Webly-Supervised Video Recognition by Mutually Voting for Relevant Web Images and Web Video Frames

Chuang Gan[1(✉)], Chen Sun[2], Lixin Duan[3], and Boqing Gong[4]

[1] IIIS, Tsinghua University, Beijing, China
ganchuang1990@gmail.com
[2] Google Research, Mountain View, USA
[3] Amazon, Seattle, USA
[4] CRCV, University of Central Florida, Orlando, USA

Abstract. Video recognition usually requires a large amount of training samples, which are expensive to be collected. An alternative and cheap solution is to draw from the large-scale images and videos from the Web. With modern search engines, the top ranked images or videos are usually highly correlated to the query, implying the potential to harvest the labeling-free Web images and videos for video recognition. However, there are two key difficulties that prevent us from using the Web data directly. First, they are typically noisy and may be from a completely different domain from that of users' interest (e.g. cartoons). Second, Web videos are usually untrimmed and very lengthy, where some query-relevant frames are often hidden in between the irrelevant ones. A question thus naturally arises: to what extent can such noisy Web images and videos be utilized for labeling-free video recognition? In this paper, we propose a novel approach to mutually voting for relevant Web images and video frames, where two forces are balanced, i.e. aggressive matching and passive video frame selection. We validate our approach on three large-scale video recognition datasets.

1 Introduction

This paper aims to classify actions and events in user-captured videos without human labeling. The ubiquity of smart phones and surveillance cameras has created videos far surpassing what we can watch. Instead of "eyeballing" the videos for potential useful information, it is desirable to develop automatic video analysis and understanding algorithms. Video recognition in the wild is a very challenging task: videos from the same categories could vary greatly in lighting conditions, video resolutions, camera movements, etc. Meanwhile, those from different categories could be inherently similar (*e.g.* "apply eye makeup" and "apply lipstick"). To recognize actions and events, one commonly adopted framework is encoding hand-crafted features (*e.g.* improved dense trajectories [41]) into video-level representations with Fisher vectors [37]. There are also recent

B. Leibe et al. (Eds.): ECCV 2016, Part III, LNCS 9907, pp. 849–866, 2016.
DOI: 10.1007/978-3-319-46487-9_52

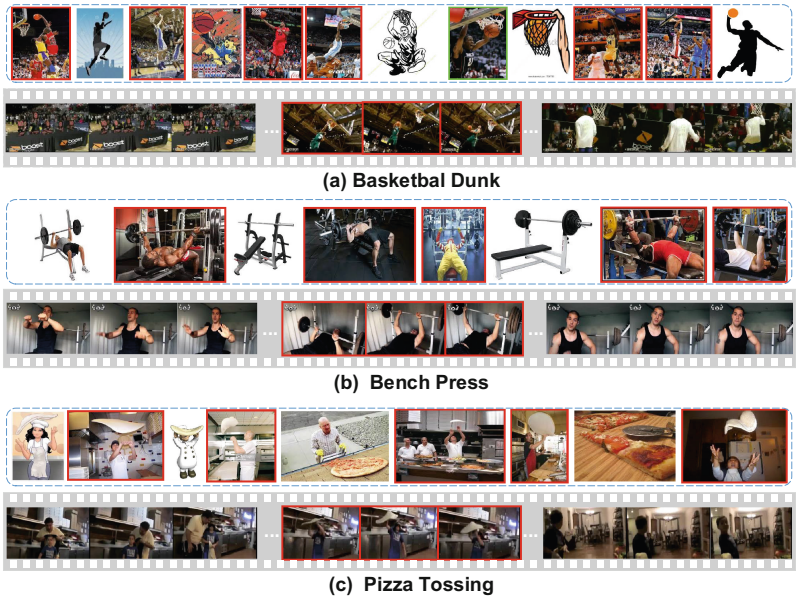

Fig. 1. To utilize Web images and videos for video classification, our key observation is that the query-relevant images and frames typically appear in both domains with similar appearances, while the irrelevant images and videos have their own distinctiveness. Here we show Web images (top) and video frames (bottom) retrieved by keywords basketball dunk, bench press and pizza tossing from search engines. The relevant ones are marked in red. (Color figure online)

approaches based on deep convolutional neural networks [11,20,29,40] or recurrent networks [16,34]. All these approaches require and implicitly assume the existence of large-scale labeled training data.

Manually labeling large amount of video examples is time-consuming and difficult to scale up. On the other hand, there are abundant image and video examples on the Web that can be easily retrieved by querying action or event names from image/video search engines. These two observations motivate us to focus on Webly-supervised video recognition by exploiting Web images and Web videos. Using video frames in addition to images not only adds more diverse examples for training better appearance models, but also allows us to train better temporal models, as found in [12,38].

However, there are two key difficulties that prevent us from using Web data directly. First, the images and videos retrieved from Web search engines are typically noisy. They may contain irrelevant results, or relevant results from a completely different domain than users' interest (*e.g.* cartoons or closeup shots of objects). To make the problem worse, Web videos are usually untrimmed and could be several minutes to hours long. Even for a correctly tagged video, the majority of its frames could be irrelevant to the actual action or event. Our goal

then becomes to identify query-relevant images and video frames from the Web data which are both noisily and weakly labeled, in order to train good machine learning models for action and event classification.

Our proposed method is based on the following observation: *the relevant images and video frames typically exhibit similar appearances, while the irrelevant images and videos have their own distinctiveness.* In Fig. 1, we show the Web images (top) and video frames (bottom) retrieved by keywords basketball dunk, bench press and pizza tossing. We can see that for the basketball dunk example, non-slam-dunk frames in the video are mostly about a basketball game. The irrelevant Web images are more likely to be cartoons. Similar observation also holds for bench press and pizza tossing, where the irrelevant images include cartoons and product shots. This observation indicates that selecting training examples from Web images and videos can be made easier, if they could be mutually filtered to keep those in common!

Our algorithm to mutually filtering Web images and video frames goes as follows: we first jointly choose images and video frames and try to match them *aggressively*. A good match between the subset of images and the subset of video frames occurs when both subsets are relevant to the action name, since "each irrelevant image or frame is irrelevant in its own way". We then impose a *passive* constraint over the video frames to be selected, such that they are collectively not too far from the original videos. We would like to be passive on the videos, in contrast to the images, because our ultimate goal is for video action recognition. Otherwise, the aggressive matching mechanism may end up with too few frames and causes a domain adaptation problem [28] between the training set and test videos. Once the Web images and video frames are selected for the actions or events of interest, they can be readily used to train action or event classifiers with a wide range of tools. Some examples include SVM, CNN and LSTM.

The remaining sections are organized as follows. Section 2 describes related work on video recognition, learning from the Web data and domain adaptation. Section 3 presents our approach to automatically selecting relevant examples from crawled images and videos to be used for Webly-supervised video recognition. Section 4 reports empirical results, followed by discussion and conclusion in Sect. 5.

2 Related Work

We discuss some related works to ours, including those on video recognition, learning from weakly-labeled Web data, and domain adaptation.

2.1 Video Recognition

Video recognition has been widely explored in Computer Vision and Multimedia communities. A survey can be found in [19]. Most of previous works use hand-designed features to extract motion and appearance information for video representation. So far, improved dense trajectories (iDT) [41] and its variants [22,42]

show state-of-the-art performance on video recognition when combined with Fisher vector coding [25].

Motivated by the success of convolutional neural networks on image recognition tasks [21,30,39,45], there are also several attempts to apply deep learning techniques for video recognition. Karpathy *et al.* [20] compare several architectures for action recognition. Tran *et al.* [40] propose to learn generic spatial-temporal features with 3D convolutional filters. Simonyan and Zisserman [29] propose a two-stream architecture to capture both spatial and motion information with a pixel stream and an optical flow stream respectively. Wang *et al.* [43] further improve the results by using deeper neural networks. Instead of learning representation using video data, recent works [47] for complex event recognition have shown CNN features from models pre-trained from ImageNet [5] achieve promising results. More recently, Recurrent Neural Networks (RNNs) are shown effective to model temporal information in videos. Srivastava *et al.* [34] propose an LSTM encoder-decoder framework to learn video representations in an unsupervised manner [34]. Donahue *et al.* [7] train a two-layer LSTM network for action classification. Ng *et al.* [24] further demonstrate that a deeper LSTM network can further improve the performance. All these approaches require high-quality labeled training data. It remains unclear whether they can also obtain reasonable video recognition results using noisy Web data.

2.2 Learning from Weakly-Labeled Web Data

Web data is inherently noisy. To handle this problem, the NEIL system [4] iteratively refines its model using the discovered object relationships. LEVAN [6] clusters visual concepts into groups, and rejects those with low visual consistency. Chen and Gupta [3] propose a semi-supervised approach to learning CNN parameters with easier examples first and more complex examples later. Sun *et al.* [36] and Zhang *et al.* [48] propose to use multi-modal data to learn visual concepts. In the video domain, Duan *et al.* [9] describe a system that uses large amount of weakly labeled Web videos for visual event recognition with transfer learning techniques. Habibian *et al.* [15] obtain textual descriptions of videos from the Web and learn an embedding for few-example event recognition. Nevertheless, these approaches all require humans to annotate a few positive videos as "seeds". Sun *et al.* [38] and Gan *et al.* [12] propose domain transfer approaches from weakly-labeled Web images for action localization and event recognition tasks, where each video is guaranteed to contain relevant snippets. In contrast, our approach screens all the downloaded web videos (of a query) simultaneously and does not impose the assumption of existing relevant frames over any individual video. To alleviate the tedious human burden and achieve Webly-supervised action recognition, several researchers have attempted to learn video concept detectors by crawling images and videos [2,14,31,46] after querying the action/event name as well as associated queries. However, the quality of the obtained data is lower compared with the fully-supervised set, as the retrieved examples are not only noisy but also without spatiotemporal localization. The noisy and weak supervision is likely to confuse the training of video classifiers.

Recently, the studies in [35] and soon followed by [44] propose solutions to train deep convolutional networks (CNNs) when there exist mislabeled images in the training set; the idea is to introduce a label noise layer placed at the top of CNNs. This paper is different in that we focus on how to remove the noisy data before actually training any classifiers. Our work can thus benefit most generic classifiers in addition to CNNs.

2.3 Domain Adaptation

In order to mutually vote for video frames and images that are relevant to the action/event, we use maximum mean discrepancy (MMD) [17,33] to match them. MMD has been widely used in domain adaptation [27], e.g., for feature representation learning [26], data instance re-weighting [17], landmark selection [13], and classifier regularization [8]. Moreover, when it goes to the technical algorithm, our formulation shares some spirit with the work [13] on landmark selection. However, we emphasize that the goal of our work is not for domain adaptation at all; neither images nor videos we retrieved are our target domain for testing. Instead, we tackle Webly-supervised video recognition by learning classifiers from both relevant Web images and video frames.

3 Proposed Approach

In this section, we present the details of our approach to jointly selecting video frames and images from the Web data, for the purpose of Webly-supervised video recognition. Our algorithm is built upon the motivating observation that "all relevant images and frames to an action name are alike; each irrelevant image or frame is irrelevant in its own way." We firstly give the overall formulation, and then describe an alternative optimization procedure for solving the problem.

3.1 Joint Selection of Action/event Relevant Web Video Frames and Web Images

For the ease of presentation, we first define the following notations. For each class (of an action or event), we denote by $\mathcal{I} = \{\mathbf{x}_m\}_{m=1}^{\mathsf{M}}$ the set of Web images, and by $\mathcal{V} = \{\mathbf{v}_n\}_{n=1}^{\mathsf{N}}$ the set of video frames, both returned by some search engines in response to the query of the class name. The Web data are quite noisy; there are both relevant items and outliers for the class. In order to filter out the relevant items, we introduce M indicator variables $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{\mathsf{M}}]^\top$, where $\alpha_m \in \{0, 1\}$ for each image $\mathbf{x}_m$, and N indicator variables $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{\mathsf{N}}]^\top$, where $\beta_n \in \{0, 1\}$ for each video frame $\mathbf{v}_n$. If $\alpha_m = 1$ (resp., $\beta_n = 1$), the corresponding image $\mathbf{x}_m$ (resp., video frame $\mathbf{v}_n$) will be identified as a relevant item to the class.

Aggressive Matching. If we conduct a pairwise comparison between a subset of the images $\mathcal{I}$ with a subset of the video frames $\mathcal{V}$, any class-irrelevant images or frames would decrease the similarity between the two subsets, because the irrelevant items are likely different from each other and also different from the relevant items. Therefore, we can let the images and video frames mutually vote for class-relevant items, by matching all possible pairwise subsets of them, respectively. Such a pair can be expressed by $(\{\alpha_m \mathbf{x}_m\}_{m=1}^{\mathsf{M}}, \{\beta_n \mathbf{v}_n\}_{n=1}^{\mathsf{N}})$. The pairs with high matching scores have lower chance of containing irrelevant images or video frames.

Because of the simplicity and effectiveness of the maximum mean discrepancy (MMD) criterion [17], we adopt it in this work to measure the degree of matching between any images and frames $(\{\alpha_m \mathbf{x}_m\}_{m=1}^{\mathsf{M}}, \{\beta_n \mathbf{v}_n\}_{n=1}^{\mathsf{N}})$. We propose to minimize the square of MMD such that the true negative images and video frames are expected to be *filtered out* (i.e., the corresponding α_m's or β_n's will tend to be zeros). In other words, the remaining images and video frames are expected to be the true positive items for the class. Formally, we formulate the following optimization problem:

$$\min_{\alpha_m, \beta_n \in \{0,1\}} \left\| \frac{1}{\sum_{m=1}^{\mathsf{M}} \alpha_m} \sum_{m=1}^{\mathsf{M}} \alpha_m \phi(\mathbf{x}_m) - \frac{1}{\sum_{n=1}^{\mathsf{N}} \beta_n} \sum_{n=1}^{\mathsf{N}} \beta_n \phi(\mathbf{v}_n) \right\|_{\mathcal{H}}^2, \tag{1}$$

where $\phi(\cdot)$ is a mapping function which maps a feature vector from its original space into a Reproducing Kernel Hilbert Space $\mathcal{H}$.

The above is an integer programming problem, which is very computationally expensive to solve. Following [13], we relax Eq. (1) by introducing $\hat{\alpha}_m = \frac{\alpha_m}{\sum_{m=1}^{\mathsf{M}} \alpha_m}$ and $\hat{\beta}_n = \frac{\beta_n}{\sum_{n=1}^{\mathsf{N}} \beta_n}$. Then, we arrive at the following optimization problem:

$$\min_{\hat{\boldsymbol{\alpha}} \in [0,1]^{\mathsf{M}}, \hat{\boldsymbol{\beta}} \in [0,1]^{\mathsf{N}}} \left(\hat{\boldsymbol{\alpha}}^\top, \hat{\boldsymbol{\beta}}^\top \right) \begin{pmatrix} K_I & -K_{IV} \\ -K_{VI}^\top & K_V \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\alpha}} \\ \hat{\boldsymbol{\beta}} \end{pmatrix}, \tag{2}$$

where $\hat{\boldsymbol{\alpha}} = [\alpha_1, \ldots, \alpha_{\mathsf{M}}]^\top$, $\hat{\boldsymbol{\beta}} = [\beta_1, \ldots, \beta_{\mathsf{N}}]^\top$, $K_I \in \mathbb{R}^{\mathsf{M} \times \mathsf{M}}$ and $K_V \in \mathbb{R}^{\mathsf{N} \times \mathsf{N}}$ are the kernel matrices computed over the images and video frames respectively, and $K_{VI}^\top = K_{IV} \in \mathbb{R}^{\mathsf{M} \times \mathsf{N}}$ denotes the kernel matrix computed between the images and video frames, respectively. We use a Gaussian RBF kernel in our experiments.

Passive Video Frame Selection. Note that Eq. (1) matches a subset of images with a subset of video frames very aggressively. While there could be many pairs of subsets whose images and frames are all relevant to the class, Eq. (1) only choose the one with the best matching (in terms of the MMD measure). This strategy is effective in removing true negative images and frames. However, it may also abandon many relevant ones in order to reach the best matching. We thus introduce a passive term to balance the aggressive matching.

Since our eventual task is video recognition, we propose to impose a passive regularization over the selected video frames, such that they are collectively not too far from the original videos:

$$\min_{\hat{\boldsymbol{\beta}}\in[0,1]^{\mathsf{M}},W} \quad \left\|V - V\cdot diag(\hat{\boldsymbol{\beta}})\cdot W\right\|_F^2, \tag{3}$$

where $V = [\mathbf{v}_1, \ldots, \mathbf{v}_{\mathsf{N}}]$, and the variable W is a linear transformation matrix which linearly reconstructs V from all the selected video frames, i.e., $V\cdot diag(\hat{\boldsymbol{\beta}})$. In order to have a low reconstruction error, one cannot keep too few video frames selected by the variables $\boldsymbol{\beta}$. On the other hand, it is fine to remove redundant frames from the candidate set $\mathcal{V}$. Our experiments show that removing the redundant frames incurs little loss on the overall performance, and even improves the performance of an LSTM-based classifier.

Combining Eqs. (2) and (3), we present our overall optimization problem as follows:

$$\min_{\substack{\hat{\boldsymbol{\alpha}}\in[0,1]^{\mathsf{M}},\\ \hat{\boldsymbol{\beta}}\in[0,1]^{\mathsf{N}},W}} \left(\hat{\boldsymbol{\alpha}}^\top, \hat{\boldsymbol{\beta}}^\top\right)\begin{pmatrix} K_I & -K_{IV} \\ -K_{IV}^\top & K_V \end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{\alpha}}\\ \hat{\boldsymbol{\beta}}\end{pmatrix} + \lambda\|V - V\cdot diag(\hat{\boldsymbol{\beta}})\cdot W\|_F^2, \tag{4}$$

where $\lambda > 0$ is a pre-defined tradeoff parameter to balance these two terms.

3.2 Optimization

To solve the optimization problem in Eq. (4), we develop a procedure to alternatively update $\{\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}\}$ and W until the value of the objective function in Eq. (4) converges.

Updating W: When we fix $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$, Eq. (4) reduces to

$$\min_{W} \ \|V - V\cdot diag(\hat{\boldsymbol{\beta}})\cdot W\|_F^2, \tag{5}$$

whose closed-form solution can be derived to update W:

$$W_{\text{new}} = \left((V\cdot diag(\hat{\boldsymbol{\beta}}))^\top (V\cdot diag(\hat{\boldsymbol{\beta}})\right)^\dagger (V\cdot diag(\hat{\boldsymbol{\beta}}))^\top, \tag{6}$$

where $\dagger$ denotes the pseudo-inverse of a matrix.

Updating $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$: We then fix W and solve for $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$. We first re-write Eq. (3) as:

$$\min_{\hat{\boldsymbol{\beta}}\in[0,1]^{\mathsf{N}}} \sum_{n,n'} \hat{\beta}_n\hat{\beta}_{n'} \underbrace{V_{:n}^\top V_{:n'} W_{n':} W_{:n}^\top}_{A_{nn'}} - 2\sum_n \hat{\beta}_n \underbrace{(V_{:n}^\top V W_{n:}^\top)}_{b_n}, \tag{7}$$

where $V_{:n}$ and $W_{:n}$ represent the n^{th} columns of V and W respectively, and $V_{n:}$ and $W_{n:}$ denote the n^{th} rows of V and W respectively. For simplicity, we define $A_{nn'} = V_{:n}^T V_{:n'} W_{n':} W_{:n}^\top$ and $b_n = V_{:n}^\top W_{n:}^\top$.

Substituting Eqs. (7) to (4), we arrive at the following:

$$\min_{\hat{\boldsymbol{\alpha}}\in[0,1]^{\mathsf{M}},\hat{\boldsymbol{\beta}}\in[0,1]^{\mathsf{N}}} \left(\hat{\boldsymbol{\alpha}}^\top, \hat{\boldsymbol{\beta}}^\top\right)\begin{pmatrix} K_I & -K_{IV} \\ -K_{VI}^\top & K_V + \lambda A \end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{\alpha}}\\ \hat{\boldsymbol{\beta}}\end{pmatrix} - 2\lambda\left(\hat{\boldsymbol{\alpha}}^\top, \hat{\boldsymbol{\beta}}^\top\right)\begin{pmatrix}\mathbf{0}\\ b\end{pmatrix}, \tag{8}$$

which can be efficiently solved by using off-the-shelf quadratic programming solvers.

3.3 Harvesting a Labeling-Free Training Set

After solving the optimization problem in Eq. (4), we have two ranking lists of the Web images and Web video frames, respectively, according to the values of $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ at the last iteration of our alternative optimization procedure. We can thus keep some percentage of the top ranked images and videos as our labeling-free training set for video recognition. In our experiments, we examine different percentages from 95 % to 10 %. We will also test the effectiveness of this labeling-free training set for different classifiers, including SVM, fine-tuned deep neural networks, and an LSTM-based classifier.

4 Experiments

In this section, we evaluate the quality of our labeling-free training set under two fundamental tasks in video recognition: action recognition and event detection. We also contrast our algorithm to some competing baselines, and compare our results with those in the recent works on Webly-supervised video recognition.

4.1 Datasets

In the experiments, we collect our training data by downloading Web images and videos from the popular search engines with text queries. Specifically, given an action/event class name as the search query, we download about 600 top-ranked images from Google and 20 videos from YouTube. Duplicated images are removed by comparing color histogram features. To comply with the query format of Google image search, all occurrences of *without*, *non-* and *not* are replaced with the minus sign. For the downloaded Youtube videos, we limit the length of each video to be less than fifteen minutes for both memory and computational concerns. Most videos have the frame rate of 30 FPS.

For the test sets, we consider three well-labeled large-scale datasets.

UCF101 [32]. This is a large video dataset for action recognition collected from YouTube. It consists of 101 action classes, 13K clips, and 27 hours of video data. The task is generally considered challenging since many videos are captured under poor lighting, with cluttered background, or severe camera motion. As our framework requires no manually labeled training set, we only use the three provided test-splits to test and evaluate our framework. Each test-split has around 3,800 videos. The averaged classification accuracy over the three splits is used as the evaluation metric.

TRECVID MED 2013 [1] **and 2014**[2]. They are the two largest publicly available video datasets for high-level event detection, and are introduced by NIST for participants in the TRECVID competition. MED 2013 contains 20 events (E006 – E015 and E021 – E030), while MED 2014 has 20 events (E021 – E040).

[1] http://nist.gov/itl/iad/mig/med13.cfm.
[2] http://nist.gov/itl/iad/mig/med14.cfm.

Each dataset has three different partitions: *Background*, *100EX* and *MEDTest*. *Background* contains about 5000 background videos not belonging to any of the considered events; *100EX* contains 100 positive videos for each event that is used as the training set in TRECVID; *MEDTest* contains around 25,000 videos (over 960 hours of videos), with per-video ground truth annotations for 20 event categories. We evaluate our approach on *MEDTest* and apply the official average precision (AP) metric used in TRECVID contests.

Data Pre-processing. For both training and testing, the crawled videos from Web and testing videos in *MEDTest* are decomposed into a set of frames. Using all video frames would be computationally expensive and is not necessary, as there are lots of redundancy among the frames. Thus, we only use the key frames. To extract these, we start by detecting shot boundaries by calculating color histograms for all frames. For each frame, we then calculate the L_1 distance between the previous color histogram and the current one. If the distance is larger than a threshold (we set it as 0.2 in this paper), this frame is marked as a shot boundary. After detecting the shots, we define the key frames each as the one in the middle of a shot. By doing this, we extract around 150 key frames for a 5 min video.

Since some videos of the three large-scale datasets are also collected from Web, we check whether our crawled videos unintentionally include any videos in the testing set. Specifically, we extract the fc6 features using VGGNet19 for each key frame in our collected videos and the videos in testing set. Then we compute the pairwise distances. We find that there are no overlapped videos.

4.2 Action Classification Experiment

Experiment Setup. Here we use the UCF101 dataset for evaluation. Our framework automatically harvests a labeling-free training set. A high-quality training set is supposed to be able to produce all kinds of good action classifiers. We thus examine three types of classifiers in our experiments:

- **CNNs** [21]: CNNs pre-trained from ImageNet have been proven to generalize well to action recognition tasks with domain-specific fine-tuning [29]. We choose the VGGNet19 [30] released by Oxford to conduct experiments. To fine-tune the VGGNet19, we use the Caffe [18] toolbox, take selected Web images/video frames as inputs, and set the width of the last fully-connected layer and the softmax layer as the number of action categories. We initialize the network with pre-trained weights, except for the last fully-connected layer which is randomly initialized. Each key frame/image is resized with the shorter side to be 256 pixels which is compatible with the input requirement of VGGNet19. During training, all data are randomly shuffled, and organized as mini-batches with the size of 128 for VGGNet19 fine-tuning using stochastic gradient descend. The learning rate starts from 10^{-4} and decreases to 10^{-5} after 20K iterations, then to 10^{-6} after 40K iterations. The training is stopped after 50K iterations. For testing, to predict an action label for a video, we average the corresponding prediction scores of all the key frames of the video.

Table 1. Webly-supervised action recognition results on UCF101, by fine-tuning VGGNet19 using **both** Web images and Web video frames. (x%: percentage abandoned)

Method	# Number of training data	Acc (%)
All crawled data	426K	64.7
Validation	368K	66.5
One-class SVM (5 %)	405K	65.4
One-class SVM (10 %)	384k	65.9
One-class SVM (15 %)	363k	65.9
Unsupervised one-class SVM (5 %)	405K	66.6
Unsupervised one-class SVM (10 %)	384k	66.9
Unsupervised one-class SVM (15 %)	363k	66.4
Landmarks (5 %)	405K	67.9
Landmarks (10 %)	384k	68.3
Landmarks (15 %)	363k	67.7
Ours (5 %)	405K	**68.7**
Ours (10 %)	384k	**69.3**
Ours (15 %)	363k	**68.9**

- **LSTM** [16]: We feed the selected video frames into an LSTM with softmax classifier. We use the LSTM implemented by Caffe [18], and set the rolling time k as 25 and the number of hidden state as 256. The LSTM weights are learnt by using the BPTT algorithm. During training, we set the size of mini-batch as 10. And the learning rate starts from 10^{-3} and decreases to 10^{-4} after 50K iterations. The training is stopped after 80K iterations. For testing, the LSTM classifier directly gives a video-level prediction.
- **SVM**: we extract the fc6 features of pre-trained VGGNet19 for images or video frames, and train a multi-class SVM classifier using the LibLinear toolbox [10] by fixing soft margin cost as 1. Similarly to the CNN classifier, we use late fusion (average) of the frame-level scores to generate the video-level predictions.

Table 2. Webly-supervised action recognition results on UCF101, by fine-tuning VGGNet19 using **either** Web images or Web video frames. (x%: percentage abandoned)

Data	Method										
	All crawled data	Validation	One-class SVM			Landmarks			Ours		
			5 %	10 %	15 %	5 %	10 %	15 %	5 %	10 %	15 %
Images	61.2	61.7	61.2	62.1	61.7	63.9	64.1	64.3	64.7	64.9	**65.1**
Videos	57.6	58.1	58.2	58.4	**58.6**	58.0	58.2	58.1	58.2	58.3	58.5

For testing on the UCF101 dataset, we uniformly sample 25 frames per video as suggested in [29], and then utilize a CNN/LSTM/SVM classifier to make predictions.

Baseline Methods. To evaluate our framework, we compare against several state of the art noise removal approaches as baselines:

- **Validation:** For each action class, we split the crawled data U into K equal and disjoint subsets. Each subset is scored by a binary SVM classifier trained on the rest $K-1$ subsets as positive and some random images of the other classes as negative. Every data point in U is predicted once. Negative-scored data are considered as noise and rejected. We use the implementation of LibSVM [1] with default hyper parameter $\lambda = 1$ to conduct experiments. In our experiment, we set K as 5.
- **One-class SVM:** We use LibSVM [1] to conduct the experiment.
- **Unsupervised one-class SVM:** We implemented Liu *et al.*' method [23] ourselves and followed the suggested details for tuning the hyper-parameters (*e.g.* using Gaussian kernels, soft labels and the number of neighbors).
- **Landmarks**: The concept of landmarks [13] is originally defined as a subset of data point from source domain that match the target domain. In our problem, we first treat Web images as the source domain (and Web video frames as

Table 3. Webly-supervised action recognition results on UCF101, by training a LSTM classifier using top 25 frames for each video.

Method	Acc (%)
Random	56.3
Validation	63.8
One-class SVM	64.6
Landmarks	64.2
Ours	**65.1**

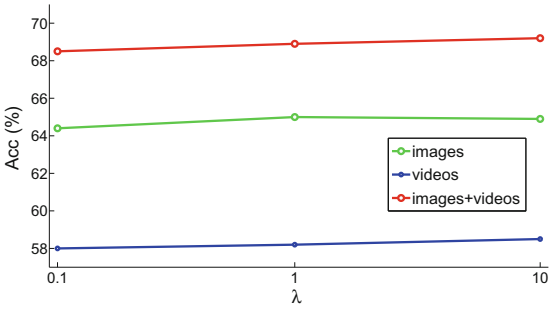

Fig. 2. Action recognition accuracies (Acc %) w.r.t the parameter λ on UCF101.

the target domain) to select "landmark" Web images. Then we reverse the source and target domains to select video frames. We use the code provided by authors for the experiments.

Results on UCF101. Table 1 reports the Webly-supervised action recognition results when our and the baseline approaches are used to select both Web images and video frames for fine-tuning CNNs. For one-class SVM, landmarks, and our own approach, we need to define the amount of data to be rejected. For fair comparison, we report the performances when rejecting 5 %, 10 %, and 15 % Web data for all the methods. To be noted, we take both Web images and video frames together as input for Validation and One-class SVM, and then keep the relevant training samples. For landmark and our own approach, we use Gaussian RBF kernels and fix the bandwidth parameter as 1 in all experiments. We solve the quadratic programming problem by using a Gurobi solver (http://www.gurobi.com). For our own approach, we fixed $\lambda = 10$ in Eq. (1) for all experiments, and we examine its effect when λ is set to 0.1, 1, and 10 with reject ratio of 10 % in Fig. 2. We also experiment with either of Web images or Web video frames for fine-tuning VGGNet19 in Table 2, with the top-ranked 25 frames per video for the LSTM classifier in Table 3, and with different percentages of Web images for the SVM classifier in Table 4.

Table 4. Webly-supervised action recognition results on UCF101, by a SVM classifier using only Web images. (x%: percentage abandoned).

Data	Method									
	All crawled data	Validation	One-class SVM			Landmarks			Ours	
			5 %	10 %	15 %	5 %	10 %	15 %	5 %	10 %	15 %
Images	53.2	54.1	54.2	54.9	54.6	55.1	55.4	55.6	56.0	56.4	**56.6**

From Table 1, we have two key observations. (1) The action recognition performance could be improved if some noisy data are removed by using one-class SVM, validation, landmarks, unsupervised one-class SVM and our proposed approach. These results validate the necessity of our study. (2) Our proposed approach to jointly selecting relevant images and video frames is more effective than the competing baselines, such as one-class SVM, unsupervised one-class SVM and Validation. In Table 2, we can find that the proposed framework can consistently achieve better results when using images only for fine-tuning CNN, with different ratios to reject the noise data. Under video frames only, the improvement of our approach compared with others is marginal, but ours still achieves better result compared with directly using the crawled data. Results in Table 4 shows that our approach is also a good companion for traditional SVM based classifications.

Table 5. Webly-supervised action recognition results on UCF101, by fine-tuning VGGNet19 using only 50 % and 10 % video frames (50 % and 90 % abandoned).

Method	# Number of training data	Acc (%)
All crawled data	360K	57.6
One-class SVM (50 %)	180K	55.7
One-class SVM (90 %)	36K	49.8
Landmarks (50 %)	180K	54.9
Landmarks (90 %)	36K	52.1
Ours (50 %)	180k	**58.8**
Ours (90 %)	36k	**58.2**

The Effectiveness of Removing Redundant Frames. In addition to removing noisy or outlier images and video frames, our approach also reduces redundant frames. The results in Table 3 show that frames selected by our proposed framework can achieve better performance than other approaches. We speculate that LSTM needs diverse sample to model the internal relationship in a sequence, and repetitively redundant frames would cripple its modeling capabilities. This requirement is a good match to our formulation.

Moreover, the redundant frames provide little extra information for the other classifiers either. To further evaluate whether our proposed framework can reduce the amount of training data to reach reasonable action classification performances when fine-tuning VGGNet19, we further conduct experiment by rejecting 50 % and 90 % frames during training. Experiment results are shown in Table 5. Surprisingly, the performance of our proposed approach has not dropped much from Table 2, even slightly better when rejecting 50 % video frames. However, one-class SVM and landmark-based approach decrease significantly. These results validate the effectiveness of our approach to reducing redundant video frames. The remaining video frames can maintain most of discriminative information and enjoy a lower computation cost.

Comparisons with State of the Arts that Use Fully Labeled Data. In Table 6, we add comparisons with the state-of-the-art results that are obtained by training classifiers from fully labeled training data. We directly quote the numbers from the published papers. Among the selected systems, LRCN [7], LSTM composite model [34], spatial stream network [29], and Karpathy et al. [20] are based on pure appearance features from static images. IDT+FV [41], C3D [40] include motion features from videos as well. We find that the performance of our Webly-supervsied approach is comparable to the spatial networks which use positive videos, but still has gaps when compared with motion features.

Table 6. Comparisons with state of the arts results using fully labeled data on UCF101.

Method	Acc (%)
LRCN [7]	71.1
LSTM composite model [34]	75.8
IDT + FV [41]	87.9
C3D [40]	82.3
Karpathy et al. [20]	65.4
Spatial stream network [29]	73.0
Ours (spatial)	69.3

4.3 Webly-Supervised Multimedia Event Detection

In order to have a better understanding of our approach, we also apply it to the large-scale TRECVID MED 2013 and 2014 datasets. There have been some systems on the MED tasks which learn event detectors from the Web data. While we only use the class names to download Web images and videos, the existing systems often employ additional queries like event related concepts. We contrast our work to the following: (1) Concept Discovery [2], (2) Bi-Concept [14], (3) Composite Concepts [14], (4) EventNet [46], and (5) Selected Concepts [31]. Approach (1) uses Web images to train event detectors, (2) – (4) use Web videos to train event detectors, and (5) firstly trains concept detectors using Web images, uses them to rank testing videos, and then re-trains event detectors with the top-ranked testing videos. We note that the strategy of (5) can be readily added as a post-processing component to other methods as well.

Table 7. Comparisons with other state-of-the-art zero-shot/webly-supervised event detection systems on MEDTest 2013.

Method	mAP (%)
Concept Discovery [2]	2.3
Bi-concept [14]	6.0
Composite Concept [14]	6.4
EventNet [46]	8.9
Selecting [31]	11.8
Ours	**16.1**

For a fair comparison, we report our results on MEDTest 2013 and directly compare them with state-of-the-art results quoted from original papers. The results in Table 7 show that our framework outperforms the other systems by a large margin. For additional analysis, we also provide per-event-class results in

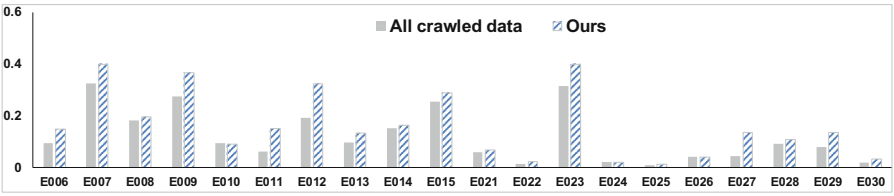

Fig. 3. Per-event detection result compared with All crawled data on MEDTest 2013 dataset.

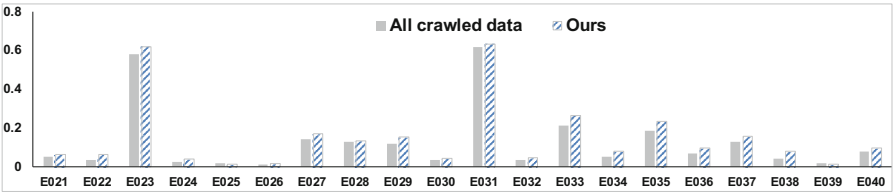

Fig. 4. Per-event detection result compared with All crawled data on MEDTest 2014 dataset.

Figs. 3 and 4, respectively on MEDTest 2013 and MEDTest 2014. The numbers are reported of using both all crawled data and our selected data (reject ratio 10 %) to fine-tune VGGNet19. We observe performance gains for 17 out of 20 classes on MEDTest 2013 and 18 of 20 classes on MEDTest 2014, verifying the effectiveness of our approach to removing noisy data from the Web images and Web video frames.

Implementation Details of Fine-Tuning. We use the Caffe [18] toolbox for fine-tuning CNNs, with a VGGNet19 model [30] that is pre-trained on ImageNet [5] by the authors. The learning rate starts from 10^{-4} and decreases to 10^{-5} after 25K iterations, then to 10^{-6} after 50K iterations. The training is stopped after 65K iterations. For testing, to predict an event label for a video, we average the corresponding prediction scores of all the key frames of the video. Momentum and weight decay coefficients are again set to 0.9 and 0.0005. All layers are fine-tuned, except the last fully-connected layer, which has to be changed to produce an output of event classes.

5 Conclusions

In this paper, we investigated to what extent the Web images and Web videos could be leveraged to conduct Webly-supervised video recognition. To distill useful data from the noisy Web ones, we proposed a unified approach to jointly removing irrelevant Web images and (also redundant) video frames. We developed an efficient alternative optimization procedure to solve our proposed formulation. Extensive experiments, for both action recognition and event detection, validate that our framework not only outperforms competing baselines, but

also beats existing systems which also exploit Web data for event detection. We expect this work to benefit future research on large-scale video recognition tasks.

Acknowledgments. This work was supported in part by NSF IIS-1566511. Chuang Gan was partially supported by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003.

References

1. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (2011)
2. Chen, J., Cui, Y., Ye, G., Liu, D., Chang, S.: Event-driven semantic concept discovery by exploiting weakly tagged internet images. In: ICMR (2014)
3. Chen, X., Gupta, A.: Webly supervised learning of convolutional networks. In: ICCV (2015)
4. Chen, X., Shrivastava, A., Gupta, A.: NEIL: extracting visual knowledge from web data. In: ICCV, pp. 1409–1416 (2013)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: CVPR (2009)
6. Divvala, S.K., Farhadi, A., Guestrin, C.: Learning everything about anything: webly-supervised visual concept learning. In: CVPR, pp. 3270–3277 (2014)
7. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
8. Duan, L., Tsang, I.W., Xu, D., Chua, T.S.: Domain adaptation from multiple sources via auxiliary classifiers. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 289–296. ACM (2009)
9. Duan, L., Xu, D., Tsang, I.H., Luo, J.: Visual event recognition in videos by learning from web data. IEEE Trans. Pattern Anal. Mach. Intell. **34**(9), 1667–1680 (2012)
10. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
11. Gan, C., Wang, N., Yang, Y., Yeung, D.Y., Hauptmann, A.G.: Devnet: a deep event network for multimedia event detection and evidence recounting. In: CVPR, pp. 2568–2577 (2015)
12. Gan, C., Yao, T., Yang, K., Yang, Y., Mei, T.: You lead, we exceed: labor-free video concept learning by jointly exploiting web videos and images. In: CVPR (2016)
13. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: discriminatively learning domain-invariant features for unsupervised domain adaptation. In: ICML, pp. 222–230 (2013)
14. Habibian, A., Mensink, T., Snoek, C.G.: Composite concept discovery for zero-shot video event detection. In: ICMR, p. 17 (2014)
15. Habibian, A., Mensink, T., Snoek, C.G.: Videostory: a new multimedia embedding for few-example recognition and translation of events. In: ACM Multimedia, pp. 17–26 (2014)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)

17. Huang, J., Gretton, A., Borgwardt, K.M., Schölkopf, B., Smola, A.J.: Correcting sample selection bias by unlabeled data. In: NIPS, pp. 601–608 (2006)
18. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: ACM Multimedia, vol. 2, p. 4 (2014)
19. Jiang, Y.G., Bhattacharya, S., Chang, S.F., Shah, M.: High-level event recognition in unconstrained videos. Int. J. Multimedia Inf. Retrieval **2**(2), 73–101 (2013)
20. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
22. Lan, Z., Lin, M., Li, X., Hauptmann, A.G., Raj, B.: Beyond gaussian pyramid: multi-skip feature stacking for action recognition. In: CVPR (2015)
23. Liu, W., Hua, G., Smith, J.R.: Unsupervised one-class learning for automatic outlier removal. In: CVPR, pp. 3826–3833 (2014)
24. Ng, J.Y.H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: CVPR (2015)
25. Oneata, D., Verbeek, J., Schmid, C., et al.: Action and event recognition with fisher vectors on a compact feature set. In: ICCV (2013)
26. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. **22**(2), 199–210 (2011)
27. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
28. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Maragos, P., Paragios, N., Daniilidis, K. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010)
29. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
31. Singh, B., Han, X., Wu, Z., Morariu, V.I., Davis, L.S.: Selecting relevant web trained concepts for automated event retrieval. In: ICCV (2015)
32. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
33. Sriperumbudur, B.K., Gretton, A., Fukumizu, K., Schölkopf, B., Lanckriet, G.R.: Hilbert space embeddings and metrics on probability measures. J. Mach. Learn. Res. **11**, 1517–1561 (2010)
34. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using lstms. In: ICML (2015)
35. Sukhbaatar, S., Fergus, R.: Learning from noisy labels with deep neural networks. **2**(3), 4 (2014). arXiv preprint arXiv:1406.2080
36. Sun, C., Gan, C., Nevatia, R.: Automatic concept discovery from parallel text and visual corpora. In: ICCV, pp. 2596–2604 (2015)
37. Sun, C., Nevatia, R.: Large-scale web video event classification by use of fisher vectors. In: WACV (2013)
38. Sun, C., Shetty, S., Sukthankar, R., Nevatia, R.: Temporal localization of fine-grained actions in videos by domain transfer from web images. In: ACM Multimedia, pp. 371–380 (2015)
39. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)

40. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: C3D: generic features for video analysis. In: ICCV (2015)
41. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
42. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: CVPR, pp. 4305–4314 (2015)
43. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L.V.: Temporal segment networks: towards good practices for deep action recognition. In: ECCV (2016)
44. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2691–2699 (2015)
45. Yan, Z., Zhang, H., Piramuthu, R., Jagadeesh, V., DeCoste, D., Di, W., Yu, Y.: HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition. In: ICCV, pp. 2740–2748 (2015)
46. Ye, G., Li, Y., Xu, H., Liu, D., Chang, S.F.: Eventnet: a large scale structured concept library for complex event detection in video. In: ACM Multimedia, pp. 471–480 (2015)
47. Zha, S., Luisier, F., Andrews, W., Srivastava, N., Salakhutdinov, R.: Exploiting image-trained CNN architectures for unconstrained video classification. In: BMVC (2015)
48. Zhang, H., Hu, Z., Deng, Y., Sachan, M., Yan, Z., Xing, E.P.: Learning concept taxonomies from multi-modal data. In: ACL (2016)