IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 August 2022, accepted 1 September 2022, date of publication 5 September 2022, date of current version 14 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204404

RESEARCH ARTICLE

A Novel Few-Shot Action Recognition Method: Temporal Relational CrossTransformers Based on Image Difference Pyramid

YIHANG DING AND YOUYUAN LIU
School of Artificial Intelligence, Southeast University, Nanjing 211189, China
Corresponding author: Yihang Ding (213190145@seu.edu.cn)

ABSTRACT Most current few-shot action recognition methods model temporal relationships on the basis of image classification and achieve satisfactory results. However, they focus on the extra temporal information of video data compared to images and use the frame tuple embedding representation of the query video for matching, but ignore the important information of ''action changing feature'' in action recognition. To use this information, we propose the Temporal Relational CrossTransformers Based on Image Difference Pyramid (TRX-IDP) method for few-shot action recognition. Based on TRX, we perform high-order image difference, sigmoid enhancement, resizing on the frame tuples which are directly used for query, and use the frame tuples to calculate the Motion History Image (MHI). Combined with the two, we construct the Image Difference Pyramid containing motion feature information. We also develop CrossTransformers query representation for IDP and restructure the linear mapping function of the model. We evaluate our model using four commonly used few-shot action recognition benchmark datasets. TRX-IDP achieves state-of-the-art performance on partial SSv2, HMDB51, and UCF101, while slightly lagging behind the current best models on Kinetics and SSv2. In addition, we perform detailed ablation experiments on TRX-IDP to prove the importance of each part of the model and to give the best hyperparameters of TRX-IDP.

INDEX TERMS Few-shot learning, action recognition, image difference pyramid, action feature representation.

I. INTRODUCTION

Few-shot learning has a history of decades, and its main aim is to learn a new class using only a few examples with labels, and to successfully classify the corresponding unlabeled samples. In addition, as deep learning has evolved in the field of action recognition [1], [2], [3], [4], it has been found that the video samples data set needed to collect deep learning is too large and the cost of labeling is very expensive [5]. To solve the problem of insufficient data with labeled samples, few-shot learning has been applied to the field of action recognition, and the recently proposed few-shot action recognition method [6], [7], [8], [9], [10] has achieved satisfactory results.

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

Before few-shot action recognition, few-shot image classification methods had achieved significant success, and these methods inspired Zhang *et al.* [8] to implement action recognition using a matching approach that searches a single support set of samples. Similarly, there are methods [7], [9] to search the average representation of support classes to realize action recognition. However, these methods ignore the temporal information between frames when using multiple frames to represent a video for matching. i.e., they do not use the temporal information of the video when modeling. In addition, a complete action requires two, three or more frames to represent, so using individual frames in the video to match one by one during the matching process is not the best method. Further, an action may occur anywhere in the video sample, i.e., the effect of temporal offset needs to be offset in the matching process. Moreover, the same type of action

may consume different lengths of time in different videos, offsetting the degree of stretching of the action during the matching process is also necessary.

Perrett *et al.* [10] conducted a specific study on the above problems and proposed Temporal-Relational CrossTransformers. TRX uses a part-to-part query comparison approach, using all frame tuples from the query video and all frame tuples in the support set to match one by one and calculate the average Euclidean distance, which solved the problem of action representation, speed and offset very well. However, all the few-shot action recognition methods mentioned above are implemented from the perspective of few-shot learning, attention mechanism matching, and embedding representation of frames or frame tuples, etc. These methods ignore an important piece of information in action recognition, i.e., **action changing feature**. Action changing features are different from temporal features. Consider the two action categories in the few-shot action recognition: ''throw something out'' and ''catch the flying thing''. In TRX's view, the two are the same, so adding IDP structure can effectively solve such problems of fine-grained classification.

In this paper, we give full consideration to the problems mentioned above regarding the few-shot action recognition and propose a novel method for the recognition of new-shot actions: Temporal Relational CrossTransformers Based on Image Difference Pyramids(TRX-IDP). In TRX-IDP, we propose the Image Difference Pyramid (IDP) in order to construct a representation of action changing feature. In IDP, the first layer is a set of ordered original images in the video, we perform a differential operation on the adjacent images in the first layer in order to highlight the changes in the action, and then perform contrast enhancement and pooling operations on the resulting differential image. Based on this, IDP creatively treats the image of the first layer as a complete video and calculates its Motion History Image(MHI) [11]. After building the IDP, we apply IDP to TRX and design embedding representations of different dimensions of the differential feature images of each layer in IDP, as well as design embedding representations of the MHI. In the CrossTransformer of TRX, we redesign the query, key, and value linear mapping for IDP. Finally, combining multiple TRX-IDPs of different length frame tuple, the query video is classified into the support class closest to its IDP query representation.

Our contribution can be summarized as follows:

- We propose a novel few-shot action recognition method, called Temporal Relational CrossTransformers Based on Image Difference Pyramids(TRX-IDP).
- Our proposed TRX-IDP allows for better fine-grained classification in few-shot action recognition.
- We achieve state-of-the-art results on three commonly used benchmark datasets (partial SSv2 [12], HMDB51 [13], and UCF101 [14]) for few-shot action recognition.
- We perform detailed ablation experiments on TRX-IDP to prove the importance of each part of the model and to give the best hyperparameters of model.

II. RELATED WORK

In this section, we introduce the following three areas of research that are relevant to this paper, including few-shot classification, few-shot image classification and few-shot action recognition.

A. FEW-SHOT CLASSIFICATION

In order to quickly build cognitive ability for new concepts with just one or a few examples, few-shot learning was created. So far, few-shot learning has become increasingly mature, and according to different realizing method, we broadly classify few-shot learning into three categories. Munkhdalai *et al.* [15] and Santoro *et al.* [16] proposed model-based methods. Finn *et al.* [17] and Ravi *et al.* [18] proposed optimization-based methods. Vinyals *et al.* [19] and Snell *et al.* [20] proposed metric-based methods. Among the three methods mentioned above, the metric-based learning method outperforms the other two methods in the classification of few-shot videos. The metric-based method aims to find a feature representation of the sample and calculates the distance between the query sample and the support set, and classifies the query sample to its nearest support set at the time of classification. The metric-based method is most relevant to this paper.

B. FEW-SHOT IMAGE CLASSIFICATION

In recent years, more and more people have researched numerous methods for few-shot image classification on the basis of few-shot learning. Similar to the classification of few shot learning, few-shot images classification can be classified into three categories: data-enhanced, optimization-based, and metric-based. Data augmentation is a method of expanding the sample data using spatial deformation [21] or semantic feature augmentation [22], etc. However, these operations may perform well on specific data sets and are not generalizable. Optimization-based methods learn a meta-learner model, aiming at fast convergence of model parameters and adaptation to new tasks, so that the model can classify unseen tasks in a limited number of steps. These methods include learning better model initialization parameters [23], [24]and faster gradient descent optimizer [25].The metric-based methods [26], [27], [28], [29] solve the few-shot image classification problem from the perspective of learning ''how to compare''. The network computes the Euclidean distance [20], [30] between the query image and the class in the support set, and classifies the query sample by the nearest neighbor method. In the metric-based methods, Doersch *et al.* [29] use an attention mechanism for their query image and support set, which inspire Perrett *et al.* [10] to propose the TRX method.

C. FEW-SHOT ACTION RECOGNITION

Unlike few-shot image classification, the difficulty of few-shot action recognition is that it needs to deal with 3D video data. In the above discussion, it has been shown that the

metric-based method is currently the best method, so most of the few-shot action recognition mainly uses the metric-based method. Compound Memory Network (CMN) [6] encodes the video using a composite embedding algorithm and predicts it through the memory of the CMN structure. Temporal Attentive Relation Network (TARN) [7] uses a self-attentive module to align query samples and support sets. Action Relationship Network (ARN) [8] uses a self-supervised permutation invariant method and spatial-temporal attention. Ordered Temporal Alignment Module (OTAM) [9] performs temporal alignment while using temporal features in the video data and gives a score using a distance matrix. Hybrid Relation guided Set Matching (HyRSM) [31] uses hybrid relation module and set matching metric to overcome problems in misaligned instances and loss of relevant information. Temporal-Relational CrossTransformers (TRX) [10] uses CrossTransformer to match the action feature subsequences of each query video with all subsequences in the support set and calculate the average Euclidean distance.

In the above few-shot action recognition methods, we note that the existing methods basically inherit the methods of few-shot learning and few-shot image classification. For video data, these methods use a set of video frames to represent the video, and then perform computation or matching through the embedding representation of frames or frame tuples to achieve matching at the video level. However, these methods ignore a feature inherent to action recognition: the **action changing feature**. In other words, these methods are robust enough to classify any video and can match even when the frames of the video itself are arranged in a disordered order, but if the action changing feature of action recognition is introduced, the performance of few-shot action recognition will be improved.

III. METHOD

We propose a novel action recognition method called Temporal Relational CrossTransformers Based on Image Difference Pyramid (TRX-IDP). In our method, firstly, key frames are extracted from video samples, and key frame sets are used to represent video samples. The subsequence of frames is extracted from the keyframe set, and we construct an Image Difference Pyramid for the subsequence of keyframes considering the changing features of the action, and combine it with Motion History Image (MHI) to construct a query representation for multi-CrossTransformer use. We also develop CrossTransformers query representation for IDP and rewrite and optimize the linear mapping function of the model.

We start with a special case of a triplet and proceed to build our complete approach in terms of complexity and robustness. The construction of the Image Difference Pyramid is introduced in Section III-A. Next, in Section III-B, we construct the query representation by combining the image pyramid with MHI, extracting a representation of a triplet from the query video and comparing it with the triplet representation in the support set. In Section III-C, we generalize this to multivariate representations and model a CrossTransformer for each tuple, and finally combine the matching similarity of each CrossTransformer output for classification.

A. IMAGE DIFFERENCE PYRAMID

We consider a video V and perform a keyframe extraction operation on V. Then we use the obtained keyframe sequence to represent V, i.e., $V : \{\mathrm{v}_1, \ldots, \mathrm{v}_F\}$, where v_i is the keyframe extracted from V and for $i < j$, v_i appears earlier in the video V than v_j, and F is the number of keyframes extracted from V. We define the triplet consisting of three frames selected from V as $P = \{\mathrm{v}_{01}, \mathrm{v}_{02}, \mathrm{v}_{03}\}$.

We perform the difference operation on v_{0i} and $\mathrm{v}_{0(i+1)}$ to get the first-order difference image $diff_{1i} = |\mathrm{v}_{0i} - \mathrm{v}_{0(i+1)}|$. Then we use the *TemperatureSigmoid* function to enhance the contrast of the differential image and finally rescale the differential image $TS(diff_{1i})$, the purpose of rescaling is not only to reduce the complexity to linearity, but also to reduce the number of invalid features. where the rescaling operation is like the average pooling of 2×2, the stride is 2, and the TS function is:

$$TS(x) = 255/(1 + e^{-0.05(x-127.5)}), \tag{1}$$

where the hyperparameters are selected based on common image contrast enhancement functions.

Thus we get the i-th image of the first-order differential of the pyramid:

$$\mathrm{v}_{1i} = rescale(TS(|\mathrm{v}_{0i} - \mathrm{v}_{0(i+1)}|)). \tag{2}$$

For a difference of order k, $1 < k < F, k < K$, K is the highest layer number of the pyamid, there are:

$$\mathrm{v}_{ki} = rescale(TS(|\mathrm{v}_{(k-1)i} - \mathrm{v}_{(k-1)(i+1)}|)), \tag{3}$$

where i satisfies $i \leq F - k$. For the case where $F = 3, K = 3$, is shown in Fig 1.

MHI [11] represents the target motion as image brightness by calculating the pixel changes at the same location during the time period. We creatively treat the sequence of a few frames as video and calculate its MHI. Let H be the intensity value of the motion history pixel and $H(x, y, t)$ can be calculated from the update function as:

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } \Psi(x, y, t) = 1 \\ \max(0, H_\tau(x, y, t-1) - \delta) & \text{otherwise,} \end{cases} \tag{4}$$

where (x, y) and t are the positions and times of the pixel points, $t \geq 1$, $H_\tau(x, y, 0) = 0$; τ is the duration, which determines the time range of the motion from the perspective of the number of frames, and here $\tau = 250$; δ is the recession parameter, and here $\delta = 100$. $\Psi(x, y, t)$ is the update function, defined using the inter-frame difference method:

$$\Psi(x, y, t) = \begin{cases} 1 & \text{if } D(x, y, t) \geq \xi \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where:

$$D(x, y, t) = |I(x, y, t) - I(x, y, t-1)|, \tag{6}$$

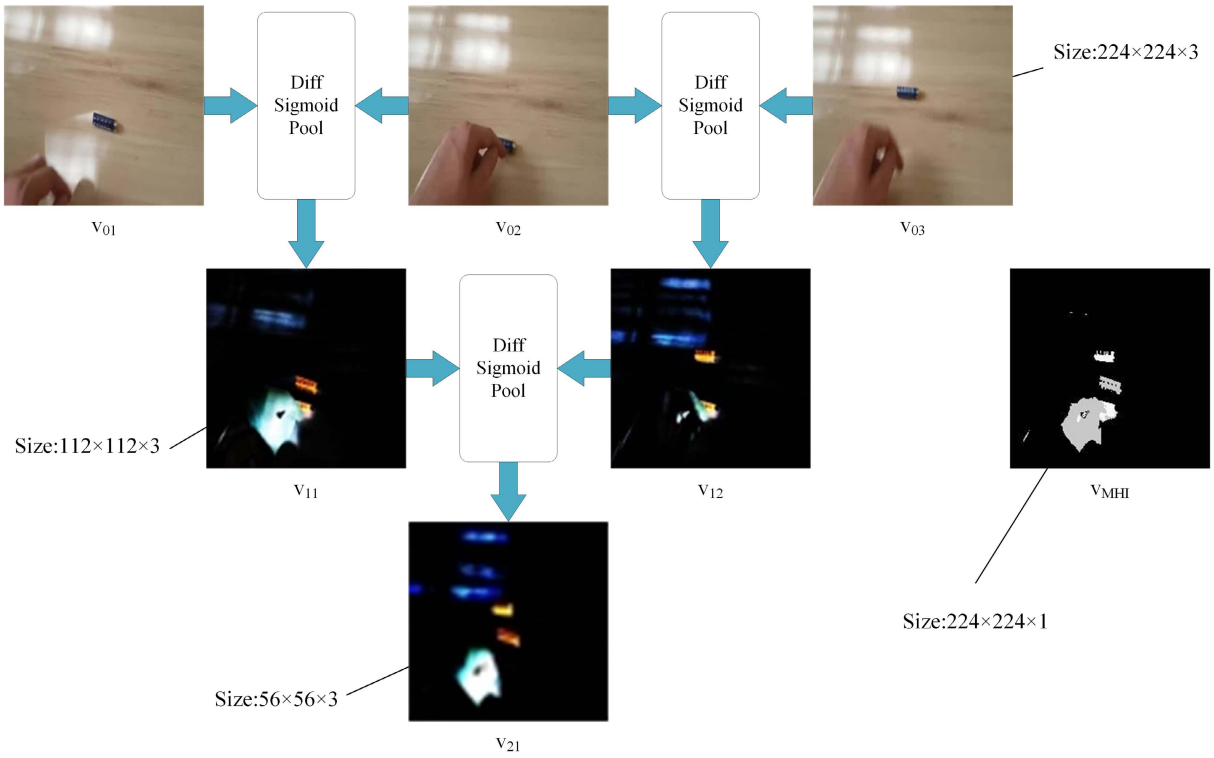

FIGURE 1. Example for an Image Difference Pyramid, when $K = 3, \Omega = \{3\}$. The original images in the first layer are selected from the SSv2 dataset, and its class is "Letting something (a battery) roll along a flat surface." The three original images are ordered from left to right with the size of 224 × 224 × 3, and the adjacent images are differenced, sigmoid enhanced, and resized to obtain the two images of the second layer with the size of 112 × 112 × 3, and the third layer with the size of 56 × 56 × 3. In addition, IDP includes MHI characteristic map, which is calculated from three images on the first layer. The size of MHI is 224 × 224 × 1.

where $I(x, y, t)$ is the intensity value of the pixel point at the coordinate (x, y) of the t-th frame of the video image sequence, ξ is the artificially given difference threshold, and here $\xi = 75$.

B. TEMPORAL CrossTransformer BASED ON IMAGE DIFFERENCE PYRAMID

1) PROBLEM FORMULATION

In few-shot action recognition, the purpose of the task is to train a neural network. It can classify an unlabeled query video into one of several classes, each class consisting of samples that are labeled and not used in training, called "support sets". In this paper, we draw few-shot action recognition tasks from the training set, and for each task, we focus on its C-way, N-shot classification problem.

We consider three frames sampled from the query video $Q : \{q_1, \ldots, q_F\}$ to represent an action feature, and we define the index of these three frames as $p = (p_1, p_2, p_3)$, where $1 \leq p_1 < p_2 < p_3 \leq F$. According to the definition in section III-A, we construct an Image Difference Pyramid for the sequence of these three frames, and the pyramid has three layers in total, where first layer is the original three frames. Then we use the pyramid to construct the query representation Q_p for use by the CrossTransformer. for the first layer of the pyramid, i.e., the original image, we define its query representation as:

$$Q_{p0} = [\Phi_0(q_{01}) + \mathrm{PE}(p_1), \Phi_0(q_{02}) + \mathrm{PE}(p_2), \Phi_0(q_{03}) + \mathrm{PE}(p_3)] \in \mathbb{R}^{3\times D}, \tag{7}$$

where $\Phi_0 : \mathbb{R}^{H\times W\times 3} \mapsto \mathbb{R}^D$ is a convolutional neural network layer that transforms the input frame into a D-dimensional embedding, $PE(\cdot)$ is position encoding based on the index of the frame, and q_{0i} is the i-th image of the 0-th order difference layer (the first layer is the 0-th order difference layer) of the Image Difference Pyramid formed by the extracted frame tuple (q_{p1}, q_{p2}, q_{p3}).

The TRX method pioneeringly uses ordered frame tuples to represent actions, but ignored the changing features of the actions themselves. Our proposed Image Difference Pyramid highlights the action features that are missed during frame tuple matching, and we define the query representation of the k layers of the pyramid as:

$$Q_{pk} = [\Phi_k(q_{k1}), \ldots, \Phi_k(q_{ki})] \in \mathbb{R}^{i\times D/4^i}, \tag{8}$$

where $i = 3 - k, i \geq 1, k < 3, \Phi_k : \mathbb{R}^{H\times W\times 3/4^k} \mapsto \mathbb{R}^{D/4^k}$. For MHI images q_{MHI} there are Q_{pMHI} queries expressed as:

$$Q_{pMHI} = [\Phi_{MHI}(q_{MHI})], \tag{9}$$

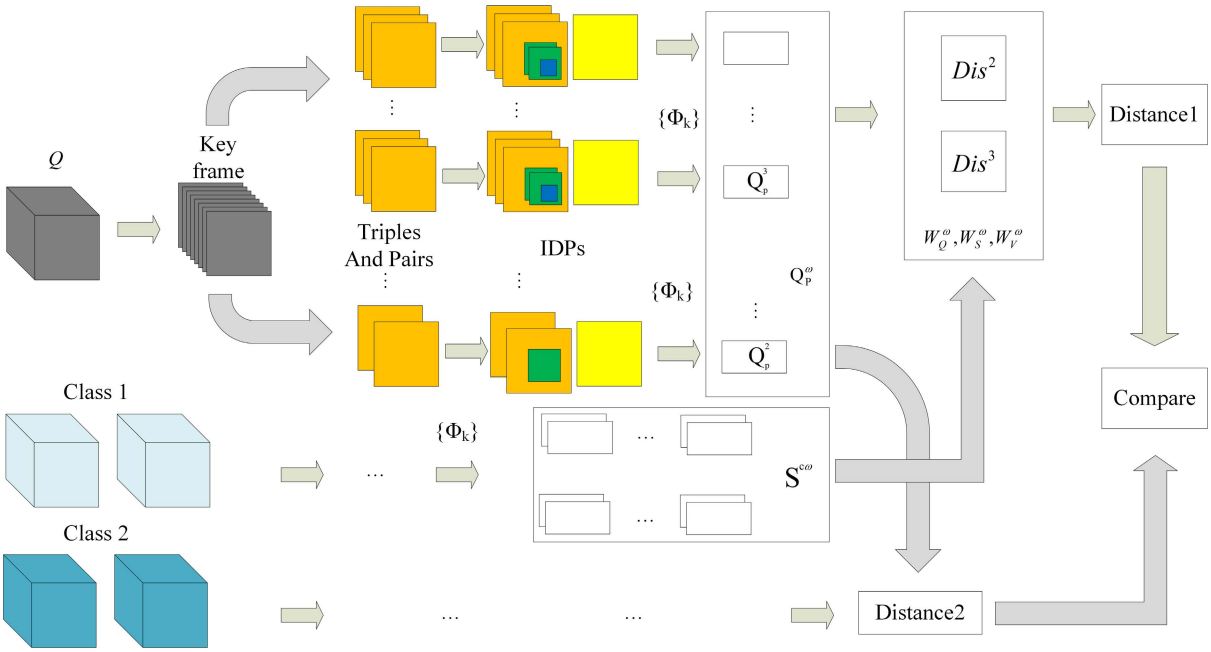

FIGURE 2. Example for TRX-IDP on a 2-way 2-shot problem, where $\Omega = \{2, 3\}$, $K = 3$. Firstly, extract the key frames of the query video Q (extract 8 frames, i.e. f=8), then arrange and combine the key frames to obtain different pair frames and triplet frames, and then calculate the IDP of different tuples. $\{\Phi_k\}$ is used to embed and encode different IDPs to get the query embedded representation Q_p^{ω} of video Q. Similarly, calculate the embedded representation $S^{c\omega}$ of the support set, match $S^{c\omega}$ and Q_p^{ω} through different Tom, get the Euclidean distance from Q to class c, and finally classify Q as the nearest class.

where $\Phi_{MHI} : \mathbb{R}^{H\times W} \mapsto \mathbb{R}^{D/3}$. q_{MHI} is the MHI feature figure generated from the three images of the pyramid first layer.

In summary, we define the query representation of Q_p as follows:

$$Q_p = [Q_{p0}, Q_{p1}, Q_{p2}, Q_{pMHI}] \in \mathbb{R}^{(3+1)\times D+2\times D/4+1\times D/4^2}. \quad (10)$$

We compare the query representation Q_p with all the triplet representations in the support set, allowing to match actions with different speeds or appearing in different locations in the video.We define the set of all triples as:

$$\Pi = \{(\pi_1, \pi_2, \pi_3) \in \mathbb{N}^3 | 1 \le \pi_1 < \pi_2 < \pi_3 \le F\}. \quad (11)$$

Using the same method as for (7)-(10), we define the representation of a triplet indexed by $m = (m_1, m_2, m_3) \in \Pi$ in video n of class c as:

$$S_{nm}^c = [S_{nm0}^c, S_{nm1}^c, S_{nm2}^c, S_{nmMHI}^c] \in \mathbb{R}^{(3+1)\times D+2\times D/4+1\times D/4^2}. \quad (12)$$

The set of all triplet representations in the support set of class c is:

$$\mathbf{S}^c = \{S_{nm}^c | 1 \le n \le N, m \in \Pi\}. \quad (13)$$

We apply the query representation generated using the image pyramid to the Temporal CrossTransformer. The CrossTransformer includes the query representation mapping W_Q, key representation mapping W_S and value representation mapping W_V, which are shared across classes:

$$W_Q, W_S : \mathbb{R}^{(3+1)\times D+2\times D/4+1\times D/4^2} \mapsto \mathbb{R}^{d_k} \quad \text{and}$$
$$W_V : \mathbb{R}^{(3+1)\times D+2\times D/4+1\times D/4^2} \mapsto \mathbb{R}^{d_v}. \quad (14)$$

The correspondence between Q_p and S_{nm}^c can be expressed as:

$$a_{nmp}^c = L(W_S \cdot S_{nm}^c) \cdot L(W_Q \cdot Q_p), \quad (15)$$

where L is a layer normalisation. Normalize a_{nmp}:

$$\tilde{a}_{nmp}^c = \frac{\exp\left(a_{nmp}^c\right)/\sqrt{d_k}}{\sum_{\alpha,\beta}\exp\left(a_{\alpha\beta p}^c\right)/\sqrt{d_k}}. \quad (16)$$

Value embeddings of the support set and of the query are as follows:

$$\mathbf{e}_{nm}^c = W_V \cdot S_{nm}^c \text{ and } t_p = W_V \cdot Q_p. \quad (17)$$

We combining normalized correspondence $\tilde{a}_{nmp}^c$ and support set embedding $\mathbf{e}_{nm}^c$:

$$\mathrm{u}_p^c = \sum_{nm} \tilde{a}_{nmp}^c \mathbf{e}_{nm}^c. \quad (18)$$

Then we can calculate the distance between the query representation Q_p and the support set S^c:

$$distance\left(Q_p, \mathbf{S}^c\right) = \left\|\mathbf{u}_p^c - \mathbf{t}_p\right\|. \quad (19)$$

Obviously a frame triplet does not represent an action very well.Therefore it is necessary to use multiple query representations for comparison.We define all queries to be represented as:

$$\mathbf{Q} = \{Q_p | p \in \Pi\}. \tag{20}$$

So far, the distance between the query Q and the support set S^c is defined as:

$$Distance(\mathbf{Q}, \mathbf{S}^c) = \sum_{p \in \Pi} distance(Q_p, \mathbf{S}^c). \tag{21}$$

C. TEMPORAL-RELATIONAL CrossTransformers BASED ON IMAGE DIFFERENCE PYRAMID

Considering that a frame triplet may not be the best representation of an action, the using of higher-order tuples is necessary. We use ω to represent the length of the tuple as TRX does. Rewrite Π in (11):

$$\Pi^\omega = \left\{(\pi_1, \ldots, \pi_\omega) \in \mathbb{N}^\omega | \forall i \, (1 \leq \pi_i < \pi_{i+1} \leq F)\right\}. \tag{22}$$

Generalize the query representation Q_p with index $p = (p_1, \ldots, p_\omega) \in \Pi^\omega$:

$$Q_p^\omega = [Q_{p0}, Q_{p1}, \ldots, Q_{p(\omega-1)}, Q_{pMHI}] \in \mathbb{R}^{Dim(\omega) \cdot D}, \tag{23}$$

where:

$$Dim(\omega) = \frac{4}{3}\omega + \frac{4^{(1-\omega)}}{9} - \frac{1}{9}. \tag{24}$$

We define the set of tuple lengths ω as Ω. For instance, $\Omega = \{2, 4\}$ represents pairs and quadruples of frames tuples. For different ω, query representation mapping W_Q, key representation mapping W_S and value representation mapping W_V in (14) are rewritten as:

$$\begin{aligned} W_Q^\omega, W_S^\omega &: \mathbb{R}^{Dim(\omega) \cdot D} \mapsto \mathbb{R}^{d_k} \quad \text{and} \\ W_V^\omega &: \mathbb{R}^{Dim(\omega) \cdot D} \mapsto \mathbb{R}^{d_v}. \end{aligned} \tag{25}$$

Combining the Temporal CrossTransformer based on Image Difference Pyramid(TX-IDP) corresponding to the different ω, we obtain the distance between the query Q and the support set S^c in general form, i.e. Temporal-Relational CrossTransformer based on Image Difference Pyramid(TRX-IDP):

$$\mathbf{TP}^\Omega(Q, \mathbf{S}^c) = \sum_{\omega \in \Omega} \frac{Distance^\omega(\mathbf{Q}^\omega, \mathbf{S}^{c\omega})}{|\Pi^\omega|}. \tag{26}$$

We classify the query Q as class c which is closest to it:

$$c = argmin_c \mathbf{TP}^\Omega \left(Q, \mathbf{S}^c\right). \tag{27}$$

1) SUMMARY OF METHOD

TRX-IDP considers frame tuple representations of different lengths, and for different ω, it needs to train different linear mappings. For different difference orders k, model also needs to train different input frame embeddings $\{\Phi_k\}$. The network uses a single cross-entropy loss and back-propagates the TRX-IDP network corresponding to each different om using the gradient of the sum distance. TRX-IDP is trained end-to-end using all $\omega \in \Omega$, different differential orders k and shared backbone parameters for all tuples. Fig. 2 shows an example of TRX-IDP.

IV. EXPERIMENTS

In this section, we first introduce the data sets used in the experiment and the experimental details such as model parameters. Then, we compare our model with other state-of-the-art models. Finally, we perform a detailed ablation study of the model to demonstrate the validity of our proposed method.

A. DATASETS AND EXPERIMENTAL SETUP

1) DATASETS

In our experiments, we use four datasets commonly used in the field of action recognition to evaluate our model, which are Kinetics-400 [32], Something-Something V2 (SSv2) [33], HMDB51 [13], and UCF101 [14], where SSv2 has two versions, full [9] and partial [12]. In the above four data sets, SSv2 has proven to be the most challenging in [34], [35]. For Kinetics-400 and SSv2 datasets, we used the same split as [6] and [9], i.e. select 100 classes from the data set, and then select 64 classes from these 100 classes as the training set, 12 classes as the verification set and 24 classes as the test set. For UCF101 and HMDB51, we evaluate our model using the splitting method from [8] and [10].

2) IMPLEMENTATION DETAILS

As in the previous works [9], [10], we use Resnet-50 [36] as the backbone and pre-train the weights using ImageNet [37]. We randomly initialize the parameters of the model (for $\Phi_k, k \neq 0$ we initialize it to 0) and set $D = 2048, d_v = 1152 = d_k$. We extract 8 keyframes from the video, i.e., $F = 8$, and then resize the resulting keyframes to 224×224. In addition, TRX-IPD selects SGD as the optimizer and sets the learning rate to 10^{-3}(when the data set is partial SSv2, the learning rate is set to 10^{-4}).

B. COMPARISON WITH STATE-OF-THE-ART METHODS

1) BASELINES AND EVALUATION

We compare TRX-IDP with several recent few-shot action recognition methods [6], [7], [8], [9], [10], [12], [31], which were introduced in Section II. The TRX-IDP method inherits the characteristics of the TRX method and performs better on the few-shot task than on the one-shot task, and to facilitate comparison with the other methods mentioned above, we evaluate our method using the standard 5-way 5-shot benchmark.

We present the results of TRX-IDP and other model performance in Table 1. The models in Table 1 all use ResNet-50 as the backbone to extract features. On Kinetics, the accuracy of OTAM has been as high as 85.8%, and TRX has improved 0.1% compared to the next, while the best model HyRSM now reaches 86.1%, our TRX-IDP has only 86.0% accuracy

TABLE 1. Results on 5-way 5-shot benchmarks of Kinetics [32], SSv2 [33], HMDB51 [13] and UCF101 [14]. In the table, we have bolded the highest accuracy values on each data set.

Method	Kinetics-400	full SSv2 [9]	partial SSv2 [32]	HMDB51	UCF101
CMN [6]	78.9	-	-	-	-
CMN-J [12]	78.9	-	48.8	-	-
TARN [7]	78.5	-	-	-	-
ARN [8]	82.4	-	-	60.6	83.1
OTAM [9]	85.8	52.3	-	-	-
TRX [10]	85.9	64.6	59.1	75.6	96.1
HyRSM [31]	**86.1**	**69.0**	56.1	76.0	94.7
TRX-IDP(Ours)	86.0	67.1	**59.8**	**76.5**	**96.3**

and does not surpass HyRSM, This is due to the fact that when Kinetics is used as a few-shot benchmark, it is similar to some image classification tasks, as the video data the temporal information and action features are not important. On SSv2 dataset, where temporal information is extremely important, OTAM achieves a performance of 52.3% for the full SSv2 dataset, TRX models the temporal relationships in comparison and thus achieves an accuracy of 64.6%, HyRSM further improves to 69.0%, and TRX-IDP achieves 67.1%, which is 1.9% behind HyRSM in comparison. In contrast, on partial SSv2, there are different results, with TRX reaching 59.1%, while HyRSM lags 3.0%, and our TRX-IDP outperforms the existing method with a performance of 59.8%. In addition, our model also achieves the highest classification accuracy on HMDB51 and UCF101, reaching 76.5%, and 96.3%, respectively. where since UCF101 and Kinetics are similar and both belong to appearance-based datasets, the improvements we make in terms of action features compared to TRX do not result in significant performance gains. Overall, the performance of our proposed TRX-IDP outperforms the original TRX on all datasets, and achieves outperformance on some SSv2, HMDB51 and UCF101 compared to HyRSM, while slightly underperforming HyRSM on full SSv2 and Kinetics.

C. ABLATION STUDY

In this section, we perform a detailed ablation study of TRX-IDP to derive the optimal hyperparameters of the model while showing the effect of each module of the model. We will evaluate the Impact of MHI on TRX in section IV-C1, the Impact of the highest pyramid layer number K on the model performance in section IV-C2, and the Impact of the model parameter Ω on TRX-IDP classification accuracy in section IV-C3.

1) IMPACT OF MHI ON TRX AND LENGTH OF TUPLE

We evaluate the impact of the separate MHI module in TRX-IDP on TRX using partial SSv2. Our comparison results are reported in Fig. 3, where we selected four cases: $\Omega = \{1\}, \{2\}, \{3\}$, and $\{4\}$. Since the MHI feature map requires at least two frames to be generated, the MHI is a zero matrix

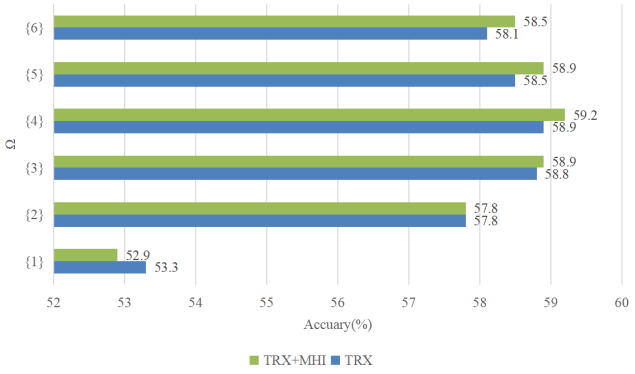

FIGURE 3. The comparison results of TRX+MHI and TRX, as the tuple length becomes longer, MHI gradually has a positive effect on the model, and the longer the tuple, the greater the effect.

when the frame tuple length is 1. The MHI feature figure plays a positive impact when the tuple length reaches 3 and a negative impact when the tuple length Less than or equal to 2, and the performance gain is higher as the tuple length gets longer. When the tuple length equals 4, the performance improvement reaches 0.4%. Therefore, when introducing the MHI, it should be ensured that the tuple length is greater than 1, and the larger the number of tuples, the greater the amount of information contained in the MHI. In addition, when the tuple length is larger than 4, the performance of the model decreases instead. Therefore, in section IV-C3, we will not discuss the case where the tuple length is greater than 4.

2) IMPACT OF K: THE HIGHEST LAYER NUMBER OF THE PYRAMID

Similar to the method used to evaluate the impact of MHI on the model, we use Kinetics and partial SSv2 to evaluate the impact of independent differential images on TRX. Since the number of pyramid layers is necessarily less than or equal to the tuple length, we choose the case where $\Omega = \{4\}$ to experiment on K. Fig. 4 demonstrates the effect of K on the image difference-based TRX. When $k = 1$, the model degenerates to TRX, and it can be seen that the classification accuracy

TABLE 2. Comparing all values of Ωfor TRX-IDP. In the table, we have bolded the highest accuracy values on each data set.

set of tuple length	Kinetics(TRX-IDP)	partial SSv2(TRX-IDP)	Kinetics(TRX)	partial SSv2(TRX)
$\Omega = \{1\}$	85.0	52.9	85.2	53.3
$\Omega = \{2\}$	85.0	58.1	85.0	57.8
$\Omega = \{3\}$	85.8	59.2	85.6	58.8
$\Omega = \{4\}$	84.8	59.6	84.5	58.9
$\Omega = \{2, 3\}$	**86.0**	**59.8**	**85.9**	**59.1**
$\Omega = \{2, 4\}$	84.6	58.9	84.4	58.4
$\Omega = \{3, 4\}$	85.6	59.7	85.3	**59.1**
$\Omega = \{2, 3, 4\}$	85.5	59.7	85.3	58.9

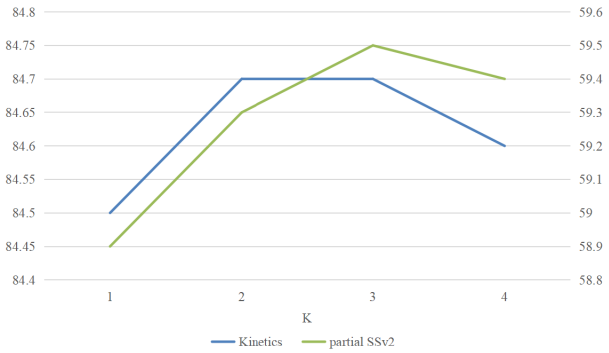

FIGURE 4. Impact of the highest pyramid layer number K on the model at $\Omega = 4$. The model performs best for $K = 3$ on both data sets.

of the model is lowest in this case, while when $k > 1$, the performance of the model is higher than the original TRX in all cases, which proves that the differential feature images can bring a positive impact on the matching process of TRX. Notice that the accuracy of the model decreases when K rises to 4, which is caused by the reduction of effective information in the higher-order differential images and the loss of motion features during successive differencing. We can see in Fig. 4 that on Kinetic, the performance of the model is 84.7% when $K = 2, 3$, while on SSv2, the model has the highest accuracy when $K = 3$: 59.5%. Therefore, we consider that the most suitable K for TRX-IDP is 3.

3) IMPACT OF Ω: THE SET OF TUPLE LENGTH

We have discussed the impact of HMI and the highest layer number K on the model and the optimal value of K in the above two subsubsections, In this part, we continue to study the impact of tuple length set on the model. In Table 2, the highest layer number K is set to 3. We can find that when using ssv2 data set for evaluation, the performance is significantly improved by 5.2% from single frame $\Omega = \{1\}$ to pair frames $\Omega = \{2\}$. When using triplet frames, the performance is further improved by 1.1%, while the growth of quadruple frames is slowed down (increase by 0.4%). When combining the two CrossTransformers, the overall accuracy has been improved, and the combination of pair frames and triplet frames has achieved the best result: 59.8%. Combining pair frames and quadruples is not a good choice, because its performance (58.9%) is even inferior to that of a single CrossTransformer (triplets and quadruples). When using three CrossTransformers $\Omega = \{2, 3, 4\}$, the performance is reduced (-0.1%). When K is used for evaluation, the overall difference is small, but the same conclusion can be obtained as when SSv2 is used: the combination of pair frame and triplet frame i.e. $\Omega = \{2, 3, 4\}$ is the best choice. Compared with TRX, IDP has improved the overall performance of TRX.

V. CONCLUSION

In this paper, we propose the Temporal Relational CrossTransformers Based on Image Difference Pyramid (TRX-IDP) method for few-shot action recognition. Our method is based on TRX. On this basis, the frame tuples used for query are subjected to high-order image difference, sigmoid enhancement and resizing. Combined with Motion History Image (MHI), the Image Difference Pyramid (IDP) containing motion feature information is constructed. We also develop the CrossTransformers query representation for IDP and rewrite and optimize the linear mapping function of the model. the TRX-IDP method outperforms TRX on few-shot benchmarks for all four datasets and achieves state-of-the-art performance on partial SSv2, HMDB51, and UCF101, while slightly lagging behind HyRSM on Kinetics-400 and full SSv2. In the future, we will try to combine the IDP module with other metric-based few-shot action recognition methods and explore them.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 568–576.

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

[3] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.

[4] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 20–36.
[5] M.-L. Antonie, O. R. Zaiane, and A. Coman, "Application of data mining techniques for medical image classification," in *Proc. 2nd Int. Conf. Multimedia Data Mining*, 2001, pp. 94–101.
[6] L. Zhu and Y. Yang, "Compound memory networks for few-shot video classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 751–766.
[7] M. Bishay, G. Zoumpourlis, and I. Patras, "TARN: Temporal attentive relation network for few-shot and zero-shot action recognition," 2019, *arXiv:1907.09021*.
[8] H. Zhang, L. Zhang, X. Qi, H. Li, P. H. Torr, and P. Koniusz, "Few-shot action recognition with permutation-invariant attention," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 525–542.
[9] K. Cao, J. Ji, Z. Cao, C.-Y. Chang, and J. C. Niebles, "Few-shot video classification via temporal alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10618–10627.
[10] T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, and D. Damen, "Temporal-relational CrossTransformers for few-shot action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 475–484.
[11] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.
[12] L. Zhu and Y. Yang, "Label independent memory for semi-supervised few-shot video classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 273–285, Jul. 2020.
[13] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.
[14] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[15] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2554–2563.
[16] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1842–1850.
[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.
[18] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
[19] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.
[20] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4080–4090.
[21] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*.
[22] Z. Chen, Y. Fu, Y. Zhang, Y.-G. Jiang, X. Xue, and L. Sigal, "Multi-level semantic feature augmentation for one-shot learning," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4594–4605, Sep. 2019.
[23] A. Antoniou, H. Edwards, and A. Storkey, "How to train your MAML," 2018, *arXiv:1810.09502*.
[24] M. A. Jamal and G.-J. Qi, "Task agnostic meta-learning for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11719–11727.
[25] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to learn quickly for few-shot learning," 2017, *arXiv:1707.09835*.
[26] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu, "Negative margin matters: Understanding margin in few-shot classification," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 438–455.
[27] C. Simon, P. Koniusz, R. Nock, and M. Harandi, "Adaptive subspaces for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4136–4145.
[28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.
[29] C. Doersch, A. Gupta, and A. Zisserman, "CrossTransformers: Spatially-aware few-shot transfer," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21981–21993.
[30] S. W. Yoon, J. Seo, and J. Moon, "TapNet: Neural network augmented with task-adaptive projection for few-shot learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7115–7123.
[31] X. Wang, S. Zhang, Z. Qing, M. Tang, Z. Zuo, C. Gao, R. Jin, and N. Sang, "Hybrid relation guided set matching for few-shot action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 19948–19957.
[32] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.
[33] R. Goyal, "The 'something something' video database for learning and evaluating visual common sense," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5842–5850.
[34] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 803–818.
[35] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "STM: SpatioTemporal and motion encoding for action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2000–2009.
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

YIHANG DING is currently pursuing the bachelor's degree with the School of Artificial Intelligence, Southeast University, China. His research interests include machine learning, computer vision, and action recognition.

YOUYUAN LIU is currently pursuing the bachelor's degree with the School of Artificial Intelligence, Southeast University, China. His research interests include machine learning, image classification, and computer vision.