-art for all mentioned datasets using our model. Our model use eleven SA layers and one CA layer, and the self-attention layers contains four space-attention layers. The results are shown in the Tables 4, 5 and 6. Unless otherwise stated, we report the results using the 1×3 views for all datasets.

For the Kinetics-400, our model training with 16 frames achieves the best performance but only using one temporal crops in the inference in the Table 4. Compared to the state-of-the-art convolution model X3D-XXL, our model brings about 0.7% gains on Top-1 accuracy, and compared to transformer-based methods, our model brings about 0.5% gains. Similarly, our model has great improvement on the Kinetics-600 in Table 5. On the SSv2, our model also matches the state-of-the-art created by the ViViT-L.

Table 4. Comparison on the Kinetics-400 dataset.

Method	Top-1	Top-5	Views
blVNet [7]	73.5	91.2	3×3
TEA [16]	76.1	92.5	10×3
TSM-R101 [17]	76.3	–	10×3
I3D NL [25]	77.7	93.3	10×3
CorrNet-101 [23]	79.2	–	10×3
LGD-R101 [18]	79.4	94.4	–
SlowFast [9]	79.8	93.9	10×3
X3D-XXL [8]	80.4	94.6	10×3
TimeSformer-L [2]	80.7	94.7	10×3
ViViT-L/16 $\times$ 2 [1]	80.6	94.7	4×3
Our model	80.6	**94.7**	1×3
Our model(16×)	81.1	**94.9**	1×3

Table 5. Comparison on the Kinetics-600 dataset.

Block size	Top-1	Top-5
AttentionNAS [24]	79.8	94.4
LGD-R101 [18]	81.5	91.6
SlowFast [9]	81.8	92.5
X3D-XL [8]	81.9	–
TimeSformer [2]	82.4	93.3
ViViT-L/16 $\times$ 2 [1]	82.5	–
Our model (16×)	**85.6**	**95.2**

Table 6. Comparison on the SSv2 dataset.

Block size	Top-1	Top-5
TRN [28]	48.8	77.6
SlowFast [9]	61.7	–
TimeSformer [2]	62.5	–
TSM [17]	63.4	88.5
TEA [16]	65.1	–
blVNet [7]	65.2	90.3
ViViT-L/16 $\times$ 2 [1]	65.4	89.8
Our model (16×)	**65.7**	**90.5**

5 Conclusion

This paper presented Motion Enhanced Video Transformer (MEViT) for video classification. Compared to existing video transformers, our model can better model the temporal features and achieve state-of-the-art performance in the video recognition datasets including Kinetics-400/600 and SSv2. It uses the proposed Motion self-attention scheme to capture the long-range patch dependencies, which considers the patch motion due to video object moving. Future efforts will be devoted to combine our approaches with other transformer architectures besides the standard ViT. Finally, we will apply our model in long-time video recognition.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: ViViT: a video vision transformer (2021)
2. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? (2021)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
4. Chen, C.F., Fan, Q., Panda, R.: CrossViT: cross-attention multi-scale vision transformer for image classification (2021)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
6. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale (2020)
7. Fan, Q., Chen, C.F.R., Kuehne, H., Pistoia, M., Cox, D.: More is less: learning efficient video representations by big-little network and depthwise temporal aggregation. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019). https://proceedings.neurips.cc/paper/2019/file/3d779cae2d46cf6a8a99a35ba4167977-Paper.pdf
8. Feichtenhofer, C.: X3D: expanding architectures for efficient video recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020
9. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast networks for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019
10. Goyal, R., et al.: The "something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5842–5850 (2017)
11. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers (2021)
12. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)

13. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc. (2012). https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
15. Laptev, I.: On space-time interest points. Int. J. Comput. Vision **64**(2), 107–123 (2005)
16. Li, Y., Ji, B., Shi, X., Zhang, J., Kang, B., Wang, L.: Tea: temporal excitation and aggregation for action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 909–918 (2020)
17. Lin, J., Gan, C., Han, S.: TSM: temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019
18. Qiu, Z., Yao, T., Ngo, C.W., Tian, X., Mei, T.: Learning spatio-temporal representation with local and global diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
19. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, pp. 10347–10357. PMLR (2021)
20. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. arXiv preprint arXiv:2103.17239 (2021)
21. Tran, D., Wang, H., Torresani, L., Feiszl, M.: Video classification with channel-separated convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019
22. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vision **103**(1), 60–79 (2013)
23. Wang, H., Tran, D., Torresani, L., Feiszli, M.: Video modeling with correlation networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020
24. Wang, X., et al.: AttentionNAS: spatiotemporal attention cell search for video classification. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12353, pp. 449–465. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58598-3_27
25. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)
26. Wu, H., et al.: CVT: introducing convolutions to vision transformers (2021)
27. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification. In: Proceedings of the European Conference on Computer Vision (ECCV), September 2018
28. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 803–818 (2018)