I Know the Relationships: Zero-Shot Action Recognition via Two-Stream Graph Convolutional Networks and Knowledge Graphs

Junyu Gao, Tianzhu Zhang, Changsheng Xu
National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
University of Chinese Academy of Sciences
gaojunyu2015@ia.ac.cn, tzzhang10@gmail.com, csxu@nlpr.ia.ac.cn

Abstract

Recently, with the ever-growing action categories, zero-shot action recognition (ZSAR) has been achieved by automatically mining the underlying concepts (e.g., actions, attributes) in videos. However, most existing methods only exploit the visual cues of these concepts but ignore external knowledge information for modeling explicit relationships between them. In fact, humans have remarkable ability to transfer knowledge learned from familiar classes to recognize unfamiliar classes. To narrow the knowledge gap between existing methods and humans, we propose an end-to-end ZSAR framework based on a structured knowledge graph, which can jointly model the relationships between action-attribute, action-action, and attribute-attribute. To effectively leverage the knowledge graph, we design a novel Two-Stream Graph Convolutional Network (TS-GCN) consisting of a classifier branch and an instance branch. Specifically, the classifier branch takes the semantic-embedding vectors of all the concepts as input, then generates the classifiers for action categories. The instance branch maps the attribute embeddings and scores of each video instance into an attribute-feature space. Finally, the generated classifiers are evaluated on the attribute features of each video, and a classification loss is adopted for optimizing the whole network. In addition, a self-attention module is utilized to model the temporal information of videos. Extensive experimental results on three realistic action benchmarks Olympic Sports, HMDB51 and UCF101 demonstrate the favorable performance of our proposed framework.

Introduction

Recent studies on supervised action recognition have advanced rapidly because of the development of deep learning techniques and large-scale labeled datasets. However, with the growing number of action categories, traditional approaches suffer from the scalability problem (Xu, Hospedales, and Gong 2016). These methods require large numbers of costly and laboriously annotated videos per action class, making them not generalized for unseen categories. To overcome such an issue, Zero-Shot Action Recognition (ZSAR) has recently drawn considerable attention since it provides an alternative methodology that does not require any positive exemplars for classifying novel categories (Liu, Kuipers, and Savarese 2011; Xu, Hospedales, and Gong 2016; Qin et al. 2017; Zhu et al. 2018).

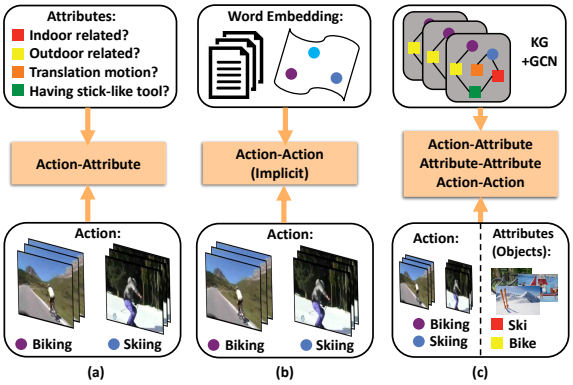

Figure 1: Three ZASR frameworks. (a) Attribute-based framework only considers the action-attribute relationships. (b) Word embedding-based framework which implicitly models the action-action relationships. (c) Our framework can directly and collectively consider all the three types of relationships.

Existing work on ZSAR generally follows two research lines: (1) As shown in Figure 1 (a), some methods utilize human-defined attributes for classification (Liu, Kuipers, and Savarese 2011), which only leverage the action-attribute relationships to distinguish novel action categories. Moreover, these attribute-based methods are hard to generalize to arbitrary zero-shot categories in a practical scenario due to the difficulties in attribute definition. (2) Other methods adopt semantic representations (e.g., word embeddings) of action names to model action-action relationship in a semantic space (Xu, Hospedales, and Gong 2016; 2017; Qin et al. 2017), as shown in Figure 1 (b). Although these approaches are simple and effective, the word embedding space can only represent action-action relationships in an implicit way. Moreover, these methods can hardly get benefit from the other side information of videos. Recently, inspired by the strong relationships between objects and actions, (Jain et al. 2015; Mettes and Snoek 2017) employ objects as attributes for ZSAR and achieve favorable perfor-

mance, where pre-trained object classifiers are used to find objects in an action video. Nevertheless, they only consider the action-object relationship based on the fixed similarity scores of word embedding vectors, which cannot get benefit from an end-to-end training.

In addition to the aforementioned issues, most of the above methods only focus on the visual cues in videos but ignore external knowledge information to improve the generalization capability of zero-shot approaches. Actually, humans have remarkable ability to recognize large-scale concepts by using semantic knowledge of the world learned through experience. Therefore, it is a natural way to use structured knowledge information to model relationships among concepts (e.g., actions and attributes), and this research direction is able to transfer learned knowledge from seen categories to unseen ones. Recently, knowledge graphs (KGs) have been successfully used in various computer vision tasks such as object detection (Fang et al. 2017), multi-label image classification (Marino, Salakhutdinov, and Gupta 2017), zero-shot image recognition (Wang, Ye, and Gupta 2018; Lee et al. 2018), etc. By incorporating KGs in these methods, the performance is significantly improved, showing KGs do have remarkable ability to bridge the knowledge gap in existing state-of-the-art approaches. Therefore, great potential is excepted to exploit knowledge graphs for ZSAR. Moreover, current methods usually ignore temporal modeling of videos, such as simply performing average pooling across video frames (Jain et al. 2015) or extracting a hand-crafted feature vector of the whole video (Qin et al. 2017). However, significant advantages of exploiting temporal information for video understanding have been shown recently (Niebles, Chen, and Fei-Fei 2010). Temporal modeling in a video sequence can help understand its dynamic patterns and further boost the ZSAR performance.

Motivated by the above observations, as shown in Figure 1 (c), we propose a novel ZSAR framework to directly and collectively model all the three types of relationships between action-attribute, action-action, and attribute-attribute by incorporating a knowledge graph in an end-to-end manner. In fact, all these relationships can contribute to the ZSAR task either in an explicit or implicit way. Here, to avoid cumbersome attribute-annotation, we adopt objects as attributes as it in (Mettes and Snoek 2017). To effectively leverage the knowledge information in a knowledge graph, we use Graph Convolutional Network (GCN) (Kipf and Welling 2016) to model the dependencies and propagate messages between different concepts in the knowledge graph. Specifically, we propose a Two-Stream GCN (TS-GCN) consisting of a classifier branch and an instance branch, where KGs are incorporated into both branches to model the three types of relationships, as shown in Figure 2. The classifier branch aims to generate classifiers for different action categories, which takes as input a set of concepts and their corresponding word-embedding vectors. The instance branch is designed to produce attribute-feature of a video instance by leveraging the object scores obtained from the video. We finally optimize the whole framework via a classification loss, using the generated classifiers and the attribute features of training videos. In addition, to perform temporal modeling of videos, we incorporate a self-attention module (Zhang et al. 2018) into the instance branch to model the dynamically changing object scores over time. During training, the classifiers for seen categories are learned in a supervised fashion. At test phase, the trained model is used to predict the classifiers of unseen categories and conduct classification on the attribute features of test videos.

The main contributions are highlighted as follows:

- We propose a novel two-stream GCN framework that can effectively leverage knowledge graphs to model the relationships between action-attribute, action-action, and attribute-attribute. To the best of our knowledge, our method is among the first to advance knowledge graphs and two-stream GCN for ZSAR.
- By designing both classifier branch and instance branch in a sharing knowledge space, the generated action classifiers and instance features can co-adapt and cooperate to achieve the classification objective in an end-to-end fashion. In addition, a self-attention module is embedded to the framework for temporal modeling.
- The proposed framework performs favorably against the state-of-the-art methods on three standard ZSAR datasets, which verifies its effectiveness.

Related Work

Zero-shot Action Recognition. With the explosive growth of action videos and the successful deep learning-based computer vision tasks (Zhang, Xu, and Yang 2017; Gao et al. 2017; 2018), the focus has now shifted to scaling visual recognition systems in terms of categories (Zhu et al. 2018; Xu, Hospedales, and Gong 2016; Gao, Zhang, and Xu 2017; Zhang et al. 2012a; 2010; 2012b; Zhang, Xu, and Yang 2018a; 2018b; Zheng et al. 2017; Han et al. 2018). Zero-shot learning addresses this issue by mining the knowledge of how unseen classes is semantically related to the known classes. Early work on ZSAR uses human annotated attributes. Liu et al. (Liu, Kuipers, and Savarese 2011) propose a latent SVM model where latent variables determine the importance of each attribute for each action class. Gan et al. (Gan, Yang, and Gong 2016) treat each category as a domain, and tackle attribute detection from the multi-source domain generalization point of view. However, the manually-specified attributes are highly subjective and cumbersome to annotate. For this reason, word embeddings have been preferred recently for addressing ZSAR. Xu et al. (Xu, Hospedales, and Gong 2017; 2016) explore word vectors as a shared semantic space to embed labels and videos for ZSAR. Qin et al. (Qin et al. 2017) adopt error-correcting output codes to address domain shift problem, which utilizes both category-level semantics and intrinsic data structures. Recently, some methods reveal that object scores are well-suited for video recognition. The work Objects2action (Jain et al. 2015) constructs a semantic embedding model by considering thousands of object categories. Spatial-aware object embedding is further designed for zero-shot localization and classification of actions (Mettes and Snoek 2017). In addition, recent work propose ZSAR by exploiting the

semantic relationships of concepts such as inter-class relationship (Gan et al. 2015) and pairwise relationship (Gan et al. 2016a). Gan et al. (Gan et al. 2016b) use knowledge information to build an analogy pool according to an external ontology for zero-shot action recognition. Nevertheless, these methods are not end-to-end trainable. Different from these methods, we incorporate knowledge graphs into a novel two-stream ZSAR framework via GCNs, which can explicitly model the relationships among object and action categories in an end-to-end manner.

Knowledge Distillation Using Graph Neural Networks. Generalization of neural networks for arbitrarily structured graphs (such as knowledge graphs) has drawn great attention in recent years. For knowledge distillation, Marino et al. (Marino, Salakhutdinov, and Gupta 2017) introduce a graph search neural network (GSNN), which can exploit large knowledge graphs into an end-to-end framework for image classification. Gao et al. (Gao, Zhang, and Xu 2018) adopt a graph convLSTM to model the dynamic knowledge evolution for video classification. Lee et al. (Lee et al. 2018) utilize a graph gated neural network to model knowledge graphs for describing the relationships between multiple labels. Although this method achieves favorable performance, it simply learns a single classifier for all classes and does not incorporate unseen class labels in the training stage, which might loss discriminative ability. Given a knowledge graph, Wang et al. (Wang, Ye, and Gupta 2018) design a zero-shot recognition model by taking as input semantic embeddings for each node in a graph convolution network. The objective function of this model is a mean-square error between the predicted and ground truth classifiers of known classes. This might limit its generalization ability since the ground truth classifiers are fixed. Moreover, this method is deficient since it only considers label-label relationships without attributes. To distill knowledge information from actions and objects (attributes), we design both classifier branch and instance branch based on GCNs (Kipf and Welling 2016). By leveraging the knowledge relationships among actions and objects, the proposed method can directly adopt classification loss and jointly learn different classifiers for each action, which results in favorable generalization ability.

Our Approach

In ZSAR, suppose we have N_s labeled videos $\mathcal{D}^s = \{\mathcal{V}^s, \mathcal{Y}^s\}$ from a source dataset with S seen categories $\mathcal{Y}^s$, where each video $\mathbf{V}^s \in \mathcal{V}^s$ is associated with an action label $y^s \in \mathcal{Y}^s$. Similarly, there is a target dataset $\mathcal{D}^t = \{\mathcal{V}^t, \mathcal{Y}^t\}$ consisting of N_t videos from U unseen action classes $\mathcal{Y}^u$. Here, $\mathcal{Y}^s \cup \mathcal{Y}^u = \mathcal{Y}$, $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. In addition, we have an object set $\mathcal{O}$ with O objects, which serve as attributes for describing the actions. The goal is to learn an objective function $\min \mathcal{L}(\mathcal{D}^s, \mathcal{O})$ that can generalize to $\mathcal{D}^t$.

Different from existing zero-shot methods which do not explicitly consider the relationships among concepts with structured knowledge graphs (Xu, Hospedales, and Gong 2016; Zhu et al. 2018) or only model label-label relationships (Wang, Ye, and Gupta 2018), we design an end-to-end framework consisting of two GCN branches: classifier branch and instance branch, as shown in Figure 2. In the following of this section, we illustrate our proposed approach for ZSAR in details. We first introduce the preliminaries of our main building block, graph convolutional network (Kipf and Welling 2016), which enables us to generalize CNN to graphs. Then, we present the whole model including the classifier branch and instance branch. Finally, the implementation details of our framework are demonstrated.

Graph Convolutional Networks

Graph convolutional networks aim to efficiently learn layer-wise propagation operations that can be applied directly on graphs. To keep this paper self-contained, we briefly introduce GCNs proposed in (Kipf and Welling 2016) as follows.

Given an undirected graph with m nodes, a set of edges between nodes, an adjacency matrix $\mathbf{A} \in \mathrm{R}^{m \times m}$, and a degree matrix $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. We consider a linear formulation of graph convolution as the multiplication of a graph signal $\mathbf{X} \in \mathrm{R}^{k \times m}$ (the column vector $\mathbf{X}_i \in \mathrm{R}^k$ is the feature representation at the i^{th} node) with a filter $\mathbf{W} \in \mathrm{R}^{k \times c}$:

$$\mathbf{Z} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X}^\top \mathbf{W}, \tag{1}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{I}$ is the identity matrix. $\hat{\mathbf{D}}_{ii} = \sum_j \hat{\mathbf{A}}_{ij}$. As a result, the input to a graph convolutional layer is $k \times m$, and the output is a $c \times m$ matrix $\mathbf{Z}$. Note that a GCN can be built by stacking multiple graph convolutional layers of the form of Eq. (1), each layer followed by a non-linear operation (such as ReLU). Readers can refer to (Kipf and Welling 2016) for more details and an in-depth discussion.

Two-stream GCN for ZSAR

To effectively use the explicit relationships among all the concepts, we incorporate a knowledge graph into our proposed method. Since we have $S + U + O$ concepts (seen action, unseen action, objects) associated with all the videos, we build the knowledge graph with the same number of nodes corresponding to these concepts. We use the term *concept* and *node* interchangeably hereafter. The graph structure is represented as an adjacency matrix, $\mathbf{A}$. Figure 2 shows the proposed two-stream GCN architecture consisting of both classifier branch and instance branch.

Classifier Branch. This branch is an L-layer GCN where each layer l takes as input the feature matrix ($\mathbf{Z}^{cls}_{l-1}$) generated from the previous layer and produces a feature matrix $\mathbf{Z}^{cls}_l$. The input to this branch is a $k \times (S + U + O)$ matrix $\mathbf{X}^{cls}$ which is the word-embedding vectors of all the concepts. Here, k is the dimensionality of the word-embedding vector. The output of the final layer is a $d \times (S + U + O)$ matrix $\mathcal{W}^{cls}$ where d is the dimensionality of the classifiers. Specifically, S classifiers $\mathcal{W}^{cls}_{1:S}$ are corresponding to the seen action categories $\mathcal{Y}^s$, which are optimized using the training data. During the training phase, another U unseen classifiers can be generalized from these seen ones via GCN. Note that the remaining O object classifiers serve as a bridge between seen and unseen action categories, which will not be explicitly used in the training/inference phase.

Instance Branch. The branch aims to produce the attribute-feature for video instances. Since video temporal informa-

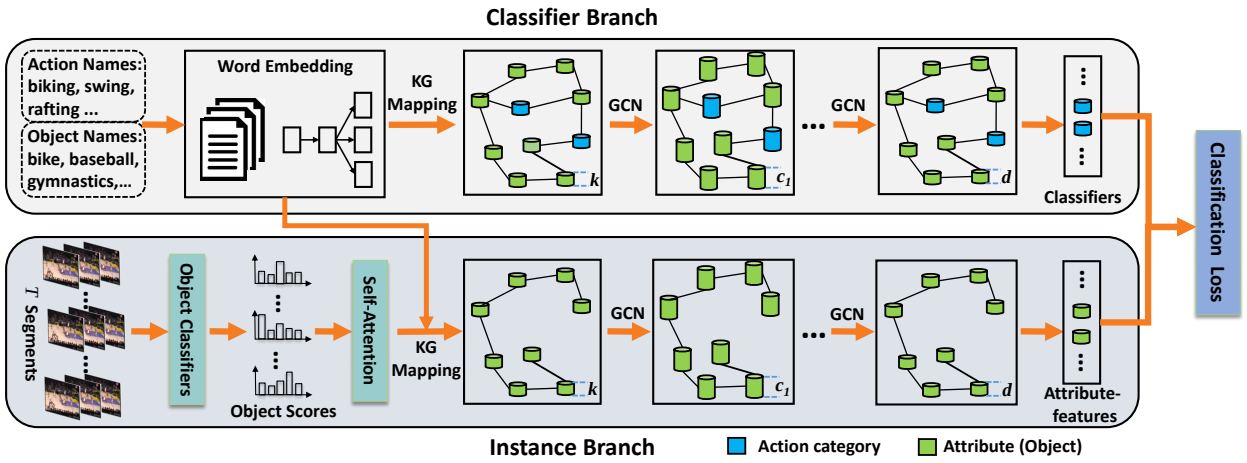

Figure 2: Our TS-GCN framework consists of both classifier branch and instance branch. We adopt a classification loss to optimize the whole framework. KG mapping means organizing concept representations as nodes in the knowledge graph.

tion plays an essential role in video understanding (Tran et al. 2015), this branch firstly conducts video temporal modeling via a self-attention module. Specifically, for a video $\mathbf{V}$, we first segment it into T equal-length segments $\{\mathbf{V}_t\}_{t=1}^{T}$. The number of segments is fixed for all videos in order to perform sequential parallelization in our framework. To get the object scores of each segment, we follow (Mettes and Snoek 2017) to employ a GoogLeNet model (Szegedy et al. 2015), trained on a 12,988-category shuffle (Mettes, Koelma, and Snoek 2016). Similar to (Jain et al. 2015), the top K most relevant objects are selected for each action, which results in O objects from the initial 12,988 categories. For the frames in a segment, we average the object probabilities at the softmax layer of this model. Therefore, the video $\mathbf{V}$ is represented as an $O \times T$ matrix. To employ the temporal information in this video, a self-attention operator is performed on $\mathbf{V}$ as follows:

$$\begin{aligned} \alpha_{s,t} &= \frac{\exp(f(\mathbf{V}_t)^\top g(\mathbf{V}_s))}{\sum_{t=1}^{T}\exp(f(\mathbf{V}_t)^\top g(\mathbf{V}_s))} \\ \hat{\mathbf{V}}_s &= \gamma \sum_{t=1}^{T} \alpha_{s,t} h(\mathbf{V}_t) + \mathbf{V}_s \end{aligned} \tag{2}$$

where $\mathbf{V}_t$ and $\mathbf{V}_s$ are the object scores of segment t and s. $f(\cdot), g(\cdot), h(\cdot)$ are three 1×1 convolutional layers with O filters. $\alpha_{s,t}$ is the attention weight which indicates the contribution of segment t to the representation of segment s. We finally multiply the output of the attended representation by a scale parameter γ and add back the input representation. γ is initialized as 0. After the self-attention operator, we get the representation of the video $\hat{\mathbf{V}}$, which is the same size as $\mathbf{V}$. $\hat{\mathbf{V}}$ and the word-embedding vectors of the object categories are used to generate the input to the following L graph convolutional layers, $\mathbf{X}^{ins} \in \mathrm{R}^{k \times O}$. Here, the o^{th} column of $\mathbf{X}^{ins}$ is calculated as:

$$\mathbf{X}_o^{ins} = \sum_{t=1}^{T} \hat{\mathbf{V}}_{t,o} \mathbf{s}_o, \tag{3}$$

where $\mathbf{s}_o$ is the word-embedding vector of the o^{th} object. Note that there are only O nodes in the instance branch, which means this branch focuses on robust attribute-feature generation of video instances. For the final-layer, the output feature-matrix is $\mathbf{Z}_L^{ins} \in \mathrm{R}^{O \times d}$. d is the same as the dimensionality of classifiers produced by the classifier branch.

Loss-function. For the S seen categories, we evaluate the cross-entropy loss over all the labeled examples:

$$\mathcal{L} = -\frac{1}{N_s} \sum_{n=1}^{N_s} \sum_{i=1}^{S} y_n^i \log(p_n^i), \tag{4}$$

where y_n^i is the ground-truth label (0 or 1) of the n^{th} training video with respect to the i^{th} seen action. p_n^i is the predicted score with a softmax operation via the two branches:

$$p_n^i = \frac{\exp(q_n^i)}{\sum_{i=1}^{S}\exp(q_n^i)}, \text{where } q_n^i = (\mathcal{W}_i^{cls})^\top \sum_{o \in \mathcal{N}(i)} \mathbf{Z}_{L,n,o}^{ins}, \tag{5}$$

where $\mathcal{W}_i^{cls}$ is the i^{th} action classifier produced by the classifier branch. And $\mathbf{Z}_{L,n,o}^{ins}$ is the final feature vector generated from the instance branch, which indicates the o^{th} object-feature (in layer L) of the n^{th} video. $\mathcal{N}(i)$ denotes the one-hop object neighbors of the i^{th} action in the knowledge graph, which means we focus on strongly-related objects for classifying a specific action. Empirically, we find that using neighbors leads to faster convergence and higher performance than using all the object features. In fact, with the message-passing of GCNs, the useful information is propagated and augmented to $\mathcal{N}(i)$ via the optimization of the framework. Moreover, using only neighbors can avoid some distraction in action classification.

Generalization to the Unseen actions. During training, we are able to not only optimize the classifiers of the S seen categories but also generalize to zero-shot categories via the relationship modeling of the two-stream GCN. At test phase, we use the generated classifiers of unseen categories (from classifier branch) to perform classification on the object features of test videos (from instance branch) as it in Eq. (5).

Discussion. The proposed two-stream framework is suitable for zero-shot learning problem with attributes. We utilize GCNs to transfer information between different concepts. Concretely, both branches jointly model the relationships between action-attribute, action-action, and attribute-attribute in a knowledge space via knowledge graphs. As a result, the learned classifiers (classifier branch) can effectively evaluate the generated attribute features (instance branch) of each video in an end-to-end fashion. Because of the joint learning with a single objective function, both branches can co-adapt and cooperate. Here, a better classifier can make the learned features more robust, and a better feature can make the learned classifiers more accurate. In this work, we utilize objects as attributes, which avoids the cumbersome annotation. Note that other information such as the visual representations of videos can also be added into the framework, which is left for our future work.

Implementation Details

Knowledge Graph Building. We use the off-the-shelf ConceptNet 5.5 (Speer, Chin, and Havasi 2017) to build our knowledge graph, which connects words and phrases of natural language edges. Its knowledge is collected from many sources including WordNet (Bond and Foster 2013), DBpedia (Auer et al. 2007) et al. Following previous work (Fang et al. 2017), we only employ its English subgraph with about 1.5 million nodes. We adopt string matching to map the concepts to the nodes in ConceptNet. Since some names of concepts have no corresponding nodes due to its rare appearance, we replace these terms with common words that can be found in ConceptNet. For instance, "skijet" is replaced by "jetski" without losing the main semantic information. The most important thing for building the knowledge graph is to determine the relationships (edges, $\mathbf{A}_{ij}$) among these nodes. Specifically, if both nodes can be found in ConceptNet with one edge being connected, we use the weight corresponding to this edge as $\mathbf{A}_{ij}$. While a knowledge graph may have multiple types of edges, we follow some previous methods (Marino, Salakhutdinov, and Gupta 2017; Fang et al. 2017) to simplify it as a single matrix (adjacency matrix) to effectively represent the semantic consistency and propagate information between nodes. Although we can fine-tune the adjacency matrix $\mathbf{A}$ during training like (Lee et al. 2018), we choose to fix it following (Marino, Salakhutdinov, and Gupta 2017). The reasons are two-fold: (1) fixing $\mathbf{A}$ is more computationally efficient than fine-tuning it. (2) Fine-tuning $\mathbf{A}$ will change the intrinsic knowledge structures in it, resulting in losing generalization ability. For each action, we select $K = 100$ objects with the highest weights.

Word Embedding. Following (Mettes and Snoek 2017), we utilize the skip-gram network of word2vec trained on the metadata of the images and videos from the YFCC100M dataset (Thomee et al. 2016). The trained model produces a 500-dimensional representation for each word. To represent each concept in a fixed length, we simply average all the word vectors (Mettes and Snoek 2017).

Model Details. Both GCNs in the two streams are composed of 3 graph convolutional layers with output channel dimensionality of 2048, 1024, 512, respectively. Note that for the classifier stream, all the concepts are considered which results in an $(S + U + O) \times (S + U + O)$ adjacency matrix $\mathbf{A}^{cls}$, while only O object concepts are employed for constructing the knowledge graph $\mathbf{A}^{ins} \in \mathrm{R}^{O \times O}$ in the instance stream. Following (Wang, Ye, and Gupta 2018), we apply the LeakyReLU as the activation function after each graph convolutional layer. We also perform L2-Normalization on the produced classifiers to regularize them into similar magnitudes. For the self-attention module, the number of segments T is set to 16 by grid search over $\{8,16,32\}$. To train our whole model, we use the ADAM (Kingma and Ba 2014) optimizer with learning rate 0.0001 and weight decay 0.0005. The model is trained for 5 epochs with a batch size of 48. We implement our framework by Tensorflow (Abadi et al. 2016). The code for our framework can be found in https://github.com/junyuGao/Zero-Shot-Action-Recognition-with-Two-Stream-GCN.

Experiments

In this section, we evaluate the performance of the proposed Two-Stream GCN (TS-GCN) method on three widely-used video datasets: Olympic Sports (Niebles, Chen, and Fei-Fei 2010), HMDB51 (Kuehne et al. 2011) and UCF101 (Soomro, Zamir, and Shah 2012). The extensive results demonstrate the effectiveness of our method for large-scale video classification. Finally, we conduct detailed component analysis of our framework.

Experimental Setup

Datasets and Splits. The three popular datasets Olympic Sports, HMDB51 and UCF101 contain 783, 6766 and 13320 videos with 16, 51, and 101 categories, respectively. To compare our method with the state-of-the-arts, we follow the $50/50$ data splits proposed by (Xu, Hospedales, and Gong 2017), i.e., videos of 50% categories are used for model training and the other 50% categories are held unseen for testing. We adopt the 50 independent splits generated by (Xu, Hospedales, and Gong 2017) and report the average accuracy and standard deviation for experimental evaluation.

Zero-shot Settings. Typically, there are two zero-shot settings: inductive setting and transductive setting. The former assumes that only the labeled videos from the seen categories are available during training while the latter can use the unlabeled data of the unseen categories for model training. Specifically, for the transductive setting, we first choose the top 2000 frequent objects in all videos, then determine their relationships via the knowledge graph. In addition, compared to traditional zero-shot settings where the seen categories are absent at the test phase, the recently introduced generalized setting takes both seen and unseen videos as test data. Following (Xu, Hospedales, and Gong 2017; Song et al. 2018), we adopt the generalized setting in a transductive manner. In this setting, we follow (Song et al. 2018) to add an additional bias loss to alleviate the bias towards seen categories.

Table 1: ZSAR accuracies on the three benchmarks compared with state-of-the-art methods. Feature: Fisher Vectors (**FV**) or Bag of Words (**BoW**) or Object scores(**Ob**); Label Embedding: Attribute (**A**) or Word Embeddings(**W**); **ID**: Inductive setting; **TD**: Transductive setting. The average % accuracy ± standard deviation is reported. Note that some methods such as Objects2Action, ZSECOC, and UR adopt less than 50 splits for evaluation.

Method	Reference	Feature	Label Embedding	ID/TD	Olympic Sports	HMDB51	UCF101
DAP	CVPR2009	FV	A	ID	45.4 ± 12.8	N/A	15.9 ± 1.2
IAP	CVPR2009	FV	A	ID	42.3 ± 12.5	N/A	16.7 ± 1.1
HAA	CVPR2011	FV	A	ID	46.1 ± 12.4	N/A	14.9 ± 0.8
SVE	ICIP2015	BoW	W	ID	N/A	13.0 ± 2.7	10.9 ± ± 1.5
ESZSL	ICML2015	FV	W	ID	39.6 ± 9.6	18.5 ± 2.0	15.0 ± 1.3
SJE	CVPR2015	FV	W	ID	28.6 ± 4.9	13.3 ± 2.4	9.9 ± 1.4
SJE	CVPR2015	FV	A	ID	47.5 ± 14.8	N/A	12.0 ± 1.2
Objects2Action	ICCV2015	Ob	W	ID	N/A	15.6	30.3
MTE	ECCV2016	FV	W	ID	44.3 ± 8.1	19.7 ± 1.6	15.8 ± 1.3
ZSECOC	CVPR2017	FV	W	ID	**59.8 ± 5.6**	22.6 ± 1.2	15.1 ± 1.7
UR	CVPR2018	FV	W	ID	N/A	**24.4 ± 1.6**	17.5 ± 1.6
TS-GCN	Ours	Ob	W	ID	56.5 ± 6.6	23.2 ± 3.0	**34.2 ± 3.1**
SVE	ICIP2015	BoW	W	TD	51.4	22.7	18.7
UDA	ICCV2015	FV	A	TD	N/A	N/A	13.2 ± 1.9
UDA	ICCV2015	FV	A + W	TD	N/A	N/A	14.0 ± 1.8
MTE	ECCV2016	FV	W	TD	56.6 ± 7.7	24.8 ± 2.2	22.9 ± 3.3
UR	CVPR2018	FV	W	TD	N/A	28.9 ± 1.2	20.1 ± 1.4
TS-GCN	Ours	Ob	W	TD	**59.9 ± 5.3**	**31.0 ± 3.2**	**41.6 ± 3.7**

Table 2: Results on the generalized zero-shot setting.

Method	Olympic	HMDB51	UCF101
SJE	32.5±6.7	10.5±2.4	8.9±2.2
ConSE	37.6±9.9	15.4±2.8	12.7±2.2
GA	42.2±10.2	20.1±2.1	17.5±2.2
Objects2Action	N/A	N/A	30.3
Ours	**50.2 ± 6.8**	**21.9 ± 3.7**	**33.4 ± 3.4**

Comparison with State-of-the-Art Methods

Compared methods. We compare our method with state-of-the-art methods. (1) Direct/Indirect Attribute Prediction method (DAP, IAP) (Lampert, Nickisch, and Harmeling 2009). (2) Human Actions by Attributes (HAA) model (Liu, Kuipers, and Savarese 2011), which is implemented by (Xu, Hospedales, and Gong 2016). (3) Self-training method with SVM and semantic Embedding (SVE) (Xu, Hospedales, and Gong 2015). (4) Embarrassingly Simple Zero-Shot Learning (ESZSL) (Romera-Paredes and Torr 2015). (5) Structured Joint Embedding (SJE) (Akata et al. 2015). (6) Unsupervised Domain Adaptation (UDA) (Kodirov et al. 2015). (7) Multi-Task Embedding (MTE) (Xu, Hospedales, and Gong 2016). (8) Objects2Action (Jain et al. 2015), which also utilize objects as attributes for ZSAR. (9) Zero-Shot with Error-Correcting Output Codes (ZSECOC) (Qin et al. 2017). (10) Universal Representation (UR) model (Zhu et al. 2018) in inductive and transductive settings.

Inductive setting. The comparison results are illustrated in Table 1. Overall, our proposed method performs favorably against state-of-the-art methods. Compared with the recent methods MTE, ZSECOC, and UR, the proposed TS-GCN achieves an absolute gain of $(18.4\%, 19.1\%, 16.7\%)$ on UCF101 dataset. We also get comparable results on HMDB51 and Olympic Sports benchmark. Compared with another method, Objects2Action,which also adopts objects with semantic embeddings for ZSAR, the proposed TS-GCN outperforms it by $(3.9\%, 7.6\%)$ on UCF101 and HMDB51, respectively. Note that the results of TS-GCN and Objects2Action on HMDB51 are worse than those on UCF101 while other methods (e.g., MTE and UR) have the inverse results. This is because many action categories in HMDB51 are not sensitive to objects, such as the actions *run*, *walk*, *sit*, and *stand*. The results can be further improved by adding low-level or deep visual features in our framework. To verify this point, we design a baseline TS-GCN+FV, which concatenates the final feature vector generated from the instance branch with the FV feature of a video instance. By doing this, the classifiers are learned from both attribute feature (the feature used in our proposed TS-GCN method) and the visual feature (FV). Compared with the proposed TS-GCN method, TS-GCN+FV gets an absolute gain of 2.1% on UCF101 thanks to the additional visual features.

Transductive setting. Since this setting allows methods to access the unlabeled data of unseen categories, the results are better than those in the inductive setting, as shown in Table 1. In this setting, we first select the most frequent objects in all videos, which help us remove rare and unreliable objects in building knowledge graph. Among all the competitors, the proposed TS-GCN gets better or comparable results. Specifically, compared with MTE and UR, our method outperforms them by $(18.7\%, 6.2\%)$ and $(21.5\%, 2.1\%)$ on the UCF101 and HMDB51 benchmarks, respectively. We also perform the best on Olympic Sports.

Note that there are another two methods achieves top performance. The Cross-Domain UR (CD-UR) method (Zhu et al.

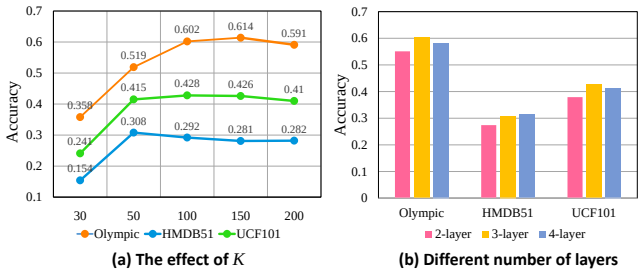

Figure 3: Comparison results among (a) different number of selected objects. (b) different number of GCN layers.

2018) adopts a large-scale dataset ActivityNet as auxiliary training data. CD-UR gets the mean accuracy of 42.5% on UCF101, while our TS-GCN achieves comparable results without using external training data. In addition, we outperform its non cross-domain version in both inductive and transductive setting as shown in Table 1. Compared with another recent method (Mettes and Snoek 2017) which designs spatial-aware object embedding for ZSAR and gets 40.4% accuracy on UCF101, our method outperforms it by 1.2% with transductive setting. Note that this method employs an object detector to leverage the spatial information of objects which can also improve the performance of our method.

Generalized ZSAR. We follow (Mishra et al. 2018) to use 20% data from the seen classes for testing and the remaining data for training. Since most methods do not report their results in this setting, we adopt other two methods for comparison: Convex Semantic Embeddings (ConSE) (Norouzi et al. 2013) and Generative Approach (GA) (Mishra et al. 2018). Table 2 presents the favorable performance of our TS-GCN method against the state-of-the-arts, which achieves the accuracy of $(50.2\%, 21.9\%, 33.4\%)$ on the three datasets. In addition, the recent spatial-aware object embedding method (Mettes and Snoek 2017) gets 32.8% accuracy on UCF101 dataset, which is lower than the proposed TS-GCN. The results clearly show that TS-GCN has remarkable generalization ability.

Further Remarks

To evaluate the effectiveness of our method, we perform an in-depth analysis in this section. The experiments are conducted in the transductive setting with 10 random splits.

How many objects should we use? The number of the selected objects per action category controls the size of the built knowledge graph. A larger K can make the built knowledge graph more comprehensive since it uses more objects to describe an action. However, too large K will result in high computational burden and may bring noisy to the knowledge graph. As shown in Figure 3 (a), a moderate value of K achieves the best performance. In addition, we find that UCF101 and Olympic Sports need more objects while HMDB51 achieves the best performance with $K = 50$. This is because some action categories in HMDB51 are not object-sensitive.

The deeper, the better? We explore the importance of the depth of GCN in our framework. The performance of using different numbers of layers on three benchmarks are shown in Figure 3 (b). The 2-layer model has the output channel numbers of 512 and 512. And we set the numbers as 2048, 1024, 1024, and 512 for the 4-layer model. Theoretically, the deeper network will enhance the message propagation between nodes thus improve the performance. However, we observe that adding more layers above the 3-layer model does not boost the accuracy of our framework significantly. One potential reason might be that the number of the training data is not large-scale (from $\sim$ 700 videos in Olympic Sports to $\sim 10,000$ videos in UCF101), which brings overfitting problem in the deeper networks.

Table 3: Comparison results of different baseline methods.

	Olympic	HMDB51	UCF101
TS-GCN-GoogleNews	60.5	29.3	40.5
TS-GCN-w/o SelfAttention	57.1	27.2	39.6
OS-GCN	40.9	17.1	25.2
TS-GCN	**59.2**	**30.8**	**42.8**

How important is temporal modeling? We consider the interaction among different video segments via a self-attention mechanism. To show its effectiveness, we design a baseline TS-GCN-w/o SelfAttention, which removes the self-attention layer and simply averages the object scores of all segments. Table 3 demonstrates that our proposed method consistently outperforms the baseline.

Different Word Embeddings. To analyze the sensitivity of our framework to different word embedding methods, we test the performance on another Word2Vec embedding method (Mikolov et al. 2013) trained on GoogleNews. Table 3 shows the comparison between the proposed method and the baseline TS-GCN-GoogleNews. Generally, the word embedding trained on YFCC100M (Thomee et al. 2016) achieves higher performance on two datasets. One potential reason might be that using visual metadata is more suitable for ZSAR tasks than training on Wikipedia or GoogleNews data (Jain et al. 2015). Moreover, the performance gap between both methods is not very significant, which shows that our framework is not sensitive to word embedding methods.

Is the Two-stream network redundant? We use the two-stream framework to jointly model the relationships between action-attribute, action-action, and attribute-attribute. To verify its necessity, we design a baseline One-Stream GCN (OS-GCN) which only uses one branch of GCN to model the relationships between all concepts. The input to OS-GCN is the word embedding of all action categories and the weighted attribute representation of each video instance calculated from Eq. (3). Table 3 shows the effectiveness of the two-stream design. The proposed TS-GCN outperforms OS-GCN by $(18.3\%, 13.7\%, 17.6\%)$ on the three benchmarks. Without the co-adaptation and co-operation between the two branches, it is difficult to explicitly learn meaningful classifiers and attribute features, which will result in training dilemma and performance degradation.

Visualization of the Learned Classifiers and Features. We perform t-SNE (Maaten and Hinton 2008) visualizations on

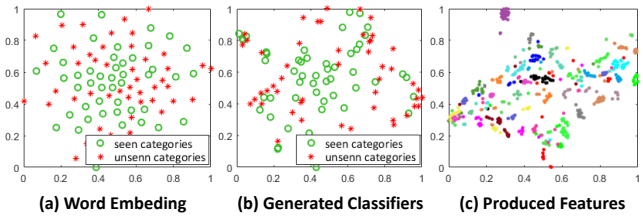

Figure 4: t-SNE visualizations for (a) word embedding of action categories. (b) classifiers generated by our TS-GCN method (from classifier branch). (c) object features (produced by the instance branch) of 500 randomly chosen videos from 20 unseen actions. Different categories are shown in different colors. The experiment is conducted on a random split of UCF101.

UCF101 experiments to show some characteristics of our proposed TS-GCN. Figure 4 (a) and (b) plot both the word embeddings and the classifiers (generated by the classifier branch) of the seen and unseen action categories. It can be seen that the word embeddings and the classifiers distributes very differently , which indicates the proposed method can leverage the relationships among concepts rather than only implicit word embeddings. Moreover, the seen and unseen classifiers are located more dispersedly than their word embeddings, which shows the good generalization ability of our TS-GCN. To verify the discriminative power of the learned object features from the instance branch, we randomly select 500 test videos from 25 unseen categories. Figure 4 (c) shows the distribution of the produced object features with different color. We can find that the learned features are effective since most samples in the same category are dispersed in a close region.

Conclusions

In this paper, we propose an end-to-end ZSAR framework with knowledge graphs to automatically generate classifiers for new categories. By designing a two-stream GCN model with a classifier branch and an instance branch, our method is able to effectively model the relationships between action-attribute, attribute-attribute, and action-action. In addition, a self-attention mechanism is adopted to model the temporal information across video segments. Comprehensive performance studies have been conducted by comparing our framework with state-of-the-art methods over three benchmark datasets. The effectiveness of our method is evidenced by its favorable performances compared with others. In the future, we will consider richer knowledge information in our framework such as the types of edges. We will also test other types of methods for modeling the knowledge information in ZSAR, such as GSNN and knowledge graph embedding. Besides, motivated by the favorable performance of our framework in video classification, we intend to apply this method to other related tasks, such as zero-shot event detection and multi-modality domain adaptation.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants 61432019, 61572498, 61532009, 61728210, 61721004, 61751211, 61772244, 61472379, 61720106006 and U1705262, and the Key Research Program of Frontier Sciences, CAS, Grant NO.QYZDJ-SSW-JSC039, the Beijing Natural Science Foundation 4172062.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*.

Akata, Z.; Reed, S.; Walter, D.; Lee, H.; and Schiele, B. 2015. Evaluation of output embeddings for fine-grained image classification. In *CVPR*.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*. Springer. 722–735.

Bond, F., and Foster, R. 2013. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1352–1362.

Fang, Y.; Kuan, K.; Lin, J.; Tan, C.; and Chandrasekhar, V. 2017. Object detection meets knowledge graphs. In *IJCAI*.

Gan, C.; Lin, M.; Yang, Y.; Zhuang, Y.; and Hauptmann, A. G. 2015. Exploring semantic interclass relationships (sir) for zero-shot action recognition. In *AAAI*.

Gan, C.; Lin, M.; Yang, Y.; de Melo, G.; and Hauptmann, A. G. 2016a. Concepts not alone: Exploring pairwise relationships for zero-shot video activity recognition. In *AAAI*.

Gan, C.; Yang, Y.; Zhu, L.; Zhao, D.; and Zhuang, Y. 2016b. Recognizing an action using its name: A knowledge-based approach. *International Journal of Computer Vision* 120(1):61–77.

Gan, C.; Yang, T.; and Gong, B. 2016. Learning attributes equals multi-source domain generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 87–97.

Gao, J.; Zhang, T.; Yang, X.; and Xu, C. 2017. Deep relative tracking. *IEEE Transactions on Image Processing* 26(4):1845–1858.

Gao, J.; Zhang, T.; Yang, X.; and Xu, C. 2018. P2t: Part-to-target tracking via deep regression learning. *IEEE Transactions on Image Processing* 27(6):3074–3086.

Gao, J.; Zhang, T.; and Xu, C. 2017. A unified personalized video recommendation via dynamic recurrent neural networks. In *ACM MM*, 127–135.

Gao, J.; Zhang, T.; and Xu, C. 2018. Watch, think and attend: End-to-end video classification via dynamic knowledge evolution modeling. In *2018 ACM Multimedia Conference on Multimedia Conference*, 690–699. ACM.

Han, J.; Zhang, D.; Cheng, G.; Liu, N.; and Xu, D. 2018. Advanced deep-learning techniques for salient and category-

specific object detection: a survey. *IEEE Signal Processing Magazine* 35(1):84–100.

Jain, M.; van Gemert, J. C.; Mensink, T.; and Snoek, C. G. 2015. Objects2action: Classifying and localizing actions without any video example. In *ICCV*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kodirov, E.; Xiang, T.; Fu, Z.; and Gong, S. 2015. Unsupervised domain adaptation for zero-shot learning. In *ICCV*.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *ICCV*.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*.

Lee, C.-W.; Fang, W.; Yeh, C.-K.; and Wang, Y.-C. F. 2018. Multi-label zero-shot learning with structured knowledge graphs. In *CVPR*.

Liu, J.; Kuipers, B.; and Savarese, S. 2011. Recognizing human actions by attributes. In *CVPR*.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.

Marino, K.; Salakhutdinov, R.; and Gupta, A. 2017. The more you know: Using knowledge graphs for image classification. In *CVPR*.

Mettes, P., and Snoek, C. G. 2017. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *ICCV*.

Mettes, P.; Koelma, D. C.; and Snoek, C. G. 2016. The imagenet shuffle: Reorganized pre-training for video event detection. In *ICMR*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mishra, A.; Verma, V. K.; Reddy, M.; Rai, P.; Mittal, A.; et al. 2018. A generative approach to zero-shot and few-shot action recognition. *arXiv preprint arXiv:1801.09086*.

Niebles, J. C.; Chen, C.-W.; and Fei-Fei, L. 2010. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*.

Norouzi, M.; Mikolov, T.; Bengio, S.; Singer, Y.; Shlens, J.; Frome, A.; Corrado, G. S.; and Dean, J. 2013. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*.

Qin, J.; Liu, L.; Shao, L.; Shen, F.; Ni, B.; Chen, J.; and Wang, Y. 2017. Zero-shot action recognition with error-correcting output codes. In *CVPR*.

Romera-Paredes, B., and Torr, P. 2015. An embarrassingly simple approach to zero-shot learning. In *ICML*.

Song, J.; Shen, C.; Yang, Y.; Liu, Y.; and Song, M. 2018. Transductive unbiased embedding for zero-shot learning. In *CVPR*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J. 2016. Yfcc100m: The new data in multimedia research. *Communications of the ACM* 59(2):64–73.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Wang, X.; Ye, Y.; and Gupta, A. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*.

Xu, X.; Hospedales, T.; and Gong, S. 2015. Semantic embedding space for zero-shot action recognition. In *IEEE International Conference on Image Processing (ICIP)*, 63–67. IEEE.

Xu, X.; Hospedales, T. M.; and Gong, S. 2016. Multi-task zero-shot action recognition with prioritised data augmentation. In *ECCV*.

Xu, X.; Hospedales, T.; and Gong, S. 2017. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision* 123(3):309–333.

Zhang, T.; Xu, C.; Zhu, G.; Liu, S.; and Lu, H. 2010. A generic framework for event detection in various video domains. In *MM*, 103–112.

Zhang, T.; Xu, C.; Zhu, G.; Liu, S.; and Lu, H. 2012a. A generic framework for video annotation via semi-supervised learning. *IEEE Transactions on Multimedia* 14(4):1206–1219.

Zhang, T.; Ghanem, B.; Liu, S.; and Ahuja, N. 2012b. Robust visual tracking via multi-task sparse learning. In *CVPR*.

Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2018. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*.

Zhang, T.; Xu, C.; and Yang, M.-H. 2017. Multi-task correlation particle filter for robust object tracking. In *CVPR*, 1–9.

Zhang, T.; Xu, C.; and Yang, M.-H. 2018a. Learning multi-task correlation particle filters for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, T.; Xu, C.; and Yang, M.-H. 2018b. Robust structural sparse tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zheng, Y.; Jeon, B.; Sun, L.; Zhang, J.; and Zhang, H. 2017. Student's t-hidden markov model for unsupervised learning using localized feature selection. *IEEE Transactions on Circuits and Systems for Video Technology*.

Zhu, Y.; Long, Y.; Guan, Y.; Newsam, S.; and Shao, L. 2018. Towards universal representation for unseen action recognition. In *CVPR*.