Journal of Ambient Intelligence and Humanized Computing (2023) 14:7159–7172
https://doi.org/10.1007/s12652-021-03567-1

ORIGINAL RESEARCH

Action recognition based on dynamic mode decomposition

Shuai Dong[1] · **Weixi Zhang**[2] · **Wei Wang**[2] · **Kun Zou**[1]

Received: 25 January 2021 / Accepted: 17 October 2021 / Published online: 27 October 2021

Abstract
Based on dynamic mode decomposition (DMD), a new empirical feature for quasi-few-shot setting (QFSS) skeleton-based action recognition (SAR) is proposed in this study. DMD linearizes the system and extracts the modes in the form of flattened system matrix or stacked eigenvalues, named the DMD feature. The DMD feature has three advantages. The first advantage is its translational and rotational invariance with respect to the change in the localization and pose of the camera. The second one is its clear physical meaning, that is, if a skeleton trajectory was treated as the output of a nonlinear closed-loop system, then the modes of the system represent the intrinsic dynamic property of the motion. Finally, the last one is its compact length and its simple calculation without training. The information contained by the DMD feature is not as complete as that of the feature extracted using a deep convolutional neural network (CNN). However, the DMD feature can be concatenated with CNN features to greatly improve their performance in QFSS tasks, in which we do not have adequate samples to train a deep CNN directly or numerous support sets for standard few-shot learning methods. Four QFSS datasets of SAR named CMU, Badminton, miniNTU-xsub, and miniNTU-xview, are established based on the widely used public datasets to validate the performance of the DMD feature. A group of experiments is conducted to analyze intrinsic properties of DMD, whereas another group focuses on its auxiliary functions. Experimental results show that the DMD feature can improve the performance of most typical CNN features in QFSS SAR tasks.

Keywords Skeleton-based action recognition · Dynamic mode decomposition · Quasi-few-shot setting · Translational and rotational invariance

1 Introduction

Action recognition (AR) demonstrates broad application prospects in intelligent security monitoring, human-machine interaction, virtual reality, and kinematic analysis (Zhu et al. 2020). With the development of deep learning (DL), AR methods based on deep convolutional neural networks (CNNs) have shown great superiority over traditional visual technologies. Those methods can be divided into three types: two-stream network (TSN), 3D CNN, and skeleton-based action recognition (SAR) methods. TSN and 3D CNN deal with a video clip in an end-to-end manner and utilize context information when actions are closely related to the context. Whereas, SAR methods operate in a decoupled manner and often consist of two stages. The first stage is human pose estimation, which detects skeleton trajectories (STs) of one or more humans from a video clip. The second stage is to classify the action category of the STs. The decomposition of human pose estimation from action classification can utilize the powerful generalization ability of well-trained pose estimation frameworks (Cao et al. 2017; Open-MMLab 2019) to eliminate the disturbance of background when the training set suffers from insufficient diversity.

Given that the collection of samples are expensive and time-consuming, some few-shot (Guo et al. 2018), one-shot (Memmesheimer et al. 2020) or zero-shot (Jasani and Mazagonwalla 2019) learning-based AR methods have been proposed to deal with sample shortage based on numerous support sets in the past two years. However, in many tasks whose goals are to detect illegal behaviors, the samples in the training set are more than the few-shot setting but not adequate to train a deep CNN directly or to generate support sets. That is the quasi-few-shot setting (QFSS). The

✉ Shuai Dong
dongshuai@zsc.edu.cn

[1] University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan 528402, China

[2] School of Computer, Guangdong University of Technology, Guangzhou 510006, China

challenge lies in seeking a priori knowledge to help the deep CNN to learn the feature better. The attention mechanism (Liu et al. 2020) and part-aware (Li et al. 2017a) convolutional operation are two useful manners to guide the training process.

In this paper, we proposed a new empirical feature for SAR based on dynamic mode decomposition (DMD). DMD is a popular realization of Koopman (Takeishi et al. 2017) and has been widely used in nonlinear dynamic analysis. By modeling the human action as a nonlinear dynamic system that determines the evolution of the ST, the system matrix or its eigenvalues can be treated as an empirical feature. The DMD feature has multiple advantages. First, DMD has a clear physical meaning. Although some information would be lost during the linearization process, DMD contains important time-frequency domain information that can recover the action appropriately when the initial state is given. Second, DMD has the property of translational and rotational invariance, that is, the DMD feature is constant when the position and pose of the camera changes. The DMD feature is also effective on 2D skeletons in a fixed scene. At last, the DMD feature can be concatenated with CNNs features to improve their accuracy.

The currently widely used CNN is optimized as a black box and extracts time domain features that are not interpretable. Whereas, the DMD feature, which is inspired by the control theory, is an empirical and interpretable feature in the frequency domain and has a fixed computational process without training owing to its clear physical meaning. Those differences allow the DMD feature to play an auxiliary role for CNN features in QFSS tasks.

The remainder of this paper is organized as follows. Section 2 reviews recent developments on AR. Section 3 proposed a new DMD-based SAR framework and proves the translational and rotational invariance of DMD. Section 4 presents and analyzes experimental results. Finally, Sect. 5 concludes the study.

2 Related work

The progress of video AR before the DL era is slow because of the inability of traditional visual technologies to perform high semantic-level tasks. A complete pipeline of traditional methods comprises feature extraction, combination, and classification. One typical method is the dense trajectory (DT) algorithm based on optical flow (Wang et al. 2013). The motion trail of the video is captured by optical flow first, then features including trajectory shape, histograms of oriented optical flow, gradient, and motion boundary are extracted. These features are encoded and used to train a support vector machine (SVM) classifier. Wang et al. also proposed the improved DT (IDT) algorithm (Wang and Schmid 2013) in the same year. Compared with DT, IDT utilized the improved optical flow graph, feature regularization, and encoding method to increased accuracy from 84.54 to 91.2% on the UCF50 dataset and from 46.6 to 57.2% on the HMD51 dataset.

Since DL flourished in 2015, many DL-based AR methods have been proposed (Kong and Fu 2018) and offered a wide range of possible applications in safety management (Zhu et al. 2020), violence detection (Sumon et al. 2019), and ambient assisted living (Singh et al. 2017). According to the architecture of the network, these methods can be divided into three categories, namely, TSN (Lin et al. 2020), 3D CNN (Tran et al. 2017; Diba et al. 2017), and SAR (Yan et al. 2018). In some works, long-short temporal memory (LSTM) networks (Singh et al. 2017) are also used to model the evolution process of STs, but their performance is inferior to TSN and 3D CNN because of the difficulty of training.

TSN mainly uses a two-stream architecture to extract semantic information from RGB frames and time domain information from optical flow, and combines features to make collaborative predictions. This technical route was first proposed by Simonyan (2014) and improved by other researchers from several aspects. Feichtenhofer et al. introduced 3D pooling (Feichtenhofer et al. 2016) and multiscale time (Feichtenhofer et al. 2018) into TSN. Wang et al. (2016) proposed temporal segment networks to address long-time videos and Zhou et al. (2018) put forward a temporal relation network to learn the dependency relationship between frames. Overall, TSN is the DL version of IDT that appropriately balances the computational burden and accuracy requirement.

Unlike TSN establishes the connection between frames with optical flow, 3D CNN executes the convolution operation in the time dimension to achieve the same goal (Tran et al. 2015). To reduce the computational burden and improve the performance of 3D CNN, many equivalent operations have been proposed. ResNet-(2+1)D architectures, which uses 2D convolution on each RGB image and 3*1*1 convolution on the temporal dimension, were proposed by Tran et al. (2015, 2017) and Qiu et al. (2017) individually. Diba et al. (2017) proposed a temporal 3D CNN to explore long-term information comprehensively, together with the temporal transition layer to replace the pooling layer. They initialized the 3D CNN with a pre-trained 2D CNN, which is also an enlightening approach. Lin et al. (2019) proposed a novel method suitable for 2D CNN models that remarkably reduces the computation and performs the cross concatenation of channels between frames to allow information sharing.

SAR consists of two steps. The first step is human pose estimation, which can be classified into top-down and bottom-up strategies. Top-down strategies use an object

detection framework to detect humans and locates skeleton joint points based on the detected boxes, whereas bottom-up strategies detect all possible joint points and cluster them to different humans. Many studies have been proposed and carried on open source frameworks OpenPose (Wei et al. 2016; Simon et al. 2017; Cao et al. 2017) or mmpose (Open-MMLab 2019).

Once the ST has been obtained by the human pose estimation module, the most intuitive method of SAR is to stack the ST into a one-channel image and input it into a one-channel 2D CNN, which is named as temporal convolution network (TCN) (Kim and Reiter 2017; Memmesheimer et al. 2020). Another direct way is to use recurrent neural networks (RNNs) to represent the temporal relation (Wang and Wang 2017; Liu et al. 2017; Singh et al. 2017). An indirect manner is to project the ST into three orthometric views and stack them as a three-channel image (Hou et al. 2018) that is suitable for a general multi-channel 2D CNN.

To enhance the performance of SAR, a priori knowledge about body parts are introduced into the network in the form of an undirected graph (Yan et al. 2018; Shi et al. 2019; Holzinger et al. 2021) or a fixed concatenation (Li et al. 2017a; Zhang et al. 2017). Similar to the performance of the graphical neural networks (Holzinger et al. 2021) in other applications, those part-aware methods provide a supervised attention mechanism substantially. At the same time, the unsupervised attention mechanism has also been exploited by some researchers. Si et al. (2019) proposed an attention-enhanced graph convolutional LSTM which achieves state-of-the-art on several public datasets; Li et al. (2019) combined an adaptive attention module with a two-stream RNN architecture. Furthermore, Zhao et al. (2019) combined a graphical neural network (GCN) with LSTM into a Bayesian framework, and Peng et al. (2020) proposed a Neural Architecture Search (NAS) framework to design a part-aware GCN automatically.

Although a complex architecture can achieve better performance when the dataset is large enough, many applications fail to satisfy this requirement. Referring to flourishing few-shot learning methods (including the one-shot and zero-shot methods) in other visual tasks, a small group of researchers starts to seek one-shot learning methods for SAR (Memmesheimer et al. 2020). A new dataset for few-shot learning of SAR is established based on the NTU dataset (Li et al. 2017b), which contains adequate support sets. Many few-shot learning methods would be extended to SAR in the following two years. Moreover, QFSS, which is closer to the requirements of real applications, deserves additional attention.

3 Method

The human body is a complex dynamic system, with the brain as the controller, action target and external environment as the inputs, and human joints as the actuators. The sequential skeleton points, that is the ST, are the observed states of the system. When finishing different actions, the system would evolve under the navigation of different controllers and output different STs. Thus, if we can recover the close system from a given ST with DMD, the action type would be recognized according to the modes extracted by DMD.

Inspired by this motivation, the DMD-based SAR framework is proposed in this section. DMD theory is introduced first; the translational and rotational invariance of the DMD feature is proven then; finally, the DMD-based action recognition framework is proposed.

3.1 Dynamic mode decomposition

Given a discrete system $\zeta_{k+1} = f(\zeta_k)$, where $\zeta_k \in \mathbb{R}^n$ is the latent state, K is the Koopman operator (Takeishi et al. 2017). K is an infinite linear operator defined as $\mathrm{K}(g(\zeta)) = g(f(\zeta))$ for $\forall g : \mathrm{M} \to \mathbb{R}(or\ \mathbb{C})$, where M is the state space of ζ, $\mathbb{R}$ (or $\mathbb{C}$) is the real (or image) set, $f(\cdot)$ is the dynamic function, and $g(\cdot)$ is the observation function.

It is assumed that K demonstrates discrete spectrums, which can be written in the form of infinite eigenvalues $\{\lambda_1, \lambda_2, \lambda_3, \cdots\}$ and eigenfunctions $\{\phi_1, \phi_2, \phi_3, \cdots\}$ with the relation $\mathrm{K}\phi_i = \lambda_i \phi_i$. The observation function based on eigenfunctions is $g(\zeta) = \sum_i \phi_i(\zeta) c_i$, i.e., $g(\zeta_k) = \sum_i \lambda_i^k \phi_i(\zeta_0) c_i$. The Koopman operator approximates a lower-dimensional nonlinear system to an infinite-dimensional linear system with sequential $K+1$ samples by seeking a state transition matrix $\boldsymbol{A} \in \mathbb{R}^{K \times K}$ that satisfies

$$\underbrace{\left[g(\zeta_2), g(\zeta_3), \cdots, g(\zeta_{K+1})\right]}_{H_2 \in \mathbb{R}^{n \times K}} \approx A \underbrace{\left[g(\zeta_1), g(\zeta_2), \cdots, g(\zeta_K)\right]}_{H_1 \in \mathbb{R}^{n \times K}}.$$

DMD is the most widely used method to calculate A.

Performing a singular value decomposition on $\boldsymbol{H}_1$, we have the following:

$$\boldsymbol{H}_1 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{1}$$

where $\boldsymbol{U} \in \mathbb{R}^{n \times n}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times K}$, and $\boldsymbol{V} \in \mathrm{R}^{K \times K}$. The diagonal elements of Σ are the singular values sorted descendingly, and all off-diagonal elements are 0. We can then obtain the similar matrix of $\boldsymbol{A}$ as follows:

$$\tilde{\boldsymbol{A}} = \boldsymbol{U}^T \boldsymbol{A} \boldsymbol{U} = \boldsymbol{U}^T \boldsymbol{H}_2 \boldsymbol{V} \boldsymbol{\Sigma}^{-1}. \tag{2}$$

$\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}$ have the same eigenvalues.

Considering that the response of a dynamic system is mainly determined by low-frequency parts, only the first r

eigenvalues are typically reserved to describe the feature of the system in practice, where $r \ll K$. Let $\boldsymbol{U}_r \in \mathbb{R}^{r\times n}$, $\boldsymbol{V}_r \in \mathbb{R}^{K\times r}$, and $\boldsymbol{\Sigma}_r \in \mathbb{R}^{r\times r}$ be the left-top submatrices of $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ with truncated eigenvalues, respectively. We can then obtain the approximated state transition matrix

$$\tilde{\boldsymbol{A}}_r = \boldsymbol{U}_r^T \boldsymbol{H}_2 \boldsymbol{V}_r \boldsymbol{\Sigma}_r^{-1}, \tag{3}$$

and its eigenvalues $\tilde{\lambda}_i, i = 1, 2, \cdots, r$.

The state matrix $\tilde{\boldsymbol{A}}_r$ determines the dynamic response of the system, including stability, response speed, and overshot. $\tilde{\lambda}_i$ is the pole of the approximate linear closed-loop system and determines the stability of the system. Thus, both $\tilde{\boldsymbol{A}}_r$ and $\tilde{\lambda}_i$ can serve as an empirical feature for SAR. The feature dimension is r^2 for the flattened $\tilde{\boldsymbol{A}}_r$ and $2r$ for the stacked $\tilde{\lambda}_i$ (r real parts and r image parts). $\tilde{\lambda}_i$ is shorter, whereas $\tilde{A}_r$ contains more information. The experimental results in the following section show that the performance distinction between them is unclear.

3.2 Translational and rotational invariance of DMD

For an action sample, when the sensor moves or rotates, the feature for SAR should be consistent. With a simple normalization method, DMD can satisfy this requirement theoretically, that is, translational and rotational invariance.

Two STs of one same action captured by different cameras are denoted as follow:

$$\boldsymbol{G}^1 = \left[g^1(\zeta_1), g^1(\zeta_2), \cdots, g^1(\zeta_{K+1})\right] = \left[\boldsymbol{\eta}_1^1, \boldsymbol{\eta}_2^1, \cdots, \boldsymbol{\eta}_{K+1}^1\right]$$

and

$$\boldsymbol{G}^2 = \left[g^2(\zeta_1), g^2(\zeta_2), \cdots, g^2(\zeta_{K+1})\right] = \left[\boldsymbol{\eta}_1^2, \boldsymbol{\eta}_2^2, \cdots, \boldsymbol{\eta}_{K+1}^2\right].$$

$\boldsymbol{G}^1$ and $\boldsymbol{G}^2$ are captured by cameras with fixed coordinates $O_1x_1y_1z_1$ and $O_2x_2y_2z_2$. The s^{th} skeleton joint at step j captured by camera i is denoted as $p_j^i = \left[x_{s,j}^i, y_{s,j}^i, z_{s,j}^i\right]$. Then the spatial coordinate of all S skeleton points can be stacked as $\boldsymbol{\eta}_j^i = g^i(\boldsymbol{\zeta}_j) = \left[\boldsymbol{p}_{1,j}^i, \boldsymbol{p}_{2,j}^i, \cdots, \boldsymbol{p}_{S,j}^i\right]^T \in \mathbb{R}^{3S\times 1}$.

The transfer matrix from $O_1x_1y_1z_1$ to $O_2x_2y_2z_2$ is denoted as

$$T_{1to2} = \begin{bmatrix} \boldsymbol{r}_{1to2} & \boldsymbol{l}_{1to2} \\ \boldsymbol{0}_{1\times 3} & \boldsymbol{1} \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{4}$$

where $\boldsymbol{r}_{1to2}$ is the rotation matrix and $\boldsymbol{l}_{1to2}$ is the translation vector. $\boldsymbol{r}_{1to2}$ and $\boldsymbol{l}_{1to2}$ satisfy

$$p_j^2 = \boldsymbol{r}_{1to2} \cdot p_j^2 + \boldsymbol{l}_{1to2}, \tag{5}$$

for $k = 1, 2, \cdots, K+1$.

The translational and rotational invariance of DMD means that $\tilde{\boldsymbol{A}}_r^1 = \tilde{\boldsymbol{A}}_r^2$. This property is proven as follow:

Proof Based on (5), we have

$$\begin{aligned} \boldsymbol{\eta}_j^2 &= \boldsymbol{R}\boldsymbol{\eta}_j^1 + \boldsymbol{L} \\ \boldsymbol{G}^2 &= \boldsymbol{R}\boldsymbol{G}^1 + \boldsymbol{L}, \end{aligned}$$

where,

$$\begin{aligned} \boldsymbol{R} &= \text{diag}\{\underbrace{\boldsymbol{r}_{1to2}, \boldsymbol{r}_{1to2}, \cdots, \boldsymbol{r}_{1to2}}_{N \text{ blocks}}\} \\ \boldsymbol{L} &= \underbrace{\left[\boldsymbol{l}_{1to2}^T, \boldsymbol{l}_{1to2}^T, \cdots, \boldsymbol{l}_{1to2}^T\right]}_{N \text{ blocks}}^T. \end{aligned}$$

Normalize $\overline{\boldsymbol{\eta}}^i$ with

$$\overline{\boldsymbol{\eta}}^i = \boldsymbol{\eta}^i - \boldsymbol{L}_0^i, \tag{6}$$

where $\boldsymbol{L}_0^i = \underbrace{\left[\boldsymbol{p}_{1,1}^i, \boldsymbol{p}_{1,1}^i, \cdots, \boldsymbol{p}_{1,1}^i\right]}_{N \text{ blocks}}^T$ with $\boldsymbol{R}\boldsymbol{L}_0^1 = \boldsymbol{L}_0^2 - \boldsymbol{L}$, we can then obtain the following relation:

$$\overline{\boldsymbol{\eta}}^2 = \boldsymbol{R}\overline{\boldsymbol{\eta}}^1 + \boldsymbol{R}\boldsymbol{L}_0^1 - \boldsymbol{L}_0^2 + \boldsymbol{L} = \boldsymbol{R}\overline{\boldsymbol{\eta}}^1.$$

Thus,

$$\boldsymbol{R}\overline{\boldsymbol{G}}^1 = \boldsymbol{R}\left[\overline{\boldsymbol{\eta}}_1^1, \overline{\boldsymbol{\eta}}_2^1, \cdots, \overline{\boldsymbol{\eta}}_{K+1}^1\right] = \left[\overline{\boldsymbol{\eta}}_1^2, \overline{\boldsymbol{\eta}}_2^2, \cdots, \overline{\boldsymbol{\eta}}_{K+1}^2\right] = \overline{\boldsymbol{G}}^2 \tag{7}$$

Let $\boldsymbol{H}_1^i = \left[\overline{\boldsymbol{\eta}}_1^i, \overline{\boldsymbol{\eta}}_2^i, \cdots, \overline{\boldsymbol{\eta}}_K^i\right]$ and $\boldsymbol{H}_2^i = \left[\overline{\boldsymbol{\eta}}_2^i, \overline{\boldsymbol{\eta}}_3^i, \cdots, \overline{\boldsymbol{\eta}}_{K+1}^1\right]$, and there is

$$\boldsymbol{H}_1^2 = \boldsymbol{R}\boldsymbol{H}_1^1, \quad \boldsymbol{H}_2^2 = \boldsymbol{R}\boldsymbol{H}_1^2 \tag{8}$$

The system matrices can be obtained with Eqs. (1 and 2) as follows:

$$\begin{aligned} \boldsymbol{H}_1^i &= \boldsymbol{U}^i \boldsymbol{\Sigma}^i \left(\boldsymbol{V}^i\right)^T \\ \tilde{\boldsymbol{A}}^i &= \left(\boldsymbol{U}^i\right)^T \boldsymbol{H}_2^i \boldsymbol{V}^i \left(\boldsymbol{\Sigma}^i\right)^{-1}. \end{aligned} \tag{9}$$

Then, we have

$$\begin{aligned} \boldsymbol{H}_1^1 &= \boldsymbol{U}^1 \boldsymbol{\Sigma}^1 \left(\boldsymbol{V}^1\right)^T \\ \boldsymbol{H}_2^2 &= \boldsymbol{R}\boldsymbol{H}_2^1 = \left(\boldsymbol{R}\boldsymbol{U}^1\right) \boldsymbol{\Sigma}^1 \left(\boldsymbol{V}^1\right)^T, \end{aligned} \tag{10}$$

and

$$\begin{aligned} \boldsymbol{A}^1 &= \boldsymbol{H}_2^1 \left(\boldsymbol{H}_1^1\right)^{-1} \\ \boldsymbol{A}^2 &= \boldsymbol{H}_2^2 \left(\boldsymbol{H}_1^2\right)^{-1} = \boldsymbol{R}\boldsymbol{H}_2^1 \left(\boldsymbol{R}\boldsymbol{H}_1^1\right)^{-1} = \boldsymbol{R}\boldsymbol{H}_2^1 \left(\boldsymbol{H}_1^1\right)^{-1} \boldsymbol{R}^{-1}. \end{aligned} \tag{11}$$

Their similar matrices are as follows:

$$\begin{aligned} \tilde{\boldsymbol{A}}^1 &= \left(\boldsymbol{U}^1\right)^T \boldsymbol{A}^1 \boldsymbol{U}^1 = \left(\boldsymbol{U}^1\right)^T \boldsymbol{H}_1^2 \left(\boldsymbol{H}_1^1\right)^{-1} \boldsymbol{U}^1 \\ \tilde{\boldsymbol{A}}^2 &= \left(\boldsymbol{R}\boldsymbol{U}^1\right)^T \boldsymbol{R}\boldsymbol{H}_2^1 \left(\boldsymbol{H}_1^1\right)^{-1} \boldsymbol{R}^{-1} \left(\boldsymbol{R}\boldsymbol{U}^1\right). \end{aligned} \tag{12}$$

As the rotation matrix is orthogonal and satisfies $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$, we can obtain the following:

$$\begin{aligned}\tilde{\boldsymbol{A}}^2 &= \left(\boldsymbol{U}^1\right)^T \left(\boldsymbol{R}^{-1}\boldsymbol{R}\right)\boldsymbol{H}_2^1\left(\boldsymbol{H}_1^1\right)^{-1}\left(\boldsymbol{R}^{-1}\boldsymbol{R}\right)\boldsymbol{U}^1 \\ &= \left(\boldsymbol{U}^1\right)^T \boldsymbol{H}_2^1\left(\boldsymbol{H}_1^1\right)^{-1}\boldsymbol{U}^1 \\ &= \tilde{\boldsymbol{A}}^1.\end{aligned}$$

Thus, there exists $\tilde{\boldsymbol{A}}_r^2 = \tilde{\boldsymbol{A}}_r^1$. □

From the proof above, DMD can guarantee rotational invariance inherently, and the normalization method in Eq. (6) can guarantee translational invariance. Thus, the normalization is a necessary preprocessing step for the DMD feature. Some other normalization methods can also guarantee translational invariance, but they have some disadvantages. For instance, $\overline{\boldsymbol{\eta}}^i = (\boldsymbol{\eta}^i - \boldsymbol{\eta}^0)/(\boldsymbol{\eta}^K - \boldsymbol{\eta}^0)$ or $\overline{\boldsymbol{\eta}}^i = (\boldsymbol{\eta}^K - \boldsymbol{\eta}^i)/(\boldsymbol{\eta}^K - \boldsymbol{\eta}^0)$, can normalize the trajectory into [0, 1] and satisfy the translational and rotational invariance. However, when $\boldsymbol{\eta}^K = \boldsymbol{\eta}^0$, they are not applicable.

3.3 DMD feature for SAR with QFSS

Deep CNNs can extract more information than DMD because of their large amount of parameters. Deep CNNs are the standard answers for SAR if the training set is adequate. However, in many QFSS SAR tasks which do not have adequate training samples, training a deep CNN is impossible. Facing this problem, we design a framework to improve the performance of CNN features on QFSS SAR tasks with the empirical DMD feature.

Denote a skeleton trajectory that has been normalized according to formula (6) in the form of a matrix

$$\boldsymbol{G} = \left[\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \cdots, \boldsymbol{\eta}_{K+1}\right],$$

where $\boldsymbol{\eta}_j = \left[\boldsymbol{p}_{1,j}, \boldsymbol{p}_{2,j}, \cdots, \boldsymbol{p}_{S,j}\right]^T \in \mathbb{R}^{3S\times 1}$ is stacked skeleton points with $p_j = \left[x_{s,j}, y_{s,j}, z_{s,j}\right]$ at step j. Then, we have

$$\begin{aligned}\boldsymbol{H}_1 &= \left[\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \cdots, \boldsymbol{\eta}_K\right] \\ \boldsymbol{H}_2 &= \left[\boldsymbol{\eta}_2, \boldsymbol{\eta}_3, \cdots, \boldsymbol{\eta}_{K+1}\right].\end{aligned}$$

By substituting $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ into Eqs. (1, 2, and 3), we can obtain the DMD feature v_{DMD} of $\boldsymbol{G}$ as follows:

$$v_{DMD} = DMD(\boldsymbol{G})$$

Input G into a CNN, the output is

$$v_{CNN} = CNN(\boldsymbol{G})$$

Then, a DMD-based SAR framework can be established, as depicted in Fig. 1, with the following five components:

(1) A human pose estimation module, for instance, OpenPose or mmpose, that can obtain skeleton trajectories from video clips;
(2) Normalization of the ST to obtain $\boldsymbol{G}$ according to formula (6);
(3) CNN feature extractor to obtain v_{CNN};
(4) DMD feature extractor to obtain v_{CNN};
(5) Final classifier to predict the action category.

As a trajectory can be recovered from its modes and eigenvectors approximately (Takeishi et al. 2017), DMD servers as an encoder in the framework. The physical meaning of the DMD feature is clear, compact, and informative. Although the order truncation operation and linearization may make some information lost, it is useful when the training set is not adequate.

The rank of DMD for a RAS task is often less than 10, and the length of v_{DMD} is less than 100. When using v_{DMD} together with v_{CNN}, the increased computation is negligible.

Considering the perspective transformation in the imaging acquisition of RGB videos, the DMD feature of a 2D skeleton trajectory cannot guarantee translational and rotational invariance. However, in some applications where the

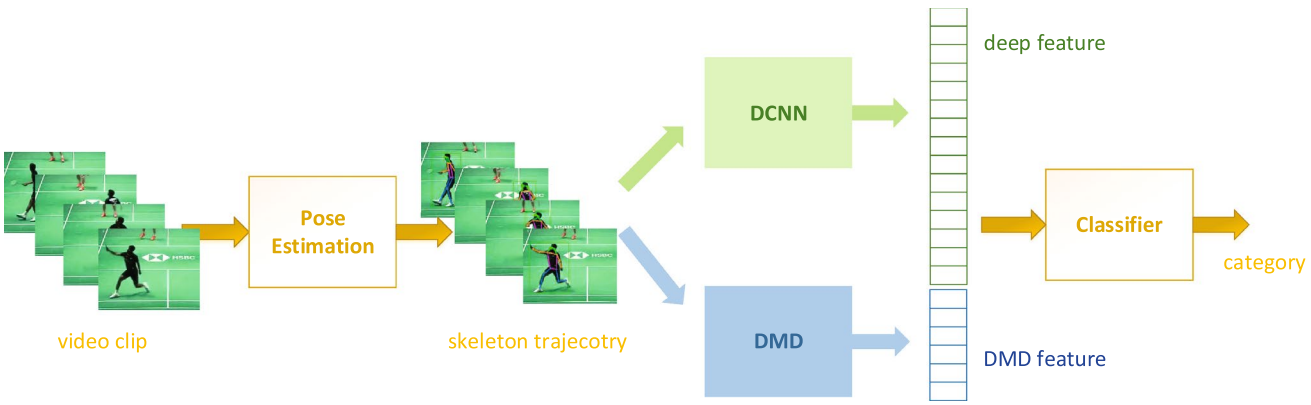

Fig. 1 Framework of SAR based on DMD

position and pose of the camera are fixed, the distortion of skeletons can be treated as one part of the action itself, and thus, the DMD feature is also suitable.

4 Experiments and analysis

To analyze the performance of the DMD feature comprehensively, two groups of experiments are conducted based on three datasets. In one group, the DMD feature is used alone to analyze its intrinsic properties. The matrix feature and eigenvalue feature of DMD is compared with basic LSTM on the CMU and Badminton datasets. In the other group, we focus on the auxiliary performance of the DMD feature. Five CNN-based methods, namely, ST-GCN (Yan et al. 2018), TCN (Kim and Reiter 2017), part-aware LSTM (PLSTM) (Shahroudy et al. 2016), ResNet18 (He et al. 2016) and basic LSTM (Graves 2012), have been chosen for comparison. DGNN (Shi et al. 2019), which used to be state-of-the-art on the NTU and Kinematic datasets, failed to converge on the miniNTU dataset. Thus, we did not present its results. ResNet18 is a special realization of TCN with its backbone as a one-channel residual network, which has a much deeper architecture than other methods. Basic LSTM contains three layers and each layer contains 100 neurons. In all methods, we have adjusted the feature length to 256 and the output layer to a linear fully-connected layer with 256 inputs and 4 (CMU and Badminton) or 40 (miniNTU) outputs. All these methods are trained with randomly initialized parameters.

4.1 Datasets

The DMD feature is an empirical feature with limited length, and it does not have a strong expressive ability like the CNN feature. Thus, the motivation of this work is to explore the applicable scenes of DMD, rather than seeking a state-of-the-art accuracy. We have chosen three datasets with very different properties to analyze DMD fully.

(1) CMU dataset. CMU dataset (CMU 2013) is a classic dataset for motion capture, in which 29 skeleton points are measured by wearable devices. Thus, its precision is much higher than other datasets. We divided a subset from the CMU dataset in this group, which includes dancing, jumping, running, and walking actions. Figure 2 shows some samples of the CUM dataset. A total of 119 samples are used for training and test, whose distribution is listed in Table 1. We removed 4 unnecessary skeleton joints to make it share the NTU's data loader. CMU is easier than other datasets.

(2) Badminton dataset. The Badminton dataset is a self-established dataset to illustrate the applicability of the DMD feature for 2D ST in a fixed scene. This dataset also contains four categories of actions, namely, backhand striking, forehand striking, backhand lifting, and forehand lifting. The 2D skeleton trajectories are obtained with the human pose estimation framework mmpose (Open-MMLab 2019) from some video clips. Some failed frames are replenished with linear interpolation. An action of badminton contains three stages, namely, move toward the shuttlecock, hit, and return to the defensive position. In addition, some athletes hold the racket in their right hand while the rest in their left hands. This makes the action more indistinguishable. Figure 3 show some samples. The training set contains 30 trajectories for each type, and the test set contains 12, 10, 10, and 13 for each type respectively. We only considered the athlete in the field below by limiting the detection region of their feet. In this dataset, the skeleton contains 17 joints. The distortion of the 2D skeleton by perspective transformation, the consistency of the athlete's movements, and the confusion of main hands between athletes, make it much more difficult than the CMU dataset.

(3) miniNTU dataset. NTU (Li et al. 2017b) is a widely used large-scale dataset for SAR. This dataset contains 60 categories of actions, and 20 of them involve multiple

Table 1 Number of samples in the CMU dataset

	Dancing	Jumping	Running	Walking
Training set	14	24	21	28
Testing set	5	7	9	11

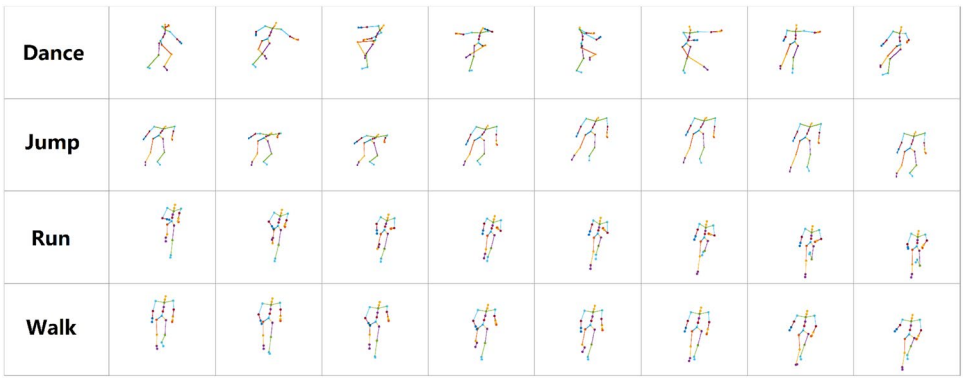

Fig. 2 Four types of actions in the CMU dataset

Fig. 3 Four types of actions in the badminton dataset

humans. The skeletons are captured with three RGBD sensors located at different poses. In this work, we considered those 40 types of actions with only one human and chose 30 training and 10 test samples for each type of action, so that it satisfies QFSS. Same as the standard NTU dataset, we also established the cross-subject and cross-view subsets. The former means that the humans in the training set and test set are different; the latter means that the trajectories in the training set and test set are captured by different cameras. It is much more difficult than the CMU and Badminton.

Another widely used dataset is the Kinetics (Kay et al. 2017) dataset, which is even more difficult than NTU. Its skeleton trajectories cannot satisfy the requirement of translational and rotational invariance, and thus, we have not tested the Kinetics dataset.

4.2 SAR based on DMD feature and ovoSVM

In this group, the intrinsic properties of DMD were explored. Because the miniNTU dataset is too difficult to fully exhibit the DMD feature's properties, we only conducted experiments on the CMU and Badminton datasets. We have considered the contrast experiment from several aspects.

First, as both DMD and LSTM can directly utilize temporal information, we have designed a DMD+ovoSVM framework as the realization of the DMD feature and chosen a basic shallow LSTM for comparison from several aspects in this group. Shallow CNN performs much poorer than DMD+ovoSVM and LSTM because it cannot extract temporal information. Thus, we did not compare DMD+ovoSVM with the shallow CNN in this group. The DMD+ovoSVM framework is a simple realization of Fig. 1, in which the classifier is an ovoSVM with radial basis function (RBF) kernels, and the DMD feature is input into the ovoSVM without concatenation with any CNN feature. Considering that the length of many skeleton trajectories in the Badminton dataset is less than 40, we limited the DMD rank to be smaller than 7, and the number of RBF kernels in ovoSVM ranges from 0.1 to 300. Second, both two types of DMD features mentioned above, that is, the flattened matrix and the stacked eigenvalues, have been considered. Finally, to explore computation reducing method, *a half truncation* and *four joints* tricks were tested. The former truncates trajectories from the middle inspired by the fact that all movements in Badminton contain recovering processes. The latter reduces the skeleton to 4 joints including two wrists and two ankles, due to the limb's movement range is relatively large than the body's.

Table 2 shows all optional hyperparameter configurations of DMD+ovoSVM. A uniformly distributed noise is added to the trajectories to augment the training and test set, and 10 duplicates of each trajectory are generated. The noise is defined as $x \to (1 + 0.05 * \Delta) \cdot x$, where $\Delta \sim U(-1, 1)$. The LSTM has three linear fully-connected layers with 100 neurons to extract the feature, and another linear fully-connected layer is used to predict the action categories. The input length of LSTM is truncated or padded with 0–200 for CMU and 40 for Badminton respectively. The appropriate hyperparameters in Table 2, including truncation, augmentation, and four joints, are also used on LSTM.

Each configuration is repeated 50 times and the best results of all configurations are listed in Table 4. Binary classification results on the striking and lifting subset of Badminton are also presented for reference. It can be found that: (1) LSTM achieved the highest accuracy on CUM with good stability, whereas, LSTM achieved the lowest accuracy on Badminton with the worst stability. (2) The matrix feature is preferred on the CMU dataset, whereas the eigenvalue

Table 2 Optional hyperparameters for DMD+ovoSVM

Hyperparameters	Values	
	CMU	Badminton
Feature type	Flattened matrix, stacked eigenvalues	
Rank of DMD	2–14	2–7
Coefficient of RBF	0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 300	
Half truncation	false	True, false
Random augmentation	10	10
Shuffled eigenvalues	True, false	
Four joints	False	True, false

feature is preferred on the badminton dataset. (3) Backhand and forehand lift actions are more difficult to classify than strike actions because of their high similarity.

Figure 4 shows the corresponding distribution of the results in Table 3. In the figures, the flattened matrix and the stacked eigenvalues are denoted as Amat and Mu, respectively. LSTM performs better than DMD+ovoSVM on the CMU dataset, but poorer on the Badminton dataset. The result of Amat on the CMU dataset is like a barbell, that is, it suffers from a large standard deviation. As DMD extracts the modes of an approximate linear system, the DMD feature has no relation with the input and would drop out some spatial information that is useful for the classification of CMU. A latent temporal condition of Badminton is that lift actions must occur in the frontcourt and strike actions must occur in the backcourt. If this temporal condition can be utilized, the classification results on complete Badminton should be close to the subsets. However, both LSTM and DMD failed to utilize this condition.

To analyze the performance of DMD more comprehensively, we compared the results of different hyperparameters. We computed the accuracy of all executions in the rank test. Figure 5 shows the distribution of accuracy versus the rank (r) of DMD. The optimal results are achieved for lift and strike actions when $r = 3$ because the difficulty to obtain the feature boundary increases when the length of the feature increases. The results of $r = 2$ for strike action are poorer than that of $r > 2$. This indicated that minimum low-frequency modes may be insufficient in describing the strike action. A tradeoff exists between the rank and length of the DMD feature, and how to determine the rank for different tasks is an important problem that deserves in-deep investigation.

Figures 6, 7 and 8 show the comparison of the half trajectory, four points, and shuffle eigenvalue tricks, respectively. The half trajectory and four points tricks do not lead to loss of accuracy. Thus, they can be used to reduce the computation significantly in some tasks. Shuffling operation on eigenvalues composes a negative effect on the high accuracy region but makes the distribution converge to the middle region.

DMD+ovoSVM can achieve the best performance near the shallow LSTM. The training speed of DMD+ovoSVM is higher and more stable. The solving process of DMD+ovoSVM only takes approximately 0.02–0.4 ms when running on the CPU Intel@i9-9900K. LSTM takes approximately 2 min for 50 epochs training when running on the CPU Intel@i9-9900K. The time decrease to 0.2–4 s when running on one piece of GPU NVIDIA@RTX2080ti. Since DMD involves singular value decomposition and matrix inversion, a GPU cannot accelerate the computation of DMD. The inability to utilize the GPU is a disadvantage of DMD.

4.3 SAR based on DMD feature and CNN feature

In this group, we considered the auxiliary role of the DMD feature for some popular deep CNNs, including ST-GCN, TCN, ResNet18, basic LSTM, and PLSTM. According to the framework in Fig. 1, the DMD feature in the form flattened matrix is concatenated with the CNN feature that is extracted by one of those deep CNNs and input into a linear fully-connected layer for classification. No trick in Table 2 has been used in this group. Table 4 shows all configuration for this group of experiments.

Tables 5 and 6 present the results on the CMU and Badminton datasets, and the miniNTU datasets, respectively. We collected the mean, maximum, and standard deviation of accuracy from 20 executions of ST-GCN and ST-GCN+DMD on the miniNTU dataset and 50 executions of others. The results of DMD+ovoSVM are also presented

Table 3 Comparison of optimal accuracy

Task configuration	DMD+ovoSVM matrix feature			DMD+ovoSVM eigenvalue feature			LSTM		
	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
CMU	0.8500	0.6084	0.2392	0.7906	0.6716	0.0355	0.8750	0.7262	0.0767
Badminton-strike	0.8636	0.7609	0.0390	0.8591	0.7948	0.0617	0.9545	0.6345	0.1437
Badminton-lift	0.8043	0.6694	0.0619	0.8522	0.6523	0.0546	0.8696	0.6522	0.1003
Badminton	0.5622	0.4088	0.0467	0.5800	0.4711	0.0272	0.4889	0.3400	0.0637

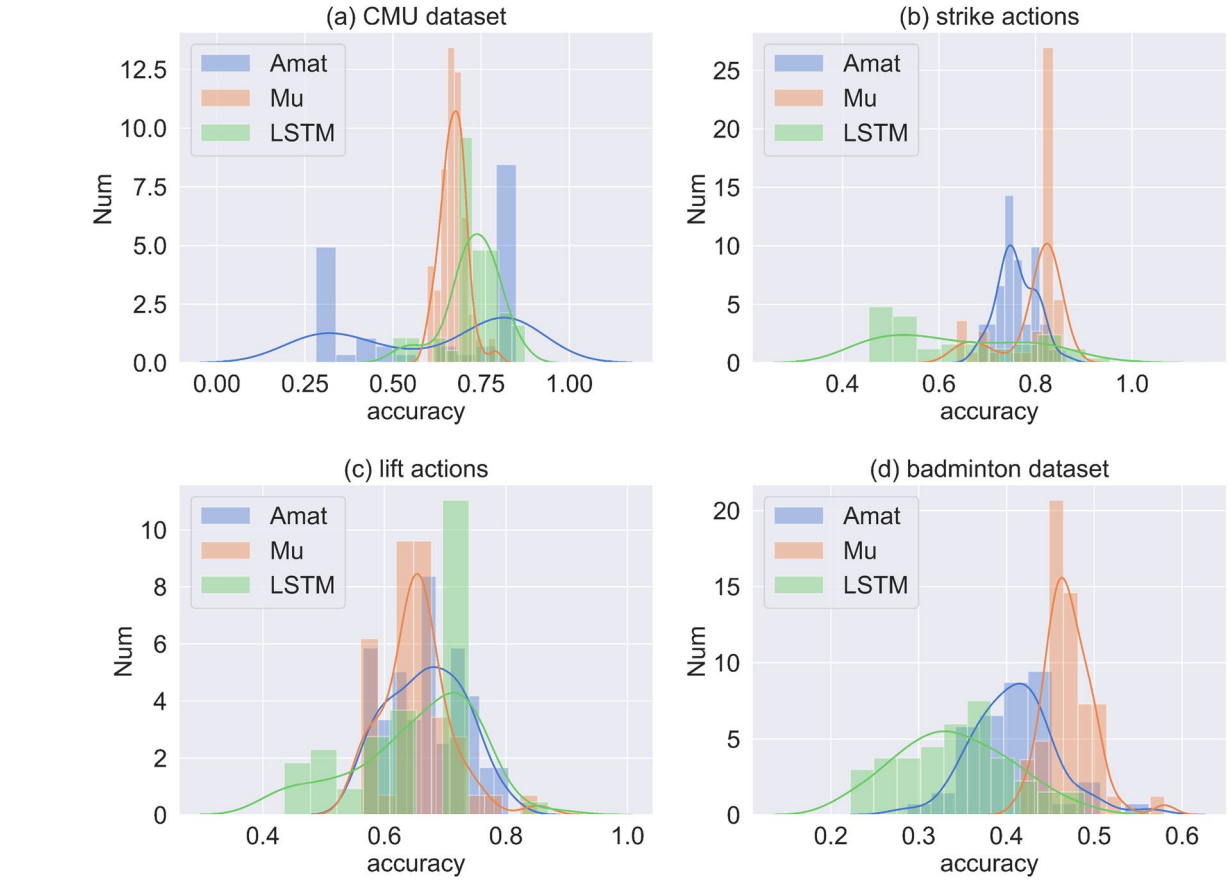

Fig. 4 Accuracy distribution of the optimal hyperparameter configuration

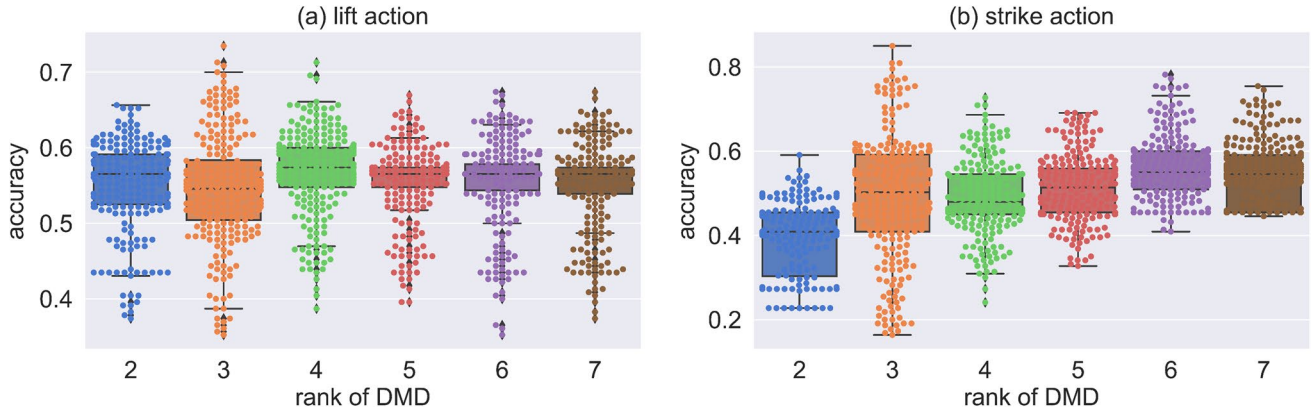

Fig. 5 Accuracy vs. rank of DMD

for reference. The result of TCN, LSTM, and PLSTM on miniNTU are ($mean = 0.025, max = 0.025, std = 0$), which means that the networks have not converged and always output a fixed prediction. Because the STs obtained by the pose estimation module often suffer from instability, the robustness of the DMD feature should be analyzed. Thus, we augmented the training and test sets 10 times with 5% uniformly distributed random noise according to $x \rightarrow (1 + 0.05 * \Delta)x$, where Δ $U(-1, 1)$. The results are listed in Tables 7 and 8.

From the results, it can be found that:

(1) The DMD feature can improve the performance of most methods, particularly, help TCN become convergent on miniNTU-xsub and PLSTM convergent on miniNTU-xview. ResNet18 can represent the frequency domain information

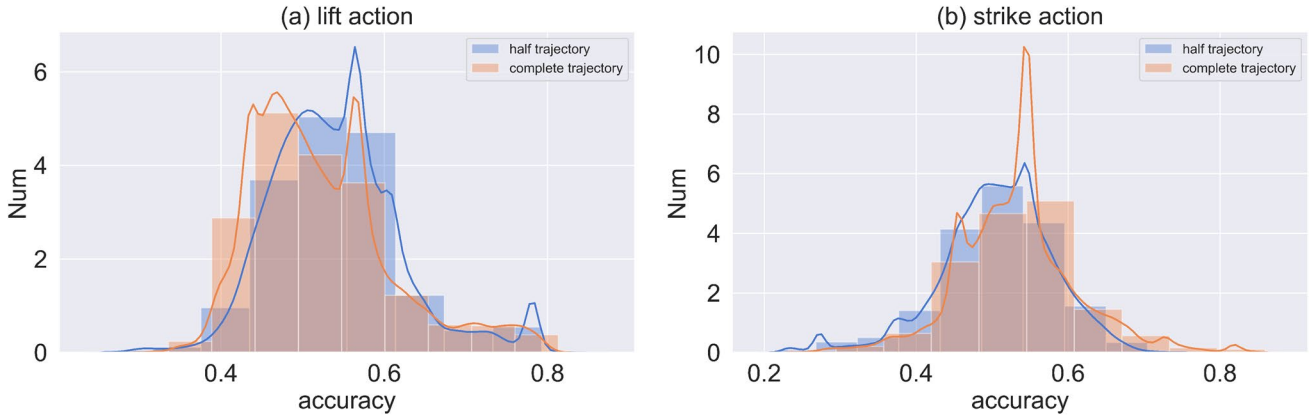

Fig. 6 Accuracy of DMD+ovsSVM with eigenvalue feature: half trajectory vs. complete trajectories

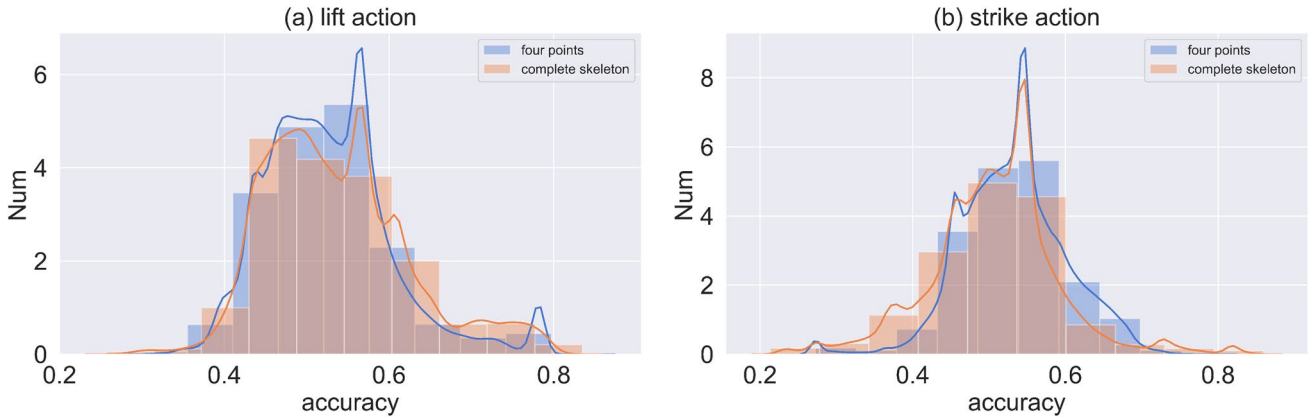

Fig. 7 Accuracy of DMD+ovsSVM with eigenvalue feature: four points vs. complete skeleton

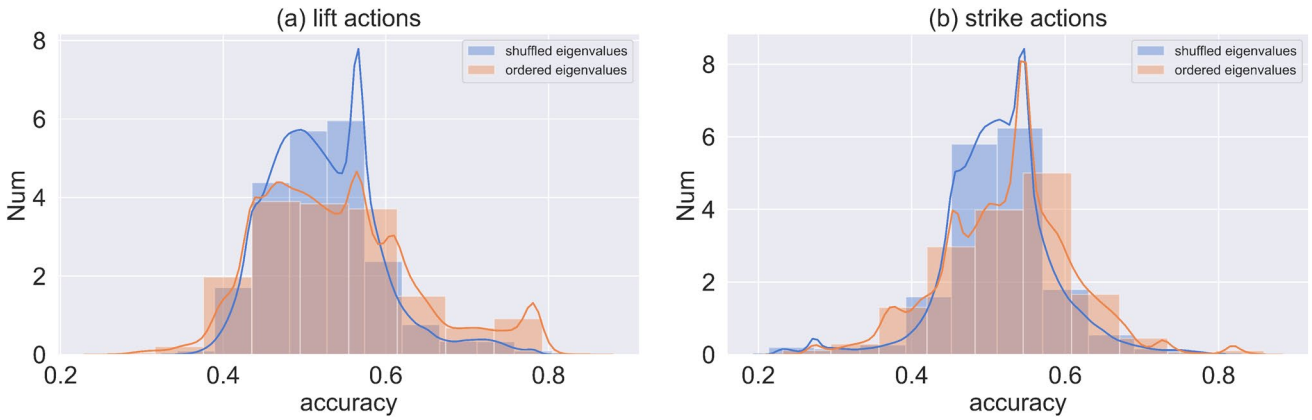

Fig. 8 Accuracy of DMD+ovoSVM with eigenvalue feature: shuffle vs. ordered

owing to its deep architecture and multiple convolution layers. Thus, the DMD feature cannot provide supplementary information for ResNet18. The DMD feature would lose some spatial information and is not as complete as a deep CNN feature.

(2) A recurrent architecture can also extract temporal information, but shallow layers would limit its feature expression ability. Thus, LSTM and PLMST perform better than TCN but much poorer than ST-GCN and ResNet18.

Table 4 Configuration for experiments of SAR based on DMD feature

Hyperparameters	Values
Length of v_{CNN}	256
Lenght of v_{DMD}	25
Max epochs	40
Base learning rate	0.1
Learning rate decaying strategy	Exponential
Optimizator	'SGD'
Weight decay	0.0001
Loss function	Basic cross entropy loss

(3) The performance of ResNet18 exceeds ST-GCN dramatically on all three datasets. However, ST-GTN is better than ResNet18 on the standard NTU dataset. In our test, the top 1 accuracies of ResNet18 are 79% and 87% on standard NTU-xsub and NTU-xview dataset, whereas ST-GCN achieves 81.3% and 89.1%. The results of ST-GCN+DMD are very close to ST-GCN, which means that DMD provides no information for ST-GCN. With the predefined relation of the human skeleton, ST-GCN has a stronger ability to extract spatial and temporal information than ResNet18. When the training samples are adequate, the spatial relation between joints brings more benefit than the frequency domain information. However, when the samples are not adequate in QFSS tasks, the predefined relation failed to perform fully.

(4) Although noise would injure the performance of all methods evidently, the auxiliary function of DMD still lasts when a 5% noise exists.

A deeper GCN, which combines the advantages of both deep architecture and part-aware knowledge, would own a better performance. However, it requires more samples and stronger computing power. When a deep architecture is unable to deploy, for instance, running on some embedded neural computing devices or lack of training samples, the DMD feature can be used to assist some simpler CNN feature to achieve higher accuracy.

Table 5 Comparison of accuracy on the CMU and Badminton datasets

Method	CMU			Badminton		
	Mean	Max	Std	Mean	Max	Std
ST-GCN	0.6725	0.8750	0.1211	0.8564	0.9200	0.0389
ST-GCN+DMD	0.6919	0.8125	0.0884	0.8564	0.9400	0.0370
TCN	0.4069	0.7188	0.1024	0.2720	0.4000	0.0465
TCN+DMD	0.5356	0.7500	0.0937	0.3188	0.4000	0.3004
ResNet18	0.8549	0.9062	0.0221	0.8864	0.9400	0.0228
ResNet18+DMD	0.8479	0.8750	0.0231	0.8840	0.9200	0.1960
LSTM	0.6771	0.7188	0.0230	0.2512	0.3200	0.0331
LSTM+DMD	0.6901	0.7500	0.0164	0.4455	0.4800	0.0189
PLSTM	0.6800	0.7188	0.0148	0.2536	0.3200	0.0397
PLSTM+DMD	0.6856	0.7188	0.0116	0.3244	0.4000	0.0297
DMD+ovoSVM	–	0.6938	–	-	0.4821	–

Table 6 Comparison of accuracy on the miniNTU dataset

Method	MiniNTU-xsub			MiniNTU-xview		
	Mean	Max	Std	Mean	Max	Std
ST-GCN	0.4637	0.5275	0.0286	0.4987	0.5550	0.0295
ST-GCN+DMD	0.4829	0.5325	0.0218	0.5110	0.5600	0.0295
TCN	0.0250	0.0250	0.0000	0.0250	0.0250	0.0000
TCN+DMD	0.4591	0.4925	0.0205	0.0898	0.1025	0.0047
ResNet18	0.4665	0.4975	0.0184	0.5423	0.5925	0.0198
ResNet18+DMD	0.4643	0.5275	0.0193	0.5552	0.6000	0.0195
LSTM	0.0250	0.0250	0.0000	0.0250	0.0250	0.0000
LSTM+DMD	0.0902	0.1025	0.0044	0.0942	0.1050	0.0052
PLSTM	0.0250	0.0250	0.0000	0.0250	0.0250	0.0000
PLSTM+DMD	0.0883	0.0950	0.0038	0.3194	0.3300	0.0057
DMD+ovoSVM	–	0.1508	–	–	0.1813	–

Table 7 Comparison of accuracy on the CMU and Badminton datasets with 5% uniformly distributed random noise

Method	CMU			Badminton		
	Mean	Max	Std	Mean	Max	Std
ST-GCN	0.6631	0.8623	0.1251	0.8364	0.9027	0.0428
ST-GCN+DMD	0.6926	0.8420	0.0984	0.8453	0.9230	0.0380
TCN	0.3823	0.7068	0.1083	0.2612	0.4000	0.0551
TCN+DMD	0.5311	0.7287	0.1012	0.2933	0.4000	0.3001
ResNet18	0.8492	0.8925	0.0412	0.8642	0.9400	0.0328
ResNet18+DMD	0.8401	0.8723	0.0431	0.8645	0.9200	0.2061
LSTM	0.6555	0.7008	0.0420	0.2376	0.3028	0.0354
LSTM+DMD	0.6789	0.7311	0.0243	0.4364	0.4800	0.0205
PLSTM	0.6632	0.6928	0.0148	0.2388	0.3107	0.0404
PLSTM+DMD	0.6768	0.7006	0.0116	0.3014	0.4012	0.0322
DMD+ovoSVM	–	0.5938	–	–	0.3855	–

Table 8 Comparison of accuracy on the miniNTU dataset with 5% uniformly distributed random noise

Method	MiniNTU-xsub			MiniNTU-xview		
	Mean	Max	Std	Mean	Max	Std
ST-GCN	0.4625	0.5217	0.0273	0.4969	0.5549	0.0333
ST-GCN+DMD	0.4825	0.5356	0.0214	0.5104	0.5543	0.0204
TCN	0.0250	0.0250	0.0000	0.0250	0.0250	0.0000
TCN+DMD	0.4578	0.4950	0.0201	0.0900	0.1023	0.0032
ResNet18	0.4632	0.4921	0.0184	0.5407	0.5899	0.0194
ResNet18+DMD	0.4640	0.5253	0.0201	0.5536	0.6000	0.0208
LSTM	0.0250	0.0250	0.0000	0.0250	0.0250	0.0000
LSTM+DMD	0.0925	0.1088	0.0041	0.0844	0.0915	0.0033
PLSTM	0.0250	0.0250	0.0000	0.0250	0.0250	0.0000
PLSTM+DMD	0.0866	0.0982	0.0056	0.3190	0.3387	0.0061
DMD+ovoSVM	–	0.0912	–	–	0.1031	–

5 Conclusion

The DMD feature for RAS is studied in this work. This feature has a clear physical meaning in the frequency domain and can guarantee translational and rotational invariance with an appropriate normalization. The DMD feature can achieve a performance close to a shallow LSTM when it is used solely in SAR tasks. A DMD-based SAR framework is proposed, in which the DMD feature is concatenated with a CNN feature. The DMD feature can improve the CNN features' accuracy evidently in QFSS SAR tasks with a small computational cost, even when a 5% noise exists. Particularly, DMD can help TCN become convergent on the miniNTU-xsub dataset and PLSTM convergent on the miniNTU-xview dataset. Because we cannot utilize a GPU to accelerate the calculation of DMD, the DMD-based SAR framework cannot be combined in an end-to-end framework. Thus, one of our works in the future is to find a realization of DMD on GPU, for instance, training a CNN to extract the modes. Furthermore, as the DMD feature only represents the modes of an approximated linear system and would lose some spatial information, another problem that deserves further research is to explore some empirical spatial features that can eliminate the information loss problem of DMD.

Acknowledgements This work is supported by National Natural Science Foundation of China (62002053), Natural Science Foundation of Guangdong Province (2021A1515011866), Guangdong Basic and Applied Basic Research Projects (2019A1515111082, 2020A1515110504), Fund for High-Level Talents Afforded by University of Electronic Science and Technology of China, Zhongshan Institute (417YKQ12, 419YKQN15), Social Welfare Major Project of Zhongshan (2019B2010, 2019B2011, 420S36), Achievement Cultivation Project of Zhongshan Industrial Technology Research Institute (419N26), the Science and Technology Foundation of Guangdong Province (2021A0101180005), and Young Innovative Talents Project of Education Department of Guangdong Province (2018KQNCX337,2019KQNCX186).

Declarations

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

Cao Z, Sheikh T, Shih-En S, Yaser W (2017) Realtime multi-person 2D pose estimation using part affinity fields. In: IEEE conference on computer vision and pattern recognition, pp 7291–7299

CMU (2013) CMU graphics lab motion capture database

Diba A, Fayyaz M, Sharma V, Karami AH, Arzani MM, Yousefzadeh R, Gool LV (2017) Temporal 3D ConvNets: new architecture and transfer learning for video classification. arXiv: 171108200 pp. 1–10

Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. IEEE computer society conference on computer vision and pattern recognition. pp. 1933–1941. https://doi.org/10.1109/CVPR.2016.213

Feichtenhofer C, Fan H, Malik J, He K (2018) Slowfast networks for video recognition. In: IEEE/CVF international conference on computer vision, pp. 6201–6210

Graves A (2012) Long short-term memory. Springer, Berlin, pp 37–45

Guo M, Chou E, Huang DA, Song S, Yeung S, Fei-Fei L (2018) Neural graph matching networks for few-shot 3D action recognition. European conference on computer vision. Munich, Germany, pp. 673–689

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. IEEE conference on computer vision and pattern recognition. Las Vegas, USA, pp 771–778

Holzinger A, Malle B, Saranti A, Pfeifer B (2021) Towards multimodal causability with Graph Neural Networks enabling information fusion for explainable AI. Inf Fusion 71:28–37. https://doi.org/10.1016/j.inffus.2021.01.008

Hou Y, Li Z, Wang P, Li W (2018) Skeleton optical spectra-based action recognition using convolutional neural networks. IEEE Trans Circuits Syst Video Technol 28(3):807–811. https://doi.org/10.1109/TCSVT.2016.2628339

Jasani B, Mazagonwalla A (2019) Skeleton based zero shot action recognition in joint pose-language semantic space. arXiv: 191111344 pp. 1–8, arXiv: 1911.11344v1

Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P, Suleyman M, Zisserman A (2017) The Kinetics human action video dataset. arXiv: 17050 6950. pp. 1–22

Kim TS, Reiter A (2017) Interpretable 3D human action analysis with temporal convolutional networks. IEEE conference on computer vision and pattern recognition workshops, pp. 1623–1631

Kong Y, Fu Y (2018) Human action recognition and prediction: a survey. arXiv: 180611230 13(9):1–19

Li B, He M, Cheng X, Chen Y, Dai Y (2017a) Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN. In: IEEE international conference on multimedia and expo workshops, pp. 601–604

Li C, Zhong Q, Xie D, Pu S (2017b) Skeleton-based action recognition with convolutional neural networks. IEEE international conference on multimedia and expo workshops. China, Hong Kong, pp. 597–600

Li L, Zheng W, Zhang Z, Huang Y, Wang L (2019) Relational network for skeleton-based action recognition. IEEE international conference on multimedia and expo, pp .826–831. arXiv: 1805.02556v1

Lin J, Gan C, Han S (2019) TSM: temporal shift module for efficient video understanding. In: IEEE/CVF international conference on computer vision (ICCV), pp. 7082–7092. https://doi.org/10.1109/ICCV.2019.00718

Lin J, Gan C, Wang K, Han S (2020) TSM: Temporal shift module for efficient and scalable video understanding on edge devices. IEEE transactions on pattern analysis and machine intelligence, p. 1, https://doi.org/10.1109/TPAMI.2020.3029799

Liu J, Wang G, Hu P, Duan Ly, Kot AC (2017) Global context-aware attention LSTM networks for 3D action recognition. IEEE conference on computer vision and pattern recognition. pp, 1647–1656

Liu R, Shen J, Wang H, Chen C, Cheung SC, Asari V (2020) Attention mechanism exploits temporal contexts: real-time 3D human pose reconstruction. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp. 5063–5072. https://doi.org/10.1109/CVPR42600.2020.00511

Memmesheimer R, Theisen N, Paulus D (2020) Signal level deep metric learning for multimodal one-shot action recognition. arXiv: 201213823v1. pp. 1–7

Open-MMLab (2019) mmpose. https://githubcom/open-mmlab/mmpose

Peng W, Hong X, Chen H, Zhao G (2020) Learning graph convolutional network for skeleton-based human action recognition by neural searching. In: AAAI conference on artificial intelligence, New York, USA, pp. 2669–2676. https://doi.org/10.1609/aaai.v34i03.5652

Qiu Z, Yao T, Mei T (2017) Learning spatio-temporal representation with pseudo-3D residual networks. IEEE international conference on computer vision. pp. 5534–5542. https://doi.org/10.1109/ICCV.2017.590

Shahroudy A, Liu J, Ng TT, Wang G (2016) NTU RGB+D: a large scale dataset for 3D human activity analysis. IEEE conference on computer vision and pattern recognition. Las Vegas, USA, pp. 1010–1019

Shi L, Zhangng Y, Cheng J, Lu H (2019) Skeleton-based action recognition with directed graph neural networks. IEEE conference on computer vision and pattern recognition. Long Beach, USA, pp. 7912–7921

Si C, Chen W, Wang W, Wang L, Tan T (2019) An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. IEEE/CVF conference on computer vision and pattern recognition. Los Angeles CA, United States, pp. 1227–1236

Simon T, Joo H, Matthews I, Sheikh Y (2017) Hand keypoint detection in single images using multiview bootstrapping. In: IEEE conference on computer vision and pattern recognition, pp. 1145–1153

Simonyan K (2014) Two-stream convolutional networks for action recognition in videos. 27th International conference on neural

information processing systems, pp. 1–11, https://arxiv.org/pdf/1406.2199.pdf, arXiv: 1406.2199v2

Singh D, Merdivan E, Psychoula I, Kropf J, Hanke S, Geist M, Holzinger A (2017) Human activity recognition using recurrent neural networks. In: Holzinger A, Kieseberg P, Tjoa AM, Weippl E (eds) Machine learning and knowledge extraction. Springer, Cham, pp 267–274

Sumon SA, Shahria MT, Goni MR, Hasan N, Almarufuzzaman AM, Rahman RM (2019) Violent crowd flow detection using deep learning. Springer, Berlin

Takeishi N, Kawahara Y, Yairi T (2017) Learning Koopman invariant subspaces for dynamic mode decomposition. arXiv: 171004340, pp. 1–18

Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. IEEE international conference on computer vision, pp. 4489–4497. https://doi.org/10.1109/ICCV.2015.510

Tran D, Ray J, Shou Z, Chang SF, Paluri M (2017) Convnet architecture search for spatiotemporal feature learning. arXiv: 17080 5038, pp. 1–10

Wang H, Schmid C (2013) Action recognition with improved trajectories. IEEE international conference on computer vision, pp. 3551–3558, https://doi.org/10.1109/ICCV.2013.441

Wang H, Wang L (2017) Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. IEEE conference on computer vision and pattern recognition, pp. 499–508

Wang H, Kläser A, Schmid C, Liu CL (2013) Dense trajectories and motion boundary descriptors for action recognition. Int J Comput Vis 103(1):60–79. https://doi.org/10.1007/s11263-012-0594-8

Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Gool LV (2016) Temporal segment networks: towards good practices for deep action recognition. In: European conference on computer vision, pp. 20–36

Wei SE, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. In: IEEE conference on computer vision and pattern recognition, pp. 4724–4732

Yan S, Xiong Y, Lin D (2018) Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI conference on artificial intelligence, New Orleans, USA, pp. 1–10, arXiv: 1801.07455v2

Zhang S, Liu X, Xiao J (2017) On geometric features for skeleton-based action recognition using multilayer LSTM networks. IEEE winter conference on applications of computer vision, pp. 148–157

Zhao R, Wang K, Su H, Ji Q (2019) Bayesian graph convolution LSTM for skeleton based action recognition. In: IEEE international conference on computer vision, Los Angeles CA, United States, pp. 6881–6891, https://doi.org/10.1109/ICCV.2019.00698

Zhou B, Andonian A, Oliva A, Torralba A (2018) Temporal relational reasoning in videos. In: European conference on computer vision, pp. 803–818

Zhu Y, Li X, Liu C, Zolfaghari M, Xiong Y, Wu C, Zhang Z, Tighe J, Manmatha R, Li M (2020) A comprehensive study of deep video action recognition. arXiv: 201206567v1, pp. 1–30