Stacked Denoising Tensor Auto-Encoder for Action Recognition With Spatiotemporal Corruptions

Chengcheng Jia, Ming Shao, *Member, IEEE*, Sheng Li, *Member, IEEE*, Handong Zhao, *Student Member, IEEE*, and Yun Fu, *Senior Member, IEEE*

***Abstract*— Spatially or temporally corrupted action videos are impractical for recognition via vision or learning models. It usually happens when streaming data are captured from unintended moving cameras, which bring occlusion or camera vibration and accordingly result in arbitrary loss of spatiotemporal information. In reality, it is intractable to deal with both spatial and temporal corruptions at the same time. In this paper, we propose a coupled stacked denoising tensor auto-encoder (CSDTAE) model, which approaches this corruption problem in a divide-and-conquer fashion by jointing both the spatial and temporal schemes together. In particular, each scheme is a SDTAE designed to handle either spatial or temporal corruption, respectively. SDTAE is composed of several blocks, each of which is a denoising tensor auto-encoder (DTAE). Therefore, CSDTAE is designed based on several DTAE building blocks to solve the spatiotemporal corruption problem simultaneously. In one DTAE, the video features are represented as a high-order tensor to preserve the spatiotemporal structure of data, where the temporal and spatial information are processed separately in different hidden layers via tensor unfolding. In summary, DTAE explores the spatial and temporal structure of the tensor representation, and SDTAE handles different corrupted ratios progressively to extract more discriminative features. CSDTAE couples the temporal and spatial corruptions of the same data through a thorough step-by-step procedure based on canonical correlation analysis, which integrates the two sub-problems into one problem. The key point is solving the spatiotemporal corruption in one model by considering them as noises in either spatial or temporal direction. Extensive experiments on three action data sets demonstrate the effectiveness of our model, especially when large volumes of corruption in the video.**

Manuscript received April 27, 2016; revised December 4, 2016 and June 22, 2017; accepted November 22, 2017. Date of publication December 8, 2017; date of current version January 18, 2018. This research was supported in part by the NSF IIS Award under Award 1651902, in part by the ONR Young Investigator Award under Award N00014-14-1-0484, and in part by the U.S. Army Research Office Award under Award W911NF-17-1-0367. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Feng Wu. *(Corresponding author: Chengcheng Jia.)*

C. Jia and H. Zhao are with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: jia.ch@husky.neu.edu; zhao.han@husky.neu.edu).

M. Shao is with the Department of Computer and Information Science, College of Engineering, University of Massachusetts Dartmouth, Dartmouth, MA 02747 USA (e-mail: mshao@umassd.edu).

S. Li is with the Adobe Research, San Jose, CA 95110 USA (e-mail: sheli@adobe.com).

Y. Fu is with the Department of Electrical and Computer Engineering, College of Engineering, and College of Computer and Information Science, Northeastern University, Boston, MA 02115 USA (e-mail: yunfu@ece.neu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2781299

***Index Terms*— Stacked tensor auto-encoder, action recognition, spatiotemporal corruption.**

I. Introduction

AS RECOGNIZED by the previous research [1], [2], temporal corrupted videos are not qualified for action recognition. Even worse, the corrupted videos cannot be regenerated under certain circumstances, especially in public surveillance, e.g., a small section of video may be lost or damaged due to visual occlusion or vibration of the camera, or the transmission issue of the network infrastructure. Besides, although promising results have been achieved on a few datasets collected under collaborations, the real-world applications are far from ideal. A major issue to be addressed is sequential disorder or missing frames caused by data corruption in a given video sequence. When analyzing a relatively short video, such consecutive temporal corruption may be catastrophic. There are a few pioneering works to make use of residual video parts to predict actions when temporal corruption occurs [3], [4]. Ryoo [3] gradually increased the length of video sequences, and subsequently generated different prediction tasks. Their work focuses on temporal corruption only occurred on the ending segment. However, in practice, a large portion of corruption may occur in a sequence, randomly at intervals, e.g. data may be damaged when being captured or saved, which is difficult to be re-generated in the case of large datasets. If we treat the unseen part from the prediction problems as the corruption, then it can be considered as a special case of random temporal corruption. In the previous works [3], [5], different occlusion ratios, locations of corruption, and occlusion of action scenarios have been considered. Cao *et al.* [5] cut a video into eight segments with small shifts along the elapsed time and meanwhile set different occlusion ratios, and each segment may be lost caused by an arbitrary temporal corruption. These methods mainly make use of the existing corrupted data, without trying to recover some missing information in a model. Additionally, occlusion or Gaussian noise in spatial space may be induced by occasional camera vibration, which is another problem accompanying spatial corruption we aim to address in this paper.

To clarify the problem, we first illustrate the spatial and temporal corruption in Figure 1. For the spatial corruption, we can see that key poses are occluded by random black squares. Compared to previous works [5], the spatial corruption may occur randomly in any segment. For the temporal corruption,

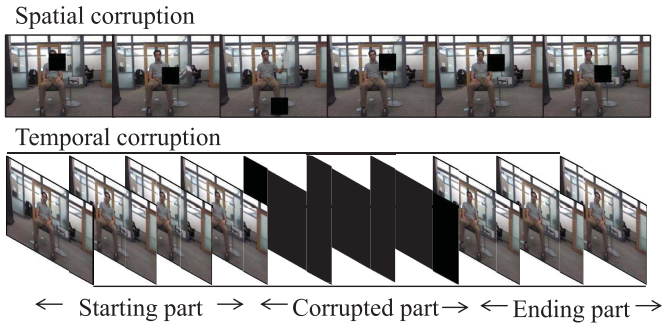


Fig. 1. Illustration of spatial and temporal corruption from an action video. It can be seen that temporal corruption can cause the loss of some key frames which crucial for recognition.

we suppose a large portion of sequential frames missing in arbitrary locations, which may result in lack of substantial knowledge. To the best of our knowledge, action recognition under this scenario has never been considered before. Second, we show temporal corruptions with unknown ratio problems in Figure 2, which shows the main problem solved by all of our solutions. We suppose three corrupted levels here, 20%, 40% and 60%, for concise illustration. For the first situation, we divide the whole video into five parts without overlapping, and each training procedure corresponds to each corrupted part. For the second situation, the adjacent missing parts have 20% overlap. While there 40% overlap between the adjacent missing parts in the third situation. In this paper, we formally name our problem as action recognition with *spatiotemporal corruption*. In brief, the challenges in action recognition solved in this paper can be described as:

- *Temporal corruption* with unknown ratios and locations
- *Spatial corruption* by object occlusion or Gaussian noise

Existing action recognition methods dealing with temporal corrupted videos rely on artificial preprocessing. Ryoo [3] calculated integral histograms of action segments for prediction. Cao *et al.* [5] used sparse coding of corrupted videos to calculate the likelihood of an action. Different from these handcraft features, deep learning methods are widely used recently for action feature extraction, e.g., Convolutional Neural Network (CNN) [6]. However, CNN does not perform well on corrupted videos with a large portion of lost frames due to the incomplete temporal information. As a robust representation learning model, Denosing Auto-Encoder (DAE) [7], [8] is appropriate to handle missing spatial data. Inspired by this, we use DAE to handle lost frames of a video in the temporal direction. In addition, the deep structure may further progressively mitigate the corruptions layer by layer. In this way, we could handle large portion of missing frames that can not be addressed by the existing models.

The main purpose of this paper is to solve challenges of both spatial and temporal corruptions simultaneously for human action recognition. Existing works only focus on monotonous problem for visual recognition, e.g., for temporal corruption of human action videos [3], [5], or for spatial corruption of videos or images [8]. In order to solve the simultaneous corruption problem in one uniform model, we consider temporal corruption as a denoising problem, and therefore we could

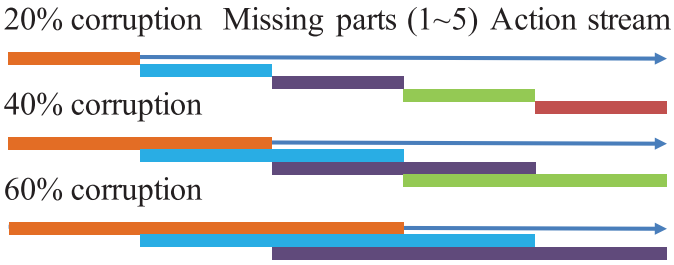


Fig. 2. Illustration of different corruption ratios and locations in an action stream. Take 20% temporal corruption for example, each part ($1 \sim 5$) is non-overlapped with others, and the missing part is lost randomly in our problem.

employ DAE for both temporal and spatial corruptions of the input videos. Furthermore, we could use two dedicated DAEs, i.e., one for temporal and the other for spatial corruption, to progressively handle large portion of corruptions, inspired by the divide-and-conquer theory. In addition, some robust action recognition methods and deep learning methods are introduced and compared, e.g., Ryoo [3], Cao *et al.* [5], Chen *et al.* [8], and C3D network [9]. 3D Convolutional Networks (C3D) is a deep learning method commonly used for action recognition recently, as it uses a 3D convolution filter to preserve the temporal information of input videos. We employ it in the experiment as a benchmark for the deep learning methods.

Data representation is crucial for designing a specific model for our problems. As we know, a direct impact of the spatiotemporal corruption is the incompleteness of the visual descriptors. Popular action descriptors, e.g., BoW [10] with 3D-SIFT [11], dense trajectory [12], HOG-4D [13], HOF [14], [15], STIP [16], trajectory [12], and tracklet [17], rely on both local spatial and temporal information. Absence of either ruins the features, notwithstanding the spatiotemporal corruption previously discussed here. Recently, the spatiotemporal factors in an action video have been modeled as a higher-order tensor [18], i.e., a multi-dimensional array, where spatial, temporal or multi-view properties are embedded in different dimensions (directions) of a tensor [6]. To tackle each factor, the tensor would need to be unfolded along the corresponding direction to become a matrix. The tensor representation for videos not only maintains the spatiotemporal structure, but also reduces the dimensionality for video descriptors [19], [20].

Our contributions are two-fold. Firstly, we propose a new problem regarding both spatial-temporal corruptions of human actions, and propose a solution to how to deal with them simultaneously. Secondly, to solve the arbitrary spatiotemporal corruption challenges, we design a joint stacked framework by combining the existing Auto-Encoder building blocks.

II. Problem Statement

A. Temporal Corruption

The temporal corruption of action videos is illustrated in Figure 1, as random parts of a video are missing. To address this challenge, we propose a Denoising Tensor Auto-Encoder (DTAE) model for *temporal corruption*. We consider temporal corruption as noise in the temporal dimension, and therefore use DAE [21] to mitigate this problem. In a DAE, the input is usually data with noise, and

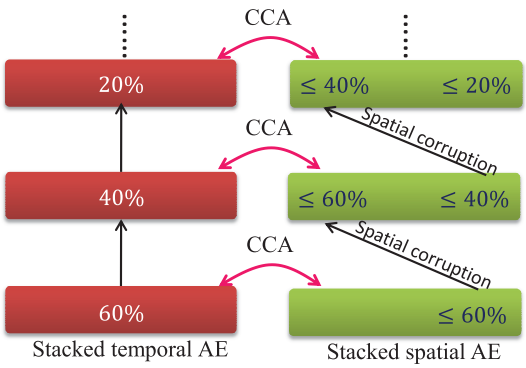


Fig. 3. Schematic of CSDTAE model, which is composed of two SDTAE models. The left SDTAE is used to deal with temporal corruption, while the right one is for spatial corruption.

the output is clean data. DAE is suitable for image denoising [22] or video denoising [23]. We convert an action video to a *third-order* tensor, and the corresponding model is called DTAE. Considering a set of tensors as the input, we aim to extract both the spatial and temporal features from the tensors. It is natural to formulate a deep structure for DTAE, where the first/second hidden layers are reserved for spatial features and the third hidden layer for temporal features. Therefore, in a deep DTAE model, the spatial and temporal corruption are handled separately within different hidden layers, and meanwhile the temporal correlation is explored in a closed form via tensor representation. A progressive strategy [24], [25] is used here to extract more discriminative features. We design a stacked scheme [26], [27] to mitigate the corruption impacts from one DTAE to another. For example, the stacked DTAE (SDTAE) is composed of three DTAEs. In the first DTAE, the input is training data with $[0\%, 60\%]$ (noted as 60%) temporal corruption, and the output belongs to $[0\%, 40\%]$ (noted as 40%) corruption. The intermediate feature from hidden layer is taken as input of the second DTAE, whose output belongs to $[0\%, 20\%]$ (noted as 20%) corruption. The intermediate of the second DTAE is used as input of the third DTAE, whose output is with 0% corruption. This stacked setting aims to mitigate the larger temporal corruptions progressively, in order to extract more discriminative features. The illustration of SDTAE is shown in the left side of Figure 3.

B. Spatial Corruption

Spatial corruption of videos with Gaussian noise is shown in Figure 4. This corruption is solved in the same way as the previous corruption types, i.e., the DTAE deals with both spatial and temporal corruption simultaneously. However, it is impractical to extract better features by aligning the arbitrary corrupted spatial and temporal information in two directions at the same time. Similar to the multi-view problem in [28], coupling two different corruptions from the same data and projecting them to a new subspace is a better approach to obtaining the common features. Inspired by this, we designed a divide-and-conquer scheme termed the Coupled SDTAE (CSDTAE) which handles the spatial and temporal corruptions separately, then joints two parts together to maximize their correlations using deep Canonical Correlation

Fig. 4. Spatial corruption with 20%, 40%, and 60% of Gaussian noise, respectively.

Analysis (CCA), shown in Figure 3. The left side is an SDTAE introduced to process temporal corruption. The right side is also an SDTAE model and is applied to deal with the spatial corruption. The right SDTAE contains three DTAEs. In each DTAE, the input and output are the same temporal corrupted data as in the left SDTAE while companying extra arbitrary spatial corruption. In this way, this SDTAE contains the same temporal information and only deals with spatial corruption progressively. Finally, the first/second/third DTAE of each SDTAE are coupled with deep CCA, which finds the maximum correlation of different corruptions for the same data. By this CSDTAE model, we deal with spatiotemporal corruption problems separately then integrate them in one model step-by-step, which transforms one intractable problem into two simple sub-problems.

III. Related Work

A. Action Recognition With Corruption

Given a 3D action video with f frames, we define two different types of video corruptions: 1) the starting part of several frames is kept, while the remaining part is missing, which is also known as action prediction [29], [30]; 2) the missing section could be located anywhere, such as at the beginning, middle or ending of the sequence, which is more flexible and challenging when compared to the first case [5]. In [29] and [30], dynamic BoW is extracted as the features, while in [5], SIFT + BoW from overlapped segments is used. When considering spatiotemporal corruption in this paper, none of the methods above can work well as they only take handcrafted features, which cannot automatically identify the representations from the corruption.

B. Tensor Representation

In recent work, action videos are converted to a *third-order* tensor, including three directions (modes) standing for different factors. The first and second modes indicate the spatial factor, and the third mode displays the temporal change. In [31] and [32], tensors are assumed to distribute in a manifold to find a common subspace by a tangent space bundle. In [33], a *third-order* tensor representation is used for multi-view gait recognition. In [34], a tensor based deep stacking network is proposed where each layer represents an unfolding tensor along a different direction. However, none of the works above considers the corruption in the action videos, especially the spatiotemporal corruption discussed in this paper.

C. Auto-Encoder (AE)

In [35] and [36], over-complete features are identified by a modified Independent Components Analysis (ICA) in a

TABLE I

ACCURACY OF THE DIFFERENT TEMPORAL CORRUPTIONS ON MSRDAILYACTIVITY3D AND MSRACTIONPAIRS DATASETS BY SVM CLASSIFIER. IN ALL THE SITUATIONS WITH DIFFERENT CORRUPTED RATIOS (GIVEN 20%, 40% AND 60% FOR INSTANCE) AND LOCATIONS (SHOWN IN FIGURE 2), WE FOCUS ON THE MOST CHALLENGING CASE TO EVALUATE OUR MODEL, E.G., PART1 IN 20% CASE

Datasets	0(%)	20 (%)					40 (%)				60 (%)		
Missing		Part1	Part2	Part3	Part4	Part5	Part1	Part2	Part3	Part4	Part1	Part2	Part3
Daily	30.62	**22.50**	23.75	25.00	27.50	27.50	23.75	**21.87**	23.75	25.62	**20.00**	23.75	21.87
Pair	85.00	**80.55**	88.88	83.88	85.55	81.11	80.00	83.88	82.77	**69.44**	79.44	82.77	**62.22**

sparse AE framework for action recognition. The work in [37] proposes to use a 3D spatiotemporal patch of an action as input and output of an AE. In [38], a split AE model is designed for reconstructing two-view data from a shared view. Recently, in [39], CCA is introduced into the deep structure to capture uncorrelated features of two subspaces as complementary knowledge. A follow-up work in [40] summarizes the existing deep CCA work with head-to-head comparisons, and proposes a new variant called deep canonically correlated Auto-Encoders.

D. Denoising Auto-Encoder (DAE)

In [41], a robust AE is identified by adding random noise to the input layer, and try to reconstruct clean data as the output. Later in [7], a two level denoising Auto-Encoder is designed as stacked DAE to extract better features. In order to speed up, a marginalized DAE is proposed in [8] by using a linear transformation to reconstruct the output data.

Different from traditional DAE, the input of our DTAE model is heavy spatiotemporal damaged data such as the 60% ratio of temporal corruption, and the output is less damaged data with the 40% corruption, for instance. DTAE learns different robust hidden layers with spatiotemporal information and their correlation. The extended SDTAE is designed to mitigate the temporal/spatial corruption in a progressive scheme. CSDTAE is proposed by coupling the two SDTAEs to make the same data with different corruptions more close in a new subspace by deep CCA in the output layer. CSDTAE is based on the divide-and-conquer idea to deal with different problems as sub-problems in one framework.

IV. PROPOSED METHOD

In this paper, we design a model for spatiotemporal corrupted action videos with unknown corruption ratios, locations and complex noises. We propose to use a DTAE to learn the robust mechanism for spatiotemporal corruption, which is detailed in Section IV-C. For unknown corruption ratios, a stacked DTAE (SDTAE) is proposed to avoid overfitting our model when facing a large damage ratio, which is detailed in Section IV-D. For unknown corruption locations and complex noises, we design a coupled SDTAE (CSDTAE) model to align heterogeneous data in each hidden layer, which is detailed in Section IV-E.

A. Tensor Representation for Action Videos

An *N-order* tensor is an N-directional array represented as $\mathcal{X}^{I_1 \dots \times_n I_n \dots \times_N I_N}$, where I_n is the dimension of n-th direction [42]. In this paper, we present a 3D action video as a *third-order* tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1, I_2, I_3 indicate the rows, columns, frames of the action video, respectively. Tucker decomposition is usually used for dimensionality reduction, which is defined as

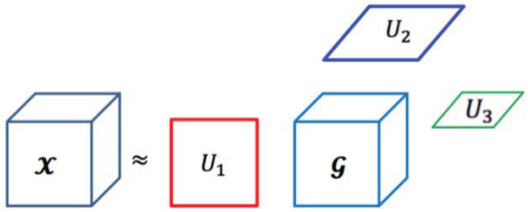


Fig. 5. Tucker decomposition of *third-order* tensor $\mathcal{X}$ by *mode-1,2,3* projection matrices U_1, U_2, U_3, and $\mathcal{G}$ is the core tensor.

$$\mathcal{X} = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3, \tag{1}$$

where U_n $(n = 1, 2, 3)$ is the *mode-n* projection matrix for dimensional reduction, and $\mathcal{G}$ is called the *core tensor*. To perform a transformation on each mode, we unfold $\mathcal{X}$ along different modes to get a matrix, i.e., we get *mode-n* unfolding matrix $X_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdot I_2 \dots I_{n-1} \cdot I_{n+1} \dots I_N)}$ by fixing the n-th direction while flatting others, here $n \leq N$, $N = 3$, and $\cdot$ is used for scalar product. The Tucker decomposition is illustrated in Figure 5.

B. Spatiotemporal Corruption Setting

We only consider limited cases in this work which are typical for spatiotemporal corruption, as there are infinite possible corruptions for any given sample. Specifically, we assume three corruption ratios: 20%, 40% and 60%. For locations, we select those with the least overlap given a fixed corruption ratio of (20%). In the following sections, we use a *fourth-order* tensor $\mathcal{X}_{20} \in \mathbb{R}^{r \times c \times f \times N}$ as a corrupted action video set in tensor representation with a 20% temporal missing levels, where r, c, f, N indicate the number of rows, columns, frames and samples respectively. Similarly, $\mathcal{X}_0, \mathcal{X}_{40}, \mathcal{X}_{60}$ stand for data with 0%, 40%, 60% corruptions. In addition, we use $\langle \mathcal{X}_{20}, \mathcal{X}_0 \rangle$ to represent the mixture of two datasets with 20% and 0% corruption ratios respectively. We aim to find the toughest cases for spatiotemporal corrupted action recognition under different ratios, and therefore we design several pre-selected experiments to pick up the worst cases on two datasets.

The results of corrupted action recognition with an SVM classifier on the MSRDailyActivity3D and MSRActionPairs datasets are shown in Table I. For the Daily dataset, we consider the lowest accuracies in different parts (shown in Figure 2) as the most tough challenges and select them for the next experiments. In the same way to Pair dataset, and we can see the 40% and 60% missing information are crucial in these samples according to the much lower accuracy.

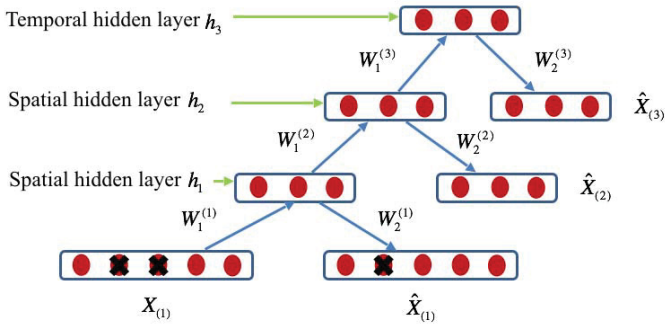


Fig. 6. DTAE model for spatial and temporal denoising. $X_{(1)}$ and $\hat{X}_{(1)}$ are made up of *mode-1* 40% and 20% corrupted data respectively, and $W_1^{(n)}$, $W_2^{(n)}$ are the corresponding *mode-n* weighted matrices in each layer. The first two spatial layers are related to *mode-1,2* transformations, and the third temporal layer indicates *mode-3* transformation.

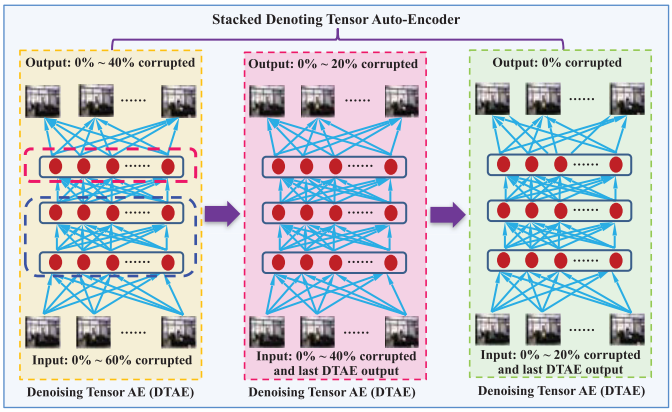


Fig. 7. The visualization of SDTAE model, which is composed of three DTAE models here. Each DTAE aims denoising with different input and output corrupted data. SDTAE is designed to reduce the spatial or temporal noises in a progressive manner.

C. Denoising Tensor AE (DTAE)

Here we introduce the foundation of a one-layer Auto-Encoder (AE). Given a dataset $X \in \mathbb{R}^{d\times N}$ with N samples where each one is represented as a column with d dimensions. Two weighted matrices $W \in \mathbb{R}^{d\times d'}$ and W^{T} are used as *encoder* and *decoder*, where d' is the transformed dimension. The aim of AE is to extract features by aligning its input and output. The objective function is described as follows:

$$\arg\min_{W,b_1,b_2} \|\sigma(W^{\mathrm{T}}\sigma(WX+b_1)+b_2)-X\|_F^2, \tag{2}$$

where σ is an activation function, b_1 and b_2 are biases. W is well-tuned in the training phase, and then this AE is used for new testing samples. In the corresponding AE model, random noise is added to the hidden layer, and then the reconstructed data is aligned to the input data to tune the weighted matrix.

In our tensor based model, a hybrid of n_1 corrupted data $\langle \mathcal{X}_{40}, \mathcal{X}_{20}\rangle$ is given as input, which is represented as $\mathcal{X} \in \mathbb{R}^{r\times c\times f\times n_1}$ for simplicity, where r, c, f indicate number of rows, columns, and frames of an action video, respectively. The output data is noted as $\hat{\mathcal{X}}$ with 20% corrupted ratio. Leveling up different input and output in a DTAE aims to reduce the larger corrupted ratio. Here i-th sample $\mathcal{Y}_i \in \mathbb{R}^{r\times c\times f}$ is a *third-order* tensor. By this representation, we only consider three hidden layers including two spatial layers and one temporal layer in a DTAE, in which we aim to find a multi-linear transformation in each layer for fast speed [8].

Figure 6 shows DTAE of *mode-n* encoder and decoder. $X_{(1)} \in \mathbb{R}^{r\times (cfn_1)}$ and $\hat{X}_{(1)} \in \mathbb{R}^{r\times (cfn_1)}$ are *mode-1* input and output, and each red node indicates one dimension of data. $W_1^{(1)} \in \mathbb{R}^{r\times r_1}$ and $W_2^{(1)} \in \mathbb{R}^{r_1\times r}$ are *mode-1* encoder and decoder weight matrices, and $W_1^{(2)} \in \mathbb{R}^{c\times c_1}$ and $W_2^{(2)} \in \mathbb{R}^{c_1\times c}$ are *mode-2* encoder and decoder weight matrices. Here both *mode-1, 2* matrices are used for spatial information processing. $W_1^{(3)} \in \mathbb{R}^{f\times f_1}$ and $W_2^{(3)} \in \mathbb{R}^{f_1\times f}$ are *mode-3* encoder and decoder weight matrices used for dealing with temporal information. $\hat{\mathcal{X}}_{(2)}$ is *mode-2* output, and $\hat{\mathcal{X}}_{(3)}$ is *mode-3* output.

In DTAE model, the input $\mathcal{X}$ and the output $\hat{\mathcal{X}}$ have different ratios of corruptions, and we need to align them via DTAE. Intuitively, we hope DTAE can reduce the corruption rate from 40% to 20%. In the meanwhile, we could learn all the *mode-n* weight matrices and find the tensor subspace for recognition. This model is inspired by marginal Stacked Denoising Autoencoder (mSDA) [8], in order to train a tensor model to reduce the temporal or spatial corrupted ratio. DTAE objective function is:

$$\arg\min_{\mathcal{W}} \|\mathcal{X}-\mathcal{W}\hat{\mathcal{X}}\|_F^2, \tag{3}$$

where $\mathcal{W} \in \mathbb{R}^{r\times c\times f}$ is weight tensor composed of *mode-n* matrix, presented as $\mathcal{W} = (W_1^{(1)}W_2^{(1)}) \otimes (W_1^{(2)}W_2^{(2)}) \otimes (W_1^{(3)}W_2^{(3)})$, $\otimes$ is Kronecker product of two matrices. The solution of *mode-n* weight matrix $W_1^{(n)}$ or $W_2^{(n)}$ $(n = 1, 2, 3)$ is calculated from the derivation of Eq. (3):

$$W_1^{(n)}W_2^{(n)} = [X_{(n)}\hat{X}_{(n)}^{\mathrm{T}}][\hat{X}_{(n)}\hat{X}_{(n)}^{\mathrm{T}}]^{-1}, \tag{4}$$

here we set $W_2^{(n)} = [W_1^{(n)}]^{\mathrm{T}}$, therefore a Gram matrix is defined as $G = W_1^{(n)}[W_1^{(n)}]^{\mathrm{T}}$, and we could get all the *mode-n* weight matrices by calculating the eigenvectors of G.

Time Complexity: We analyze and compare the time complexity of DTAE and mSDA [8]. Suppose $D \in \mathbb{R}^{d\times m}$ is a training set with m samples, each of which is a d-dimensional vector, we present this dataset as an $(N+1)$-*order* tensor $\mathcal{D} \in \mathbb{R}^{\sqrt[N]{d}\times\ldots\times\sqrt[N]{d}\times m}$. mSDA is proposed for denoising by a linear transformation, and it takes $O(d^2)$. The main step of DTAE is also a linear transformation, and in tensor framework its time complexity is $O(Nd^{\frac{2}{N}})$. We can see that our method spends less time especially when the dimension d is large.

In DTAE, we could deal with the spatiotemporal corruption in each layer, instead of mixing them together in traditional AEs regardless of the different affect of factors. However, DTAE can handle only one level of corruption (40% → 20% corruption). Furthermore, we generate an improved model for different levels, which is called Stacked DTAE model and is described in the following section.

D. Stacked DTAE (SDTAE)

We take the temporal SDTAE for example. SDTAE is composed of three DTAE models to extract discriminative features processively, which is shown in Figure 7. In each DTAE, the first two hidden layers of which indicate spatial

information and the third hidden layer indicates the temporal information. For the first DTAE, the input $\mathcal{X}^{(1)}$ and output are all made up of temporal corrupted data, but the input data with 60% corruption ratio is aligned to the 40% corrupted output data set. Similarly in the second DTAE, the input data $\mathcal{X}^{(2)}$ with 40% corruption ratio is aligned to 20% corrupted output data set. In the third DTAE, the input data $\mathcal{X}^{(3)}$ with 20% corruption ratio is aligned to 0% corrupted output dataset. This setting extracts discriminative features procedurally in the stacked scheme, and is generative to any corrupted data in the test phase.

We assume the weight tensor $\mathcal{W}$ is the same in three DTAEs, regarding to the temporal corrupted test data with unknown ratios. SDTAE objective function is:

$$\arg\min_{\mathcal{W}} \sum_{k=1}^{3} \|\mathcal{X}^{(k)} - \mathcal{W}\hat{\mathcal{X}}^{(k)}\|_F^2, \tag{5}$$

where k DTAEs in one SDTAE, and $\mathcal{X}^{(k)}$ and $\hat{\mathcal{X}}^{(k)}$ are the input and output of k-th DTAE. The solution of $W_1^{(n)}$ or $W_2^{(n)}$ $(n = 1, 2, 3)$ is calculated from the derivation of Eq (5):

$$W_1^{(n)} W_2^{(n)} = [\sum_{k=1}^{3} X_{(n)}^{(k)} (\hat{X}_{(n)}^{(k)})^{\mathrm{T}}][\sum_{k=1}^{3} \hat{X}_{(n)}^{(k)} (\hat{X}_{(n)}^{(k)})^{\mathrm{T}}]^{-1}. \tag{6}$$

SDTAE addresses temporal corruption with unknown ratios processively, meanwhile deals with spatial corruption by denoising step-by-step. In order to fill the gap between heterogeneous corruptions, we couple the corrupted data through deep CCA in output layers processively, which divides the whole problem into two simple sub-problems.

E. Coupled SDTAE (CSDTAE)

Although SDTAE is designed to solve spatiotemporal corruptions simultaneously, we need to integrate two SDTAEs into one uniform model, by maximizing their correlation step-by-step. Here CSDTAE is designed to couple two SDTAEs with temporal and spatial corruptions procedurally, which eases the heterogeneous (spatiotemporal) corruptions problem by dividing them into two sub-problems. In Figure 8, the input $\mathcal{X}_{\mathrm{T}}^{(1)}$ and output of the first temporal DTAE are the same data, but 60% corrupted input is aligned to 40% corrupted output. In the second temporal DTAE $\mathcal{X}_{\mathrm{T}}^{(2)}$, the 40% corrupted input is aligned to 20% corrupted output. In the first/second spatial DTAE $\mathcal{X}_{\mathrm{S}}^{(1)}/\mathcal{X}_{\mathrm{S}}^{(2)}$, the input and output are the same 60%/40% temporal corrupted data with arbitrary spatial noises, including Gaussian noise or block occlusion. Deep CCA is used to couple the output layers of each pair of DTAEs from the temporal and spatial scheme, to ease the diversity of temporal and spatial corruptions in different scales. CSDATE model is described in Eq (7):

$$\begin{aligned} \arg\min_{\mathcal{W},\mathcal{U},\mathcal{V}} \; & \Big\{ \|\mathcal{X}_{\mathrm{T}}^{(k)} - \mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)}\|_F^2 + \|\mathcal{X}_{\mathrm{S}}^{(k)} - \mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)}\|_F^2 \\ & - \mathrm{tr}\left(U_n^{(k)\mathrm{T}} \mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)} (\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})^{\mathrm{T}} V_n^{(k)} \right) \Big\}, \\ \text{s.t. } & \|U_n^{(k)\mathrm{T}} \mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)}\|_F^2 = 1, \quad \|V_n^{(k)\mathrm{T}} \mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)}\|_F^2 = 1, \end{aligned} \tag{7}$$

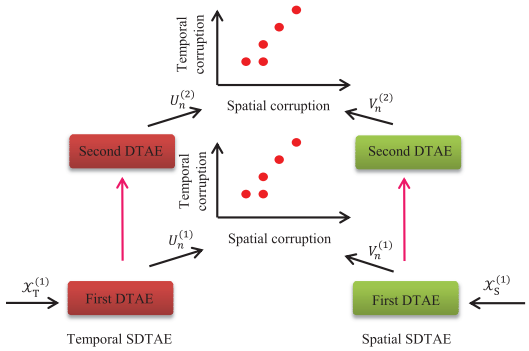


Fig. 8. CSDTAE joints the temporal and spatial schematics together, while each is a stacked scheme composed by two units (DTAEs). Left is temporal stacked scheme, which reduces the diversity of corruption ratios in a processive manner, e.g. input is 40% corrupted data and output is 20% corrupted data. Right is spatial stacked scheme, which deals with only Gaussian noise while no temporal corruption. The left and right are connected in two steps, each of which is coupled by deep CCA, making heterogeneous corrupted data closer in a common subspace.

where $k = 1, 2$ indicates the first and second DTAE, $\mathcal{X}_{\mathrm{T/S}}^{(k)}$ indicates the input in k DTAE, $\mathcal{W}$ and $\mathcal{S}$ are used for projections in temporal and spatial stacked scheme, and $U_n^{(k)}$, $V_n^{(k)}$ $(n = 1, 2, 3)$ are used for deep CCA by projecting $\mathcal{X}_{\mathrm{T}}^{(k)}$ and $\mathcal{X}_{\mathrm{S}}^{(k)}$ to one common tensor subspace. In brief, the first and second terms indicate the temporal and spatial SDATE models, respectively, and the third term aims to calculate the spatiotemporal correlation in the k-th step. The two constraints aim to ensure temporal and spatial data in orthogonal subspaces, respectively. The solution refers to [40], i.e., deep CCA is performed on the output layer of each pair of DTAEs, to make data with spatial and temporal corruptions closer in a common subspace.

We use the Augmented Lagrange Method to calculate the variables in Eq. (7) by

$$\begin{aligned} \mathcal{L} = & \|\mathcal{X}_{\mathrm{T}}^{(k)} - \mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)}\|_F^2 + \|\mathcal{X}_{\mathrm{S}}^{(k)} - \mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)}\|_F^2 \\ & - \mathrm{tr}\left(U_n^{(k)\mathrm{T}} \mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)} (\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})^{\mathrm{T}} V_n^{(k)} \right) \\ & + tr[Y_1^{(k)\mathrm{T}} (U_n^{(k)\mathrm{T}} (\mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)})(\mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)})^{\mathrm{T}} U_n^{(k)}) - 1] \\ & + tr[Y_2^{(k)\mathrm{T}} (V_n^{(k)\mathrm{T}} (\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})(\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})^{\mathrm{T}} V_n^{(k)}) - 1] \\ & + \frac{\mu}{2} [\|U_n^{(k)\mathrm{T}} (\mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)})(\mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)})^{\mathrm{T}} U_n^{(k)} - 1\|_{\mathrm{F}}^2 \\ & + \|V_n^{(k)\mathrm{T}} (\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})(\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})^{\mathrm{T}} V_n^{(k)} - 1\|_{\mathrm{F}}^2], \end{aligned} \tag{8}$$

where $Y_1^{(k)}$, $Y_2^{(k)}$ are auxiliary matrices, and μ is the penalty factor. The weighted tensor $\mathcal{W}$ can be calculated by Eq. (3), in which each matrix $W^{(k)}$ is updated by

$$W^{(k)} = (\mathcal{X}_T^{(k)} \mathcal{X}_T^{(k)T})(\hat{\mathcal{X}}_T^{(k)} \mathcal{X}_T^{(k)T})^{-1}. \tag{9}$$

The correlation matrices $U_n^{(k)}$ and $V_n^{(k)}$ are updated by

$$\begin{aligned} P_U U_n^{(k)} + U_n^{(k)} (\frac{Y_1}{\mu} - \mathbf{I}) + Q_U = 0, \\ P_V V_n^{(k)} + V_n^{(k)} (\frac{Y_2}{\mu} - \mathbf{I}) + Q_V = 0, \end{aligned} \tag{10}$$

where $Q_U = -\frac{1}{\mu}[(\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})(\mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)})^{-1}]^{\mathrm{T}} V_n^{(k)}$ is considered as a constant, so does $P_U = (\mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)})(\mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)})^T$. Besides,

$Q_V = -\frac{1}{\mu}[(\mathcal{W}\mathcal{X}_{\mathrm{T}}^{(k)})(\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})^{-1}]^{\mathrm{T}}U_n^{(k)}$ is also considered as a constant, so does $P_U = (\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})(\mathcal{S}\mathcal{X}_{\mathrm{S}}^{(k)})^T$.

V. Experiment

A. Dataset and Data Representation

1) MSR Daily Activity 3D Dataset: It contains 16 different of depth actions performed by 10 subjects, each performs every action twice. In total, there are 320 RGB and 320 depth samples. In this dataset, the first five people are used for training, and the rest for testing.

2) MSR Action Pairs Dataset: There are six pairs of depth actions performed by 10 people, with three trials each. There are a total of 360 RGB samples and 360 depth action samples. There is the same setting of training and testing samples as MSRDailyActivity3D dataset.

3) UT-Interaction#1 Dataset: This RGB dataset contains six classes, each of which has 10 sets, and each set has 100 frames. 10-fold Cross-Validation (CV) is used for testing. The-state-of-the-art methods take one frame as a sample, while in our model, all the 100 frames in one set are considered to compose one 3D sample. In other words, there are 54 samples for training and six for testing in each fold of CV.

Action representation: Each RGB/depth action video sizes $480 \times 640 \times f$, where $f \in [50, 200]$ is the number of frames. We sample each video to uniform the length of 50 frames by generating linearly space, each is resized to be 96×128 and cropped to be 48×42, therefore each video is represented to be a *third-order* tensor of size $48 \times 42 \times 50$. Here we extract HOG [13] with features of size 720×50 from each video, where 720 is the dimension of HOG.

B. Experiment Setting

In this part we introduce the details of experimental setting. As the action videos are corrupted in different ratios and uncertain locations, it leads us to find the most challenging case to evaluate our model. Therefore, we permute different corruption parts for testing as baseline, in order to select the intractable cases for training and testing. Tensor representation for action videos can help preserve spatiotemporal structure. Accordingly, we employ our one DTAE model to extract different features from each layer. Meanwhile, this tensor structure could handle both spatial and temporal corruptions of a video.

The ideal solution to our problem is to handle any corrupted ratios in a video, while exploring the reasonable correlations between different ratios. According to this, we design a stacked DTAE (SDTAE) model to leverage different ratios step-by-step. In order to deal with the complex noises, we generate a coupled SDTAE (CSDTAE) model, which couples data with heterogeneous noise, certain corrupted ratio but in uncertain locations. Consequently, a common subspace is found for testing corrupted data with all the complex noises. We also compare the result of coupling different ratios with that of coupling complex noises, to see which model can be well generalized to unknown test data.

C. Compared Methods

Here we briefly introduce the proposed method, as well as the-state-of-art AE methods for RGB-D data from the MSRDailyActivity3D and MSRActionPairs datasets, and existing compared methods for temporal corruption on the UT-interaction#1 dataset.

1) Methods for RGB-D Data:

- SVM: The baseline of the experiments, which uses HOG and linear SVM for evaluations.
- mSDA: A marginal Stacked Auto-Encoder (mSDA) [8] is proposed for denoising and uses a linear transformation as activation function to speed up.
- TAE (Ours): A basic tensor Auto-Encoder (TAE) for comparison with three hidden layers, where the first and second layers use a linear transformation on spatial feature, and the third layer performs on the temporal feature.
- DTAE (Ours): A denoising version of TAE. The input is the corrupted data, while the output is clean data or less corrupted data. We add Gaussian noises in the videos, which simulates the spatiotemporal corruption. Therefore, we could evaluate the robustness of the proposed method.
- SDTAE (Ours): It stacks a few DTAEs, which is able to progressively mitigate the corruption ratios between input and output layers. It can deal with unknown corruption ratios of the test data.
- CSDTAE (Ours): Coupled SDTAE, which is composed of SDTAE of two different corruption ratios followed by a deep CCA AE. Deep CCA (DCCA) for hidden layers enables to couple features from different TAEs to work collaboratively to recover corruption.
- C3D network [9]: C3D is a Convolution Neural Network model, which uses a 3D convolution filter on the input videos, and generates a 3D volume as output, therefore preserving temporal information of the input videos.

2) Methods for RGB Data:

- Ryoo-nondynamic: Ryoo [3] proposed a probabilistic method by calculating integral histograms for human action prediction.
- Ryoo-dynamic: Ryoo [3] used the previous likelihood to update the entire likelihood.
- SC: Cao *et al.* [5] employed sparse coding (SC) on each segmentation of a video to derive likelihood of activity.
- MSSC: Cao *et al.* [5] proposed a mixture of segments with different temporal lengths (MSSC) to deal with intra-class variations.
- Baseline: A baseline sparse coding method which takes one segment of a video as a row in the basis matrix [5].
- KNN-nondynamic: K-nearest neighbor method uses integral histograms as features.
- KNN-dynamic: K-nearest neighbor method uses previous likelihood to update the entire likelihood.

D. Experimental Results

1) DTAE and SDTAE Results: Figure 9(a) shows the results of temporal corruption, and we can see that SVM performs better without corruption of data. However, it does not work well encountering corruption at high levels. In mSDA, we use three layers as well as others, and do not add noise in the layers with a size value of 8000. The TAE, DTAE and

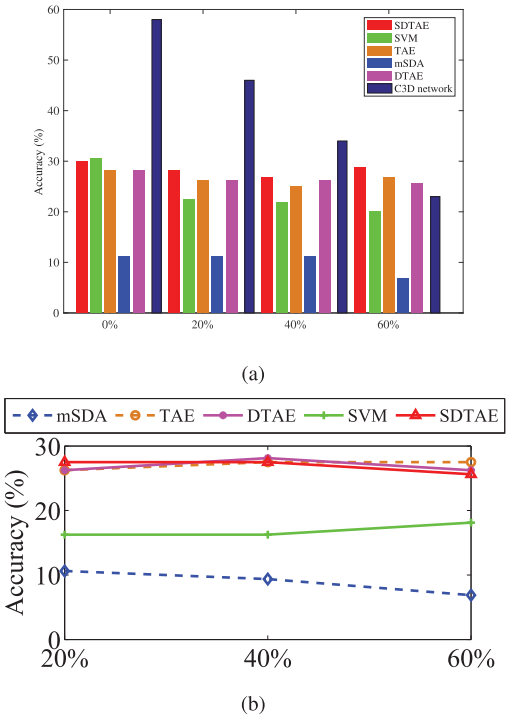


Fig. 9. Accuracy of compared methods on MSRDailyActivity3D dataset under (a) different temporal corruption ratios and (b) spatiotemporal corruption ratios. (a) Temporal corrupted. (b) Spatiotemporal corrupted.

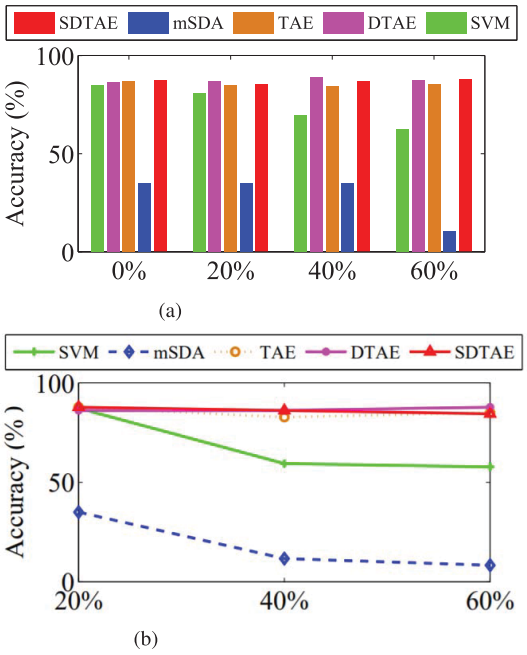


Fig. 10. Accuracy of compared methods on MSRActionPairs dataset under (a) different temporal corruption ratios and (b) spatiotemporal corruption ratios. (a) Temporal corrupted. (b) Spatiotemporal corrupted.

SDTAE obtain higher accuracies on larger corrupted levels, which means our tensor-based AE models are robust enough to handle severe corruption. SVM gets higher accuracy than SDTAE without corruption, but its performance decreases as the level of data corruption increases. Besides, SDTAE performs better than TAE and DTAE when encountering larger corruption ratios (such as 20%, 40%, 60%). We use a pre-trained C3D model and non-corrupted MSRdaily data for fine-tuning, i.e., we replace the last layer with our own output, including the number of categories. C3D network performs better on non-corrupted testing data, as the C3D model is fine-tuned by non-corrupted training data. Also, the pre-trained model includes many other training data, which is not employed by our methods. However, the accuracy goes down dramatically given larger corruptions. In the last setting with 60% corruption rate, it performs worse than our SDTAE model. It indicates that C3D does not perform well when there are lots of missing frames in one video. The reason is that C3D filter works on every 16 frames, and if the missing frames of the data are more than 16, there will be less reliable features extracted for recognition. In other words, C3D is literally affected by the quality of a video, and not robust against the corruptions in the videos.

The spatiotemporal corrupted situation (in hidden layers) is shown in Figure 9(b), and we can see that SVM and mSDA perform worse than the results of temporal corruption, which is reasonable in common sense. Besides, we add 20%, 40% and 60% noise in the hidden layers of mSDA. We can see that SVM reduces 2 $\sim$ 5% encountering spatial noise, however, our TAE, DTAE and SDTAE models perform better even encountering higher spatiotemporal corruption, which also indicates the robustness of our tensor-based AE models. In Figure 10(a), we can see that SVM gets decreasing accuracies encountering larger corruptions, while our tensor-based models perform better at higher levels of corruption. Figure 10(b) also shows that our models are robust enough to handle high levels of spatiotemporal corruption.

2) CSDTAE Results: For the 20% corrupted data, we want to verify whether the multi-location and complex noises affect the recognition. For this task, we use deep CCA to find two subspaces by U_1 and U_5 for Part1 and Part5 respectively. Here we design two groups of setting: 1) U_1 and U_5 are CCA directions to project Part1 and Part5 respectively; 2) U_1 and U_5 project Part5 and Part1 reversely. The settings aim to verify whether CCA projections work well for the testing data with unknown corrupted locations and complex noises. Table II shows the results of pair-wise locations. We can see that the performance will be degenerated by exchanging U_1 and U_5, which means that coupling pairwise parts may not work well encountering test data with uncertain corrupted locations and complex noises. The training time of CSDTAE with 20% corruption on 180 videos is 57 seconds, meaning 3 seconds for one video with 50 frames on average. The processing rate is 17 fps.

Focusing on the corrupted data with unknown locations of temporal corruption and complex noises, we design the model to couple the 20% data with all the corrupted locations and heterogeneous noises of the 0% corrupted data for training, with the aim to level up the irregular locations and noises, and obtain the common projection designated hybrid U. In Table III, the first two rows show the results of each part with U_1 and U_5 projection by different training parts, and the last two rows give the results with hybrid U with all parts for training, where T and ST mean the temporal and spatiotemporal corruption, respectively. We can see that U

TABLE II

ACCURACY OF 20% CORRUPTION ON MSRACTIONPAIRS DATASET. IN ORDER TO VERIFY WHETHER THE DIFFERENT CORRUPTED LOCATIONS AND COMPLEX NOISES HAVE AN EFFECT ON THE PERFORMANCE, WE USE PART1 AND PART5 FOR TRAINING, AND PROJECT THEM TO THE CORRESPONDING SUBSPACES BY U_1 AND U_5 FOR TESTING

Methods Missing	Project different parts to different subspaces	20 % corruption Part1	Part5
SDTAE	$U_1 \rightarrow$ Part1, $U_5 \rightarrow$ Part5	**86.11**	82.22
CSDTAE	$U_1 \rightarrow$ Part1, $U_5 \rightarrow$ Part5	85.00	**83.33**
SDTAE	$U_1 \rightarrow$ Part5, $U_5 \rightarrow$ Part1	83.33	82.77
CSDTAE	$U_1 \rightarrow$ Part5, $U_5 \rightarrow$ Part1	83.88	82.78

TABLE III

ACCURACY OF 20% CORRUPTION WITH DIFFERENT LOCATIONS ON MSRACTIONPAIRS DATASET. THE FIRST TWO ROWS SHOW THE RESULTS FOR WHEN WE COUPLED ONLY PART1 AND PART5 FOR TRAINING, AND THE LAST TWO ROWS DISPLAY THE RESULTS FOR WHEN WE COUPLED ALL THE PARTS WITH COMPLETE DATA FOR TRAINING. THE BLUE FONT INDICTS LOWER ACCURACIES

Training part	20 % corruption Part1	Part2	Part3	Part4	Part5
Part1 (U_1)	**83.33**	**81.66**	**82.77**	**76.11**	78.33
Part5 (U_5)	**74.44**	**77.22**	**74.44**	**73.33**	**76.11**
All parts (U, T)	82.77	84.44	86.11	82.22	77.77
All parts (U, ST)	81.66	84.44	85.00	81.66	79.44

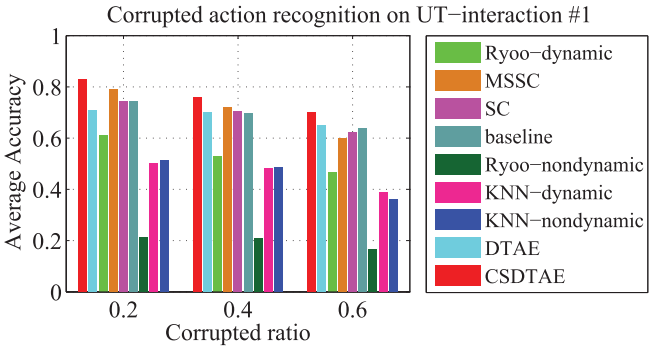


Fig. 11. Temporal corrupted action recognition on UT-interaction#1 dataset. We can see that the proposed DTAE model is competitive while CSDTAE model performs better than others under certain levels of corruption.

performs better than U_1 and U_5 in terms of accuracy for most cases. This indicates that using all the hybrid parts for training works well on the unknown test datasets as compared with the pair-wise training (lower accuracies in *blue font*). Therefore CSDTAE is robust by the proper setting for training.

Figure 11 shows the results of our two models 1) DTAE and 2) CSDTAE on the UT-interaction#1 dataset for temporal corrupted action recognition compared with the-state-of-the-art methods detailed in [5]. The data are corrupted with Gaussian noise and block occlusion for coupling. DTAE gets the competitive results with others, while CSDTAE obtains better results than all the others at different corrupted levels, which means the coupled model is robust and can handle data with a higher ratio of data corruptions.

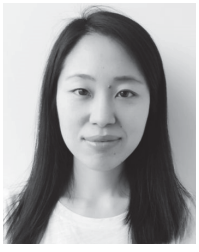
VI. CONCLUSION

We proposed a CSDTAE model, which joints the temporal and spatial SDTAEs to ease spatiotemporal corrupted problem in a divide-and-conquer scheme. For spatial/temporal corrupted action videos with different ratios, we generated an SDTAE model in a stacked mechanism to reduce the diversity processively and to mitigate the affect of corruption. To handle temporal corruption, each DTAE in SDTAE dealt with different corruption ratios. For spatial corruption, each DTAE contained the same temporal corruption ratio but different spatial noise. The two SDTAEs were considered as sub-problems and integrated into one framework by coupling the output layers using deep CCA in each DTAE. Experiments on three action datasets demonstrated the effectiveness of our models, especially for large volumes of spatiotemporal corruption in the videos.

REFERENCES

[1] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. CVPR*, 2015, pp. 4305–4314.
[2] W. Zhu *et al.* (2016). "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks." [Online]. Available: https://arxiv.org/abs/1603.07772
[3] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proc. ICCV*, 2011, pp. 1036–1043.
[4] Y. Kong, D. Kit, and Y. Fu, "A discriminative model with multiple temporal scales for action prediction," in *Proc. ECCV*, 2014, pp. 596–611.
[5] Y. Cao *et al.*, "Recognize human activities from partially observed videos," in *Proc. CVPR*, 2013, pp. 2658–2665.
[6] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[7] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.
[8] M. Chen, K. Q. Weinberger, F. Sha, and Y. Bengio, "Marginalized denoising auto-encoders for nonlinear representations," in *Proc. ICML*, 2014, pp. 1476–1484.
[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. ICCV*, 2015, pp. 4489–4497.
[10] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int. J. Comput. Vis.*, vol. 79, no. 3, pp. 299–318, 2008.
[11] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. ACM MM*, 2007, pp. 357–360.
[12] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.
[13] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. CVPR*, 2013, pp. 716–723.
[14] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proc. CVPR*, 2009, pp. 1932–1939.
[15] H. Zhao and Y. Fu, "Semantic single video segmentation with robust graph representation," in *Proc. IJCAI*, 2015, pp. 2219–2226.
[16] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.
[17] M. Raptis and S. Soatto, "Tracklet descriptors for action modeling and video analysis," in *Proc. ECCV*, 2010, pp. 577–590.
[18] C. Jia, M. Shao, and Y. Fu, "Sparse canonical temporal alignment with deep tensor decomposition for action recognition," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 738–750, Feb. 2017.
[19] M. Griebel and H. Harbrecht, "A note on the construction of L-fold sparse tensor product spaces," *Construct. Approx.*, vol. 38, no. 2, pp. 235–251, 2013.
[20] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 288–303, Feb. 2010.

[21] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, 2011.
[22] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. NIPS*, 2013, pp. 809–817.
[23] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, "High-level event recognition in unconstrained videos," *Int. J. Multimedia Inf. Retr.*, vol. 2, no. 2, pp. 73–101, Jun. 2013.
[24] S. Jiang, M. Shao, C. Jia, and Y. Fu, "Learning consensus representation for weak style classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[25] M. Shao, Z. Ding, H. Zhao, and Y. Fu, "Spectral bisection tree guided deep adaptive exemplar autoencoder for unsupervised domain adaptation," in *Proc. AAAI*, 2016, pp. 2023–2029.
[26] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.
[27] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. ICASSP*, 2013, pp. 3377–3381.
[28] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Jun. 2006, pp. 519–528.
[29] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *Proc. ICCV*, 2009, pp. 1593–1600.
[30] UT-Interaction Dataset. (2010). *ICPR Contest on Semantic Description of Human Activities (SDHA)*. [Online]. Available: http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html
[31] Y. M. Lui, "Tangent bundles on special manifolds for action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 6, pp. 930–942, Jun. 2012.
[32] Y. M. Lui and J. R. Beveridge, "Tangent bundle for human action recognition," in *Proc. FGR*, 2011, pp. 97–102.
[33] Y. Makihara, A. Mansur, D. Muramatsu, Z. Uddin, and Y. Yagi, "Multi-view discriminant analysis with tensor representation and its application to cross-view gait recognition," in *Proc. FGR*, 2015, pp. 1–8.
[34] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1944–1957, Aug. 2013.
[35] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with reconstruction cost for efficient overcomplete feature learning," in *Proc. NIPS*, 2011, pp. 1017–1025.
[36] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. CVPR*, Jun. 2011, pp. 3361–3368.
[37] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Spatio-temporal convolutional sparse auto-encoder for sequence classification," in *Proc. BMVC*, 2012, pp. 1–12.
[38] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. ICML*, 2011, pp. 689–696.
[39] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. ICML*, 2013, pp. 1247–1255.
[40] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. ICML*, 2015, pp. 1083–1092.
[41] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, 2008, pp. 1096–1103.
[42] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.



Chengcheng Jia received the B.E. degree from Northeastern Normal University in 2007, the M.E. and the Ph.D. degrees in computer science from Jilin University, China, in 2010 and 2013, respectively, and the Ph.D. degree from the Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA, in 2016. She is currently a Computer Vision Engineer with Huawei Technologies, Santa Clara, CA, USA, where she is involved in automatic driving. Her research interests include machine learning, computer vision, deep learning, and tensor factorization.

Ming Shao (S'11–M'16) received the B.E. degree in computer science, the B.S. degree in applied mathematics, and the M.E. degree in computer science from Beihang University, Beijing, China, in 2006, 2007, and 2010, respectively, and the Ph.D. degree in computer engineering from Northeastern University, Boston, MA, USA, in 2016. He is currently a tenure-track Assistant Professor with the College of Engineering, University of Massachusetts Dartmouth, since 2016. His current research interests include sparse modeling, low-rank matrix analysis, deep learning, and applied machine learning on social media analytics. He was a recipient of the Presidential Fellowship of State University of New York at Buffalo from 2010 to 2012, and the Best Paper Award Winner/Candidate of the IEEE ICDM 2011 Workshop on Large Scale Visual Analytics and ICME 2014. He has served as a reviewer for many IEEE transactions and journals, including TPAMI, TKDE, TNNLS, TIP, and TMM. He has also served on the program committee for the conferences, including AAAI, IJCAI, and FG.

Sheng Li (S'11–M'17) received the B.Eng. degree in computer science and engineering and the M.Eng. degree in information security from the Nanjing University of Posts and Telecommunications, China, and the Ph.D. degree in computer engineering from Northeastern University, Boston, MA, USA, in 2010, 2012, and 2017, respectively. He is currently a Data Scientist at Adobe Research, San Jose, CA, USA. He has published over 50 papers at leading conferences and journals. He received the Best Paper Awards (or nominations) at SDM 2014, the IEEE ICME 2014, and the IEEE FG 2013. He serves on the Editorial Board of Neural Computing and Applications, and serves as an Associate Editor of *IET Image Processing*. He has also served as a reviewer for several the IEEE Transactions and a program committee member for IJCAI, AAAI, KDD, the IEEE FG, PAKDD, and DSAA. His research interests include robust machine learning, visual intelligence, and behavior modeling.

Handong Zhao (S'15) received the B.Eng. degree in computer science and the M.Eng. degree in computer technology from Tianjin University, China, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. He has a broad research interest in machine learning, data mining, and computer vision. He holds the Dean's Fellowship at Northeastern. He was a recipient of the Best Paper Honorable Mention Award at the 2013 ACM International Conference on Internet Multimedia Computing and Service.

Yun Fu (S'07–M'08–SM'11) received the M.S. degree in statistics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign, respectively. He is currently an Interdisciplinary Faculty Member with the College of Engineering and the College of Computer and Information Science, Northeastern University, since 2012. His research interests are machine learning, computer vision, social media analytics, and big data mining. He is a fellow of IAPR and SPIE, a Lifetime Senior Member of the ACM, a Lifetime Member of AAAI, OSA, and the Institute of Mathematical Statistics, a member of the Global Young Academy, INNS, and a Beckman Graduate Fellow during 2007 to 2008. He received seven Young Investigator Awards: the 2016 National Academy of Engineering Grainger Foundation Frontiers of Engineering Award, the 2016 IEEE CIS Outstanding Early Career Award, the 2016 UIUC ECE Young Alumni Achievement Award, the 2015 National Academy of Engineering US Frontiers of Engineering, the 2014 ONR Young Investigator Award, the 2014 ARO Young Investigator Award, and the 2014 INNS Young Investigator Award, nine Best Paper Awards: the IEEE TAC 2017, the ACM MM 2017, the SPIE DSS 2016, the SIAM SDM 2014, the IEEE ICME 2014 Candidate, the IEEE FG 2013, the IEEE ICDM-LSVA 2011, the IAPR ICFHR 2010, and the IEEE ICIP 2007, three Industrial Research Awards: the 2016 Samsung GRO Award, the 2015 Adobe Faculty Research Award, and the 2010 Google Faculty Research Award, and two Service Awards: the 2012 IEEE TCSVT Best Associate Editor and the 2011 IEEE ICME Best Reviewer. He has extensive publications and serves as an associate editor, the chair, a PC member, and a reviewer for many top journals and international conferences/workshops.