Objects2action: Classifying and localizing actions without any video example

Mihir Jain* Jan C. van Gemert*‡ Thomas Mensink* Cees G. M. Snoek*†
*University of Amsterdam ‡Delft University of Technology †Qualcomm Research Netherlands

Abstract

The goal of this paper is to recognize actions in video without the need for examples. Different from traditional zero-shot approaches we do not demand the design and specification of attribute classifiers and class-to-attribute mappings to allow for transfer from seen classes to unseen classes. Our key contribution is objects2action*, a semantic word embedding that is spanned by a skip-gram model of thousands of object categories. Action labels are assigned to an object encoding of unseen video based on a convex combination of action and object affinities. Our semantic embedding has three main characteristics to accommodate for the specifics of actions. First, we propose a mechanism to exploit multiple-word descriptions of actions and objects. Second, we incorporate the automated selection of the most responsive objects per action. And finally, we demonstrate how to extend our zero-shot approach to the spatio-temporal localization of actions in video. Experiments on four action datasets demonstrate the potential of our approach.*

1. Introduction

We aim for the recognition of actions such as *blow dry hair* and *swing baseball* in video without the need for examples. The common computer vision tactic in such a challenging setting is to predict the *zero-shot* test classes from disjunct train classes based on a (predefined) mutual relationship using class-to-attribute mappings [1, 8, 20, 32, 36]. Drawbacks of such approaches in the context of action recognition [24] are that attributes like 'torso twist' and 'look-down' are difficult to define and cumbersome to annotate. Moreover, current zero-shot approaches, be it for image categories or actions, assume that a large, and labeled, set of (action) train classes is available a priori to guide the knowledge transfer, but today's action recognition practice is limited to at most hundred classes [16, 19, 35, 42]. Different from existing work, we propose zero-shot learning for action classification that does not require tailored definitions and annotation of action attributes, and not a single video or action annotation as prior knowledge.

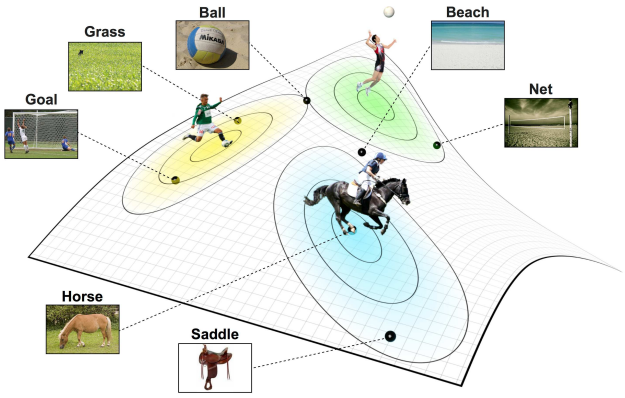

Figure 1. We propose *objects2action*, a semantic embedding to classify actions, such as *playing football*, *playing volleyball*, and *horse-riding*, in videos without using any video data or action annotations as prior knowledge. Instead it relies on commonly available textual descriptions, images and annotations of objects.

We are inspired by recent progress in *supervised* video recognition, where several works successfully demonstrated the benefit of representations derived from deep convolutional neural networks for recognition of actions [14, 17, 39] and events [40, 49]. As these nets are typically pre-trained on images and object annotations from ImageNet [5], and consequently their final layer represent object category scores, these works reveal that object scores are well-suited for video recognition. Moreover, since these objects have a lingual correspondence derived from nouns in WordNet, they are a natural fit for semantic word embeddings [6, 9, 26, 29, 41]. As prior knowledge for our zero-shot action recognition we consider a semantic word embedding spanned by a large number of object class labels and their images from ImageNet, see Figure 1.

Our key contribution is *objects2action*, a semantic embedding to classify actions in videos without using any video data or action annotations as prior knowledge. Instead it relies on commonly available object annotations, images and textual descriptions. Our semantic embedding has three main characteristics to accommodate for the specifics of actions. First, we propose a mechanism to exploit multiple-word descriptions of actions and ImageNet objects. Sec-

ond, we incorporate the automated selection of the most responsive objects per action. And finally, we demonstrate our zero-shot approach to action classification and spatio-temporal localization of actions.

Before going into detail, we will first connect our approach to related work on action recognition and zero-shot recognition.

2. Related work

2.1. Action Recognition

The action classification literature offers a mature repertoire of elegant and reliable methods with good accuracy. Many methods include sampling spatio-temporal descriptors [7, 47], aggregating the descriptors in a global video representation, such as versions of VLAD [12, 33] or Fisher Vectors [34] followed by supervised classification with an SVM. Inspired by the success of deep convolutional neural networks in image classification [18], several recent works have demonstrated the potential of learned video representations for action and event recognition [14, 17, 39, 40, 49]. All these deep representations are learned from thousands of object annotations [5], and consequently, their final output layer corresponds to object category responses indicating the promise of objects for action classification. We also use a deep convolutional neural network to represent our images and video as object category responses, but we do not use any action annotations nor any training videos.

Action classification techniques have recently been extended to action localization [2, 13, 21, 30, 46] where in addition to the class, the location of the action in the video is detected. To handle the huge search space that comes with such precise localization, methods to efficiently sample action proposals [13, 30, 46] are combined with the encodings and labeled examples used in action classification. In contrast, we focus on the zero-shot case where there is no labeled video data available for classification nor for localization. We are not aware of any other work on zero-shot action localization.

2.2. Zero-Shot Recognition

The paradigm of zero-shot recognition became popular with the seminal paper of Lampert *et al.* [20]. The idea is that images can be represented by a vector of classification scores from a set of *known classes*, and a semantic link can be created from the known class to a *novel class*. Existing zero-shot learning methods can be grouped based on the different ways of building these semantic links.

A semantic link is commonly obtained by a human provided class-to-attribute mapping [1, 8, 20, 32], where for each unseen class a description is given in terms of a set of attributes. Attributes should allow to tell classes apart, but should not be class specific, which makes finding good attributes and designing class-to-attribute mappings a non-trivial task. To overcome the need for human selection, at least partially, Rohrbach *et al.* evaluate external sources for defining the class-to-attribute mappings [36]. Typically, attributes are domain specific, *e.g.* class and scene properties [20] or general visual concepts [23] learned from images, or action classes [37] and action attributes [24] learned from videos. In our paper we exploit a diverse vocabulary of object classes for grounding unseen classes. Such a setup has successfully been used for action classification when action labels are available [14]. In contrast, we have a zero-shot setting and do not use any action nor video annotations.

Zero-shot video event recognition as evaluated in TRECVID [31] offers meta-data in the form of a an event kit containing the event name, a definition, and a precise description in terms of salient concepts. Such meta-data can cleverly be used for a class-to-attribute mapping based on multi-modal concepts [11, 48], seed a sequence of multimodal pseudo relevance feedback [15], or select relevant tags from Flickr [3]. In contrast to these works, we do not assume any availability of additional meta-data and only rely on the action name.

To generalize zero-shot classification beyond attribute-to-class mappings, Mensink *et al.* [25] explored various metrics to measure co-occurrence of visual concepts for establishing a semantic link between labels, and Froome *et al.* [9] and Norouzi *et al.* [29] exploit semantic word embeddings for this link. We opt for the latter direction, and also use a semantic word embedding [22, 26, 27] since this is the most flexible solution, and allows for exploiting object and action descriptions containing multiple words, such as the WordNet synonyms and the subject, verb and object triplets to describe actions [10, 43] used in this paper.

3. Objects2action

In zero-shot classification the train classes $\mathcal{Y}$ are different from the set of *zero-shot* test classes $\mathcal{Z}$, such that $\mathcal{Y} \cap \mathcal{Z} = \emptyset$. For training samples $\mathcal{X}$, a labeled dataset $\mathcal{D} \equiv \{\mathcal{X}, \mathcal{Y}\}$ is available, and the objective is to classify a test sample as belonging to one of the test classes $\mathcal{Z}$. Usually, test samples v are represented in terms of classification scores for all train classes $p_{vy}\, \forall y \in \mathcal{Y}$, and an affinity score g_{yz} is defined to relate these train classes to the test classes. Then the zero-shot prediction could be understood as a convex combination of known classifiers [1, 25, 29]:

$$\mathcal{C}(v) = \underset{z}{\operatorname{argmax}} \sum_{y} p_{vy}\, g_{yz}. \tag{1}$$

Often there is a clear relation between training classes $\mathcal{Y}$ and test classes $\mathcal{Z}$, for example based on class-to-attribute relations [1, 20] or all being nouns from the ImageNet/Wordnet hierarchy [9, 29]. It is unclear, however,

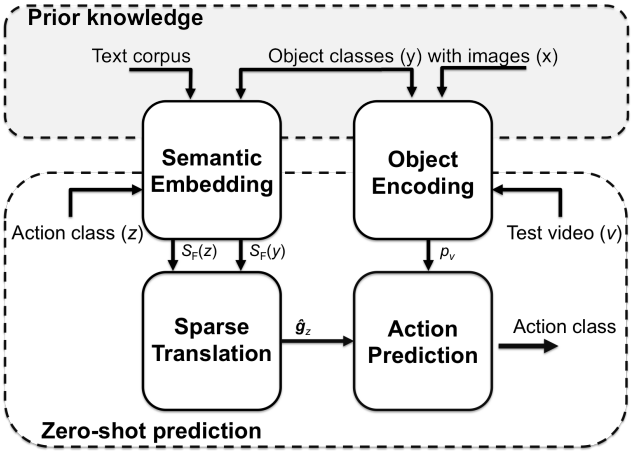

Figure 2. Dataflow in objects2action. Intermediate processes, data and corresponding symbols are specified in Section 3. Sparse tranlsation is only shown for action to object affinity. Note that we do not require any action class labeled visual examples nor any video examples as prior knowledge.

how to proceed when train classes and test classes are semantically disjoint.

Our setup, see Figure 2, differs in two aspects to the standard zero-shot classification pipeline: i) our zero-shot test examples are videos $\mathcal{V}$ to be classified in actions $\mathcal{Z}$, while we have a train set $\mathcal{D}$ with images $\mathcal{X}$ labeled with objects $\mathcal{Y}$ derived from ImageNet [5]. Therefore, we aim to *transfer* from the domain of images $\mathcal{X}$ to the domain of videos $\mathcal{V}$, and ii) we aim to *translate* objects semantics $\mathcal{Y}$ to the semantics of actions $\mathcal{Z}$.

Object encoding We encode a test video v by the classification scores to the $m = |Y|$ objects classes from the train set:

$$\boldsymbol{p}_v = [p(y_1|v), \ldots, p(y_m|v)]^T \tag{2}$$

where the probability of an object class is given by a deep convolutional neural network trained from ImageNet [18], as recently became popular in the video recognition literature [14, 17, 39, 40, 49]. For a video v the probability $p(y|v)$ is computed by averaging over the frame probabilities, where every 10^{th} frame is sampled. We exploit the semantics of in total 15,293 ImageNet object categories for which more than 100 examples are available.

We define the affinity between an object class y and action class z as:

$$g_{yz} = s(y)^T s(z), \tag{3}$$

where $s(\cdot)$ is a semantic embedding of any class $\mathcal{Z} \cup \mathcal{Y}$, and we use $\boldsymbol{g}_z = [s(y_1) \ldots s(y_m)]^T s(z)$ to represent the translation of action z in terms of objects $\mathcal{Y}$. The semantic embedding function s is further detailed below.

3.1. Semantic embedding via Gaussian mixtures

The objective for a semantic embedding is to find a d-dimensional space, in which the distance between an object $s(y)$ and an action $s(z)$ is small, if and only if their classes y and z are found in similar (textual) context. For this we employ the skip-gram model of word2vec [26,27] as semantic embedding function, which results in a look-up table for each word, corresponding to a d-dimensional vector.

Semantic word embeddings have been used for zero-shot object classification [29], but in our setting the key differences are i) that train and test classes come from different domains: objects in the train set and actions in the test set; and ii) both the objects and actions are described with a small description instead of a single word. In this section we describe two embedding techniques to exploit these multi-word descriptions to bridge the semantic gap between objects and actions.

Average Word Vectors (*AWV*) The first method to exploit multiple words is take the average vector of the embedded words [28]. The embedding $s(c)$ of a multi-words description c is given by:

$$s_{\mathrm{A}}(c) = \frac{1}{|w|} \sum_{w \in c} s(w). \tag{4}$$

This model combines words to form a single average word, as represented with a vector inside the word embedding. While effective, this cannot model any semantic relations that may exist between words. For example, the relations for the word *stroke*, in the sequence *stroke, swimming, water* is completely different than the word relations in the sequence *stroke, brain, ambulance*.

Fisher Word Vectors (*FWV*) To describe the precise meaning of distinct words we propose to aggregate the word embeddings using Fisher Vectors [38]. While these were originally designed for aggregating local image descriptors [38], they can be used for aggregating words as long as the *discrete* words are transformed into a *continuous* space [4]. In contrast to [4], where LSI is used to embed words into a continuous space, we employ the word embedding vectors of the skip-gram model. These vectors for each word are then analogous to local image descriptors and a class description is analogous to an image.

The advantage of the FWV model is that it uses an underlying generative model over the words. This generative model is modeling semantic topics within the word embedding. Where AWV models a single word, the FWV models a distribution over words. The *stroke* example could for example be assigned to two clear, distinct topics *infarct* and *swimming*. This word sense disambiguation leads to a more precise semantic grounding at the topic-level, as opposed to single word-level.

In the Fisher Vector, a document (*i.e.* a set of words) is described as the gradient of the log-likelihood of these

observations on an underlying probabilistic model. Following [38] we use a diagonal Gaussian Mixture Model with k components as probabilistic model, which we learn on approximately $45K$ word embedding vectors from the 15K object classes in ImageNet.

The Fisher Vectors with respect to the mean μ_k and variance σ_k of mixture component k are given by:

$$\mathcal{G}^c_{\mu_k} = \frac{1}{\sqrt{\pi_k}} \sum_{w \in c} \gamma_w(k) \left(\frac{s(w) - \mu_k}{\sigma_k} \right), \quad (5)$$

$$\mathcal{G}^c_{\sigma_k} = \frac{1}{\sqrt{2\pi_k}} \sum_{w \in c} \gamma_w(k) \left(\frac{(s(w) - \mu_k)^2}{\sigma_k^2} - 1 \right), \quad (6)$$

where π_k is the mixing weight, and $\gamma_w(k)$ denotes the responsibility of component k and we use the closed-form approximation of the Fisher information matrix of [38]. The final Fisher Word Vector is the concatenation of the Fisher Vectors (Eq. (5) and Eq. (6)) for all components:

$$s_{\mathrm{F}}(c) = [\mathcal{G}^c_{\mu_1}, \mathcal{G}^c_{\sigma_1}, \ldots, \mathcal{G}^c_{\mu_k}, \mathcal{G}^c_{\sigma_k}]^T. \quad (7)$$

3.2. Sparse translations

The action representation $\boldsymbol{g}_z$ represents the translation of the action to all objects in $\mathcal{Y}$. However, not all train objects are likely to contribute to a clear description of a specific action class. For example, consider the action class *kayaking*, it makes sense to translate this action to object classes such as *kayak*, *water*, and *sea*, with some related additional objects like *surf-board*, *raft*, and *peddle*. Likewise a similarity value with, *e.g.*, *dog* or *cat* is unlikely to be beneficial for a clear detection, since it introduces clutter. We consider two sparsity metrics that operate on the action classes or the test video.

Action sparsity We propose to sparsify the representation $\boldsymbol{g}_z$ by selecting the T_z most responsive object classes to a given action class z. Formally, we redefine the action to object affinity as:

$$\hat{\boldsymbol{g}}_z = [g_{zy_1}\delta(y_1, T_z), \ldots, g_{zy_m}\delta(y_m, T_z)]^T \quad (8)$$

where $\delta(y_i, T_z)$ is an indicator function, returning 1 if class y_i is among the top T_z classes. In the same spirit, the objects could also have been selected based on their distance, considering only objects within an ϵ_z distance from $s(z)$. We opt for the selection of the top T_z documents, since it is easier to define an a priori estimate of the value. Selecting T_z objects for an action class z means that we focus only on the object classes that are closer to the action classes in the semantic space.

Video sparsity Similarly, the video representation $\boldsymbol{p}_v$ is, by design, a dense vector, where each entry contains the (small) probability $p(y|v)$, of the presence of train class y in the video v. We follow [29] and use only the top T_v most prominent objects present in the video:

$$\hat{\boldsymbol{p}}_v = [p(y_1|v)\delta(y_1, T_v), \ldots, p(y_m|v)\delta(y_m, T_v)]^T \quad (9)$$

where $\delta(y_i, T_v)$ is an indicator function, returning 1 if class y_i is among the top T_v classes. Increasing the sparsity, by considering only the top T_v class predictions will reduce the effect of adding random noise by summing over a lot of unrelated classes with a low probability mass and is therefore likely to be beneficial for zero-shot classification.

The optimal values for both T_z and T_v are likely to depend on the datasets, the semantic representation and the specific action description. Therefore they are considered as hyper-parameters of the model. Typically, we would expect that $T \ll m$, *e.g.*, the 50 most responsive object classes will suffice for representing the video and finding the best action to object description.

3.3. Zero-shot action localization

Objects2action is easily extendable to zero-shot localization of actions by exploiting recent advances in sampling spatio-temporal tube-proposals from videos [13, 30, 46]. Such proposals have shown to give a high localization recall with a modest set of proposals.

From a test video, a set $\mathcal{U}$ of spatio-temporal tubes are sampled [13]. For each test video we simply select the maximum scoring tube proposal:

$$\mathcal{C}(v) = \underset{z \in \mathcal{Z}, u \in \mathcal{U}_v}{\operatorname{argmax}} \sum_y p_{uy}\, g_{yz}, \quad (10)$$

where u denotes a spatio-temporal tube proposal, and $\boldsymbol{p}_{uy}$ is the probability of the presence of object y in region u.

For the spatio-temporal localization, a tube proposal contains a series of frames, each with a bounding-box indicating the spatial localization of the action. We feed just the pixels inside the bounding-box to the convolutional neural network to obtain the visual representation embedded in object labels. We will demonstrate the localization ability in the experiments.

4. Experiments

In this section, we employ the proposed object2action model on four recent action classification datasets. We first describe these datasets and the text corpus used. Second, we analyze the impact of applying the Fisher Word Vector over the baseline of the Average Word Vector for computing the affinity between objects and actions, and we evaluate the action and video sparsity parameters. Third, we report zero-shot classification results on the four datasets, and we compare against the traditional zero-shot setting where actions are used during training. Finally, we report performance of zero-shot spatio-temporal action localization.

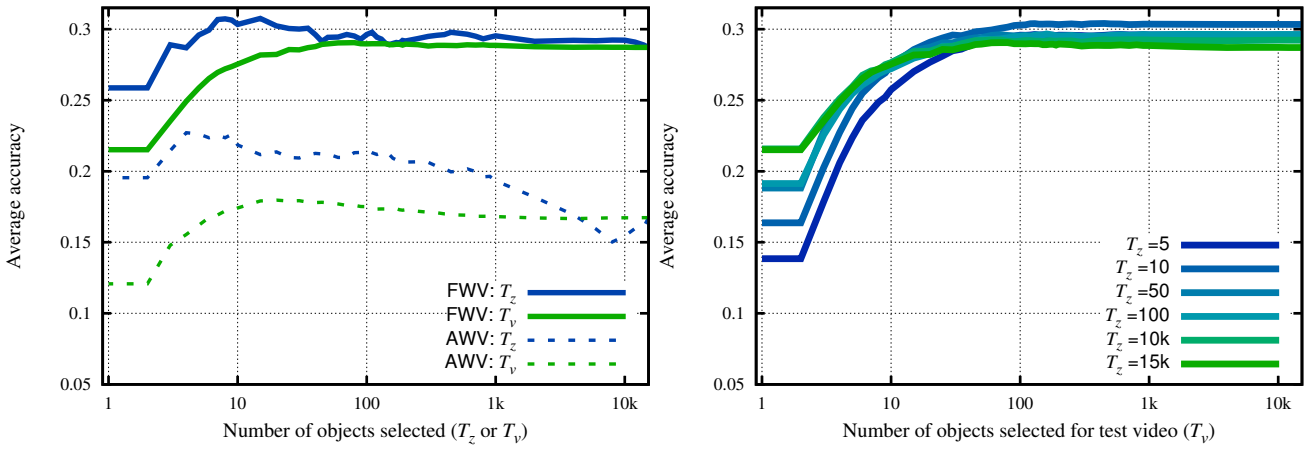

Figure 3. Impact of video T_v and action T_z sparsity parameters, individually (*left*) and when combined (*right*) on UCF101 dataset.

4.1. Prior knowledge and Datasets

Our method is based on freely available resources which we use as prior knowledge for zero-shot action recognition. For the four action classification datasets datasets used we only use the *test* set.

Prior knowledge We use two types of prior knowledge. First, we use deep convolutional neural network trained from ImageNet images with objects [18] as visual representation. Second, for the semantic embedding we train the skip-gram model of word2vec on the metadata (title, descriptions, and tags) of the YFCC100M dataset [44], this dataset contains about 100M Flickr images. Preliminary experiments showed that using visual metadata results in better performance than training on Wikipedia or GoogleNews data. We attribute this to the more visual descriptions used in the YFC100M dataset, yielding a semantic embedding representing visual language and relations.

UCF101 [42] This dataset contains 13,320 videos of 101 action classes. It has realistic action videos collected from YouTube and has large variations in camera motion, object appearance/scale, viewpoint, cluttered background, illumination conditions, *etc*. Evaluation is measured using average class accuracy, over the three provided test-splits with around 3,500 videos each.

THUMOS14 [16] This dataset has the same 101 action classes as in UCF101, but the videos are have a longer duration and are temporally unconstrained. We evaluate on the testset containing 1,574 videos, using mean average precision (mAP) as evaluation measure.

HMDB51 [19] This dataset contains 51 action classes and 6,766 video clips extracted from various sources, ranging from YouTube to movies, and hence this dataset contains realistic actions. Evaluation is measured using average class accuracy, over the three provided test-splits with each 30 videos per class (1,530 videos per split).

UCF Sports [35] This dataset contains 150 videos of 10 action classes. The videos are from sports broadcasts capturing sport actions in dynamic and cluttered environments. Bounding box annotations are provided and this dataset is often used for spatio-temporal action localization. For evaluation we use the test split provided by [21] and performance is measured by average class accuracy.

Embedding	Sparsity	Best accuracy	Accuracy at T_z=10, T_v=100
AWV	Video	18.0%	17.5%
	Action	22.7%	21.9%
	Combine	22.7%	21.6%
FWV	Video	29.1%	29.0%
	Action	30.8%	30.3%
	Combine	30.8%	30.3%

Table 1. Evaluating AWV and FWV for object to class affinity, and comparing action and video sparsity on UCF101 dataset.

4.2. Properties of Objects2action

Semantic embedding We compare the AWV with the FWV as semantic embedding. For the FWV, we did a run of preliminary experiments to find suitable parameters for the number of components (varying $k = \{1, 2, 4, 8, 16, 32\}$), the partial derivatives used (weight, mean, and/or variance) and whether to use PCA or not. We found them all to perform rather similar in terms of classification accuracy. Considering a label has only a few words (1 to 4), we therefore fix $k = 2$, apply PCA to reduce dimensionality by a factor of 2, and to use only the partial derivatives w.r.t. the mean (conforming the results in [4]). Hence, the total dimensionality of FWV is d, equivalent to the dimensionality of AWV, which allows for a fair comparison. The two embeddings are compared in Table 1 and Figure 3 (*left*), and FWV clearly outperforms AWV in all the cases.

Sparsity parameters In Figure 3, we evaluate the action sparsity and video sparsity parameters. The left plot shows average accuracy versus T_z and T_v. It is evident that

Embedding	Sparsity	UCF101	HMDB51	THUMOS14	UCF Sports
AWV	None	16.7%	8.0%	4.4%	13.9%
	Video	17.5%	7.7%	10.7%	13.9%
	Action	21.9%	9.9%	19.9%	25.6%
FWV	None	28.7%	14.2%	25.9%	23.1%
	Video	29.0%	14.5%	27.8%	23.1%
	Action	**30.3%**	**15.6%**	**33.4%**	**26.4%**
Supervised		63.9%	35.1%	56.3%	60.7%

Table 2. Evaluating semantic embeddings, action and video spartsity: Average accuracies (mAP for THUMOS14) for the four datasets. Action sparsity and FWV both boost the performance consistently. Supervised upper-bound using object scores as representation.

action sparsity, *i.e.*, selecting most responsive object classes for a given action class leads to a better performance than video sparsity. The video sparsity (green lines) is more stable throughout and achieves best results in the range of 10 to 100 objects. Action sparsity is a bit sinuous, nevertheless it always performs better, independent of the type of embedding. Action sparsity is at its best in the range of selecting the 5 to 30 most related object classes. For the remaining experiments, we fix these parameters as $T_z = 10$ and $T_v = 100$.

We also consider the case when we apply sparsity on both video and actions (see the right plot). Applying sparsity on both sides does not improve performance, it is equivalent to the best action sparsity setting, showing that selecting the most prominent objects per action suffice for zero-shot action classification. Table 1 summarise the accuracies for the best and fixed choices of T_z and T_v.

4.3. Zero-shot action classification

In this section we employ the obtained parameters of Object2action, from the previous section, for zero-shot action classification on the test splits of all four datasets. We evaluate the benefit of using the FWV over AWV, and the effect of using sparsity (video sparsity, action sparsity or no sparsity at all). The results are provided in Table 2. We observe that the FWV always outperforms AWV, and that it is always beneficial to apply sparsity, and action sparsity with FWV performs the best. We also provide the supervised upper-bounds using the same video representation of object classification scores in Table 2. Here and for all the experiments, we power normalize ($\alpha = 0.5$) the video representations before applying ℓ_2 normalization.

Comparison to few-shot supervised learning In this experiment we compare the zero-shot classifier against few-shot supervised learning, on the THUMOS14 dataset. For this we consider two types of video representation. The first representations, uses the state-of-the-art motion representation of [47], by encoding robust MBH descriptors along the improved trajectories [47] using Fisher Vectors. We follow the standard parameter cycle, by applying PCA, using a GMM with $K = 256$ Gaussians, employing power and ℓ_2 normalization. The second representation uses the object scores $\boldsymbol{p}_v$ of a video, here also we apply power and ℓ_2 normalization. For both representations, we train one-vs-rest linear SVM classifiers and we average performance over 20 runs for every given number of train examples.

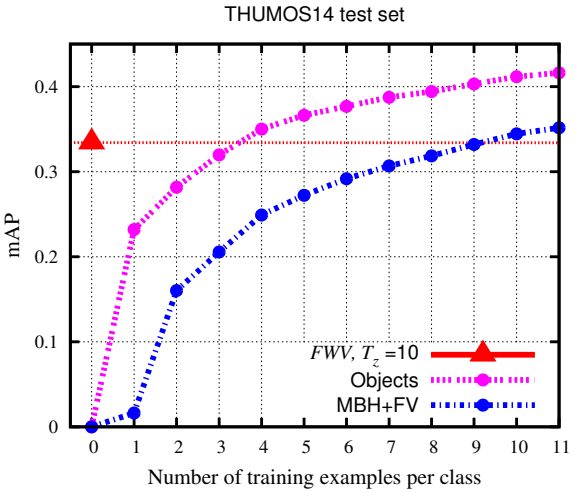

Figure 4. Our approach compared with the supervised classification with few examples per class: State-of-the-art object and motion representations respectively require 4 and 10 examples per class to catch up with our approach, which uses no example.

The results in mAP are shown in Figure 4. Interestingly, to perform better than our zero-shot classification, fully supervised classification setup requires 4 and 10 samples per class for object and motion representations respectively.

Object transfer versus action transfer We now experiment with the more conventional setup for zero-shot learning, where we have training data for some action classes, disjoint from the set of test action classes. We keep half of the classes of a given dataset as train labels and the other half as our zero-shot classes. The action classifiers are learned from odd (or even) numbered classes and videos from the even (or odd) numbered classes are tested.

We evaluate two types of approaches for action transfer, *i.e.*, when training classes are also actions. The first method uses the provided action attributes for zero-shot classification with direct attribute prediction [20]. Since attributes are available only for UCF101, we experiment on this dataset.

Method	Train	Test	UCF101	HMDB51
Action attributes	Even	Odd	16.2%	—
	Odd	Even	14.6%	—
Action labels	Even	Odd	16.1%	12.4%
	Odd	Even	14.4%	13.4%
Objects2action	ImageNet	Odd	**37.3%**	**15.0%**
		Even	**38.9%**	**24.5%**

Table 3. Object transfer versus action transfer in a conventional zero-shot set-up. Direct attribute prediction [20] is used with action attributes, FWV is used to embed action labels, and in our objects2action.

The train videos of the training classes are used to learn linear SVMs for the provided 115 attributes. The second method uses action labels embedded by FWV to compute affinity between train and test action labels. We use the same GMM with $k = 2$ components learned on ImageNet object labels. Here linear SVMs are learned for the training action classes. The results are reported for UCF101 and HMDB51 datasets. For both the above approaches for action transfer, we use MBH descriptors encoded by Fisher vectors for video representation. The results are reported in Table 3.

For comparison with our approach, the same setup of testing on odd or even numbered classes is repeated with the object labels. The training set is ImageNet objects, so no video example is used for training. Table 3 compares object transfer and action transfer for zero-shot classification. Object transfer leads to much better learning compared to both the methods for action transfer. The main reason for the inferior performance using actions is that there are not enough action labels or action attributes to describe the test classes, whereas from 15k objects there is a good chance to find a few related object classes.

Zero-shot event retrieval We further demonstrate our method on the related problem of zero-shot event retrieval. We evaluate on the TRECVID13 MED [31] testset for EK0 task. There are 20 event classes and about 27,000 videos in the testset. Instead of using the manually specified event kit containing the event name, a definition, and a precise description in terms of salient attributes, we only rely on the class label. In Table 4, we report mAP using event labels embedded by AWV and FWV. We also compare with the state-of-the-art approaches of Chen *et al.* [3] and Wu *et al.* [48] reporting their settings that are most similar to ours. They learn concept classifiers from images (from Flickr, Google) or YouTube video thumbnails, be it that they also use the complete event kit description. Using only the event labels, both of our semantic embeddings outperform these methods.

Free-text action search As a final illustration we show in Figure 6 qualitative results from free-text querying of action videos from the THUMOS14 testset. We used

Method		mAP
Wu *et al.* [48] (Google images)		1.21%
Chen *et al.* [3] (Flickr images)		2.40%
Wu *et al.* [48] (YouTube thumbnails)		3.48%
Objects2action	AWV	3.49%
	FWV	**4.21%**

Table 4. Zero-shot event retrieval on TRECVID13 MED testset: Comparison with the state-of-the-art methods having similar zero-shot setup as ours. Inspite of using only event labels and images, we outperform methods that use event description and video thumbnails.

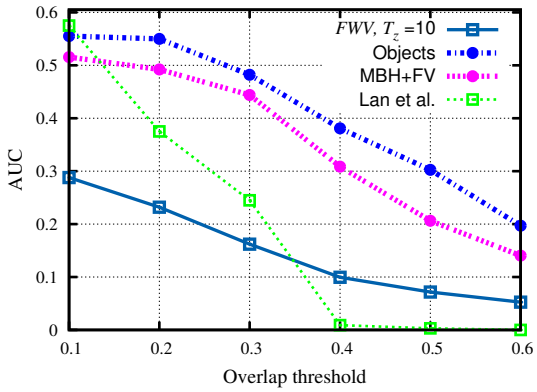

Figure 5. Action localization without video example on UCF Sports: AUCs for different overlap thresholds are shown for $T_z = 10$ and also for the fully supervised setting with motion and object representations. The performance is promising considering no example videos are used.

the whole dataset for querying, and searched for actions that are not contained in the 101 classes of THUMOS14. Results show that free-text querying offers a tool to explore a large collection of videos. Results are best when the query is close to one or a few existing action classes, for example "Dancing" retrieves results from "salsa-spin" and other dancing clips. Our method fails for the query "hit wicket", although it does find cricket matches. Zero shot action recognition through an object embedding unlocks free text querying without using any kind of expensive video annotations.

4.4. Zero-shot action localization

In our final experiment, we aim to localize actions in videos, *i.e.*, detect when and where an action of interest occurs. We evaluate on the UCF Sports dataset, following the latest convention to localize an action spatio-temporally as a sequence of bounding boxes [13,21,45]. For sampling the action proposal, we use the tubelets from [13] and compute object responses for each tubelet of a given video. We compare with the fully supervised localization using the object and motion representations described in Section 4.3. The top five detections are considered for each video after non-maximum suppression.

Figure 6. Illustration of never seen actions on THUMOS14 testset: For a given textual action query the top five retrieved videos are shown. The 101 classes of THUMOS14 do not contain these five action label queries. The first two queries are somewhat close to classes '*Sumo wrestling*' and '*Salsa spin*'. All retrieved videos for the query: 'Fight in ring' include sumo wrestling. The videos retrieved for the second query: 'Dancing' also includes two instances of dancing other than salsa. All results for the these two queries are technically correct. The third query 'Martial arts' finds mostly gymnasts, and a karate match. The fourth query is: 'Smelling food', where we still obtain cakes, food items and dining table in the background. For the fifth query: 'hit wicket' (in cricket) we do not succeed but retrieve some cricket videos. This illustration shows the potential for free keyword querying of action classes without using any examples.

The three are compared in Figure 5, which plots area under the ROC (AUC) for varying overlap thresholds. We also show the results of another supervised method of Lan *et al.* [21]. It is interesting to see that for higher thresholds our approach performs better. Considering that we do not use any training example it is an encouraging result. There are other state-of-the-art methods [13, 14, 45] not shown in the figure to avoid clutter. These methods achieve performance comparable to or lesser than our supervised case.

For certain action classes many objects and scene from the context might not be present in the groundtruth tubelets. Still our approach finds enough object classes for recognizing the zero-shot classes in the tubelets, as we have large number of train classes. In contrast, finding atomic parts of actions such as 'look-up', 'sit-down', 'lift-leg' etc are difficult to collect or annotate. This is one of the most critical advantages we have with objects, that it is easier to find many object or scene categories.

5. Conclusion

We presented a method for zero shot action recognition without using any video examples. Expensive video annotations are completely avoided by using abundantly available object images and labels and a freely available text corpus to relate actions into an object embedding. In addition, we showed that modeling a distribution over embedded words with the Fisher Vector is beneficial to obtain a more precise sense of the unseen action class topic, as compared to a word embedding based on simple averaging. We explored sparsity both in the object embedding, as well as in the unseen action class, showing that sparsity is beneficial over mere feature-dimensionality.

We validate our approach on four action datasets and achieve promising results for action classification and localization. We also demonstrate our approach for action and event retrieval on THUMOS14 and TRECVID13 MED respectively. The most surprising aspect of our objects2action is that it can potentially find any action in video, without ever having seen the action before.

Acknowledgments This research is supported by the STW STORY project and the Dutch national program COMMIT.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013. 1, 2

[2] L. Cao, Z. Liu, and T. S. Huang. Cross-dataset action detection. In *CVPR*, 2010. 2

[3] J. Chen, Y. Cui, G. Ye, D. Liu, and S.-F. Chang. Event-driven semantic concept discovery by exploiting weakly tagged internet images. In *ICMR*, 2014. 2, 7

[4] S. Clinchant and F. Perronnin. Textual similarity with a bag-of-embedded-words model. In *ICTIR*, 2013. 3, 5

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1, 2, 3

[6] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, 2013. 1

[7] I. Everts, J. C. van Gemert, and T. Gevers. Evaluation of color spatio-temporal interest points for human action recognition. *TIP*, 23(4):1569–1580, 2014. 2

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 2

[9] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 1, 2

[10] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013. 2

[11] A. Habibian, T. Mensink, and C. Snoek. Composite concept discovery for zero-shot video event detection. In *ICMR*, 2014. 2

[12] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013. 2

[13] M. Jain, J. van Gemert, H. Jégou, P. Bouthemy, and C. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014. 2, 4, 7, 8

[14] M. Jain, J. van Gemert, and C. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 2015. 1, 2, 3, 8

[15] L. Jiang, T. Mitamura, S. Yu, and A. Hauptmann. Zero-example event search using multimodal pseudo relevance feedback. In *ICMR*, 2014. 2

[16] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 1, 5

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1, 2, 3

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 3, 5

[19] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *ICCV*, 2011. 1, 5

[20] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1, 2, 6, 7

[21] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011. 2, 5, 7, 8

[22] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014. 2

[23] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei. Object bank: An object-level image representation for high-level visual recognition. *IJCV*, 107(1):20–39, 2014. 2

[24] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, 2011. 1, 2

[25] T. Mensink, E. Gavves, and C. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014. 2

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 1, 2, 3

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 2, 3

[28] D. Milajevs, D. Kartsaklis, M. Sadrzadeh, and M. Purver. Evaluating neural word representations in tensor-based compositional settings. In *EMNLP*, 2014. 3

[29] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014. 1, 2, 3, 4

[30] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014. 2, 4

[31] P. Over, G. Awad, J. Fiscus, and G. Sanders. Trecvid 2013–an introduction to the goals, tasks, data, evaluation mechanisms, and metrics. In *TRECVID Workshop*, 2013. 2, 7

[32] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 1, 2

[33] X. Peng, L. Wang, Y. Qiao, and Q. Peng. Boosting vlad with supervised dictionary learning and high-order statistics. In *ECCV*, 2014. 2

[34] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, 2014. 2

[35] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 1, 5

[36] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, 2011. 1, 2

[37] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012. 2

[38] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 2013. 3, 4

[39] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2, 3

[40] C. Snoek et al. MediaMill at TRECVID 2013: Searching concepts, objects, instances and events in video. In *TRECVID*, 2013. 1, 2, 3

[41] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013. 1

[42] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012. 1, 5

[43] C. Sun and R. Nevatia. Semantic aware video transcription using random forest classifiers. In *ECCV*, 2014. 2

[44] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015. 5

[45] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013. 7, 8

[46] J. van Gemert, M. Jain, E. Gati, and C. Snoek. APT: Action localization proposals from dense trajectories. In *BMVC*, 2015. 2, 4

[47] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2, 6

[48] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, and P. Natarajan. Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *CVPR*, 2014. 2, 7

[49] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative CNN video representation for event detection. In *CVPR*, 2015. 1, 2, 3