



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

*Università degli Studi di Padova*

*Padua Research Archive - Institutional Repository*

Learning without prejudice: Avoiding bias in webly-supervised action recognition

*Original Citation:*

*Availability:*

This version is available at: 11577/3265158 since: 2019-12-03T11:29:14Z

*Publisher:*

*Published version:*

DOI: 10.1016/j.cviu.2017.08.006

*Terms of use:*

Open Access

This article is made available under terms and conditions applicable to Open Access Guidelines, as described at <http://www.unipd.it/download/file/fid/55401> (Italian only)

(Article begins on next page)

# Learning without Prejudice: Avoiding Bias in Webly-Supervised Action Recognition

Christian Rupprecht<sup>a,b,\*</sup>, Ansh Kapil<sup>a</sup>, Nan Liu<sup>a</sup>, Lamberto Ballan<sup>c</sup>, Federico Tombari<sup>a</sup>

<sup>a</sup>Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany

<sup>b</sup>Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218, USA

<sup>c</sup>Università degli Studi di Padova, Via Trieste 63, 35121 Padova, Italy

## Abstract

Webly-supervised learning has recently emerged as an alternative paradigm to traditional supervised learning based on large-scale datasets with manual annotations. The key idea is that models such as CNNs can be learned from the noisy visual data available on the web. In this work we aim to exploit web data for video understanding tasks such as action recognition and detection. One of the main problems in webly-supervised learning is cleaning the noisy labeled data from the web. The state-of-the-art paradigm relies on training a first classifier on noisy data that is then used to clean the remaining dataset. Our key insight is that this procedure biases the second classifier towards samples that the first one understands. Here we train two independent CNNs, a RGB network on web images and video frames and a second network using temporal information from optical flow. We show that training the networks independently is vastly superior to selecting the frames for the flow classifier by using our RGB network. Moreover, we show benefits in enriching the training set with different data sources from heterogeneous public web databases. We demonstrate that our framework outperforms all other webly-supervised methods on two public benchmarks, UCF-101 and Thumos'14.

*Keywords:* Action Recognition, Webly-Supervised Learning

## 1. Introduction

In recent years, deep learning has fueled a significant progress in several computer vision tasks. One of the main reasons behind such achievements is the development of large-scale datasets with annotations [1, 2, 3, 4], which enable training of deep neural networks with millions of parameters without over-fitting. This has led deep learning based models to approach human level performance on various visual data classification and recognition tasks [2]. However, data annotation has intrinsic limitations – both in terms of time and cost. This is even more critical for video data, since annotating action labels and defining temporal bounds for thousands of videos is particularly tedious and time consuming, which makes it a non-scalable solution. Moreover, this manual annotation process often introduces a bias towards very specific tasks and domains [5, 6].

To overcome these limitations, the *webly-supervised* paradigm has emerged as an appealing alternative which aims at learning features and training models by solely relying on noisy web data [7, 8, 9, 10, 11]. The clear advantage is that the amount of data obtainable from the Internet is huge and continuously growing, and vastly exceeds what is achievable through manual annotation. Notably, webly-supervised approaches already perform competitively to the state-of-the-art for certain visual recognition tasks, such as image classification and object detection. For example, in their recent paper Chen and Gupta



Figure 1: Web images and videos collected from heterogeneous web sources are characterized by different appearance and noise. In this work we present a fully *webly-supervised* approach for recognizing and localizing a large number of action categories in trimmed and un-trimmed videos.

[12] show that a CNN trained only with web images gives comparable performance to the ImageNet [1] pre-trained network architecture for object detection, without using a single manually annotated label. Similarly, Joulin *et al.* [11] show that it is possible to train CNNs on massive weakly-labeled image col-

\*Corresponding author.

Email address: christian.rupprecht@in.tum.de (Christian Rupprecht)

lections – such as the 100 million Flickr images dataset [13] – for learning good visual representations.

Inspired by this line of work, we propose a CNN-based, webly-supervised method for the task of human action recognition from real world videos, where the visual data is entirely collected from the web. This task has only recently started to be addressed in literature [14, 15, 16] and poses several challenges. First, the labels obtained through web search queries are often noisy and do not represent the content – in terms of actions – of the retrieved video. Hence, a proper filtering has to be adopted to remove outliers from the samples that will be used as training set. Second, a key challenge is represented by *untrimmed* videos. Conversely to their *trimmed* counterpart, which contain one single action for their whole length, these videos also include several background frames, i.e. without any human activity. Often, such videos are long and might contain multiple actions, belonging to either the same or even different classes. To overcome such challenges, a common strategy is to train an initial set of classifiers from a particular source (e.g. images) and use the classifier scores to filter out noise and represent the other source (e.g. videos) [17, 18, 19, 15]. A good example of this approach is the recent work of Gan *et al.* [15] in which video concepts are discovered using two separate classifiers. The first one is trained on noisy data obtained from a specific web source (i.e. YouTube videos), then it is used to clean the samples used to train the second classifier (trained on Google images).

Our approach stems from the key insight that this procedure biases the second classifier towards samples that the first one understands well, since it subjectively limits the variability of the training set. The proposed idea is thus to train two independent classifiers – in particular, two CNNs – one specialized on web images and video frames, the other one encompassing temporal information obtained from optical flow computed on subsequent video frames. We demonstrate that the use of two independent networks can yield significant benefits in terms of classification accuracy, rather than having the image-based classifier selecting the frames for the optical flow-based classifier. In addition, we also improve the generalization ability of the networks, since most action images retrieved with Google Image Search are usually background free with the human in the center. The idea is thus to enrich the training set with different heterogeneous web data sources. To this end, we include images from Flickr, as well as frames collected from real world videos available on Youtube, the latter being particularly useful also to reduce the *semantic gap* between the actions being sought and the images downloaded from the web.

The proposed approach is tested on two publicly available datasets, namely UCF-101 [20] and Thumos'14 [21], demonstrating state-of-the-art results with respect to other webly supervised approaches, as well as a performance comparable to several recent approaches trained with clean, manually annotated datasets.

## 2. Related work

There is a long history of research in video event detection and action recognition. Thorough related surveys are given in [22, 23, 24].

*Video Recognition and Action Understanding.* Early work focused on defining hand-crafted features for video analysis such as 3D Histogram of Oriented Gradients (with the third dimension being time) and spatio-temporal features based on optical-flow [25, 26, 27]. State-of-the-art hand-crafted features, i.e. the dense trajectories of Wang *et al.* [28], achieved excellent results on multiple action recognition benchmarks [29, 30]. However, handcrafting features requires a rich domain knowledge. In real-world videos, the construction of such features may vary consistently among different domains, and hence application specific features might be required [31, 32].

Following the success of deep learning architectures in image classification [33, 2], there have been attempts to extend this paradigm to videos as well. To this end, [34] proposes a technique to extend CNNs for video analysis, where features are learned from both, spatial and temporal dimensions by using 3D convolutions. Karpathy *et al.* [35] and Tran *et al.* [36] learn spatio-temporal filters and employed different pooling schemes across the temporal axis to address time. Simonyan and Zisserman [37] propose a two-stream convolutional network for action recognition in which they learn separate CNNs for different input sources. First, they train a model on RGB video frames for spatial features, and then they use optical flows between video frames to incorporate motion information. Temporal segment networks have been recently proposed to overcome the limitation of these two-stream CNNs in modeling long-range temporal structure [38]. Duta *et al.* [39] investigate fusion schemes of different features for action recognition. However, the common setup is to rely on a large dataset annotated by human experts in a fully supervised setting.

*Learning from Web supervision.* Previous work for learning visual knowledge from the web typically focuses on images and addresses tasks such as image classification and object detection [7, 8, 9, 40]. Recently, some works have also proposed to learn CNNs [11, 12] and visual concepts [10, 41, 42] from noisy web data. This is also closely related to the vast literature on image and video tagging [43, 44, 45]. Inspired by this line of work, we focus on the specific problem of action recognition in large-scale video data.

There have been few attempts to tackle video understanding tasks using a webly-supervised approach. In their seminal work, Ikizler *et al.* [17] present a model for action recognition in videos using web data. After collecting images from the web, a person detector is used to filter out noisy images, then a linear classifier is learned on the resulting features and applied to the test videos. Similarly, in [18, 19] a set of concept detectors is discovered from the web using Google or Flickr images. After this preliminary learning step, these concept models are applied to the videos to generate video-level representations, which can be used in a fully supervised setting for event detection. In [46]

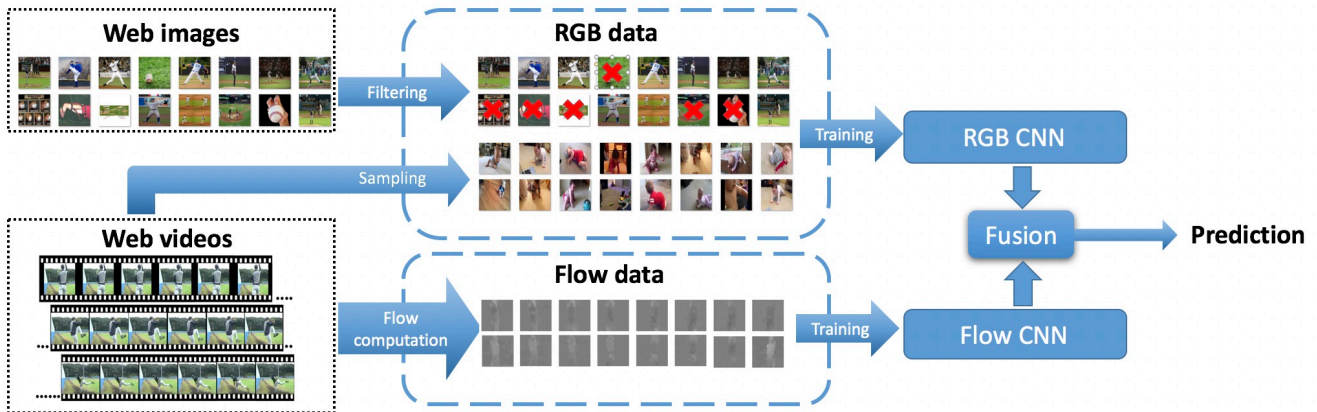


Figure 2: Overview of the proposed model.

Ma *et al.* show that adding web images to supervised training sets can improve performance of the resulting classifiers. More closely related to our work is the recent “Lead-Exceed” Neural Network presented in [16], where a CNN network trained on web videos is refined using images collected from the web in a curriculum learning manner. Similarly, [14] uses a domain transfer scheme between images and videos to filter out the noise in the other domain, and obtain so-called Localized Actions Frames (LAFs). An LSTM over LAFs is then applied for fine-grained action recognition and localization. The localization problem is also tackled in [47], where they present a weakly-supervised model to perform spatio-temporal localization in videos. However, in these approaches the proposed procedure biases the recognition process towards samples that are recognized in the first domain or modality.

### 3. Proposed Approach

In this work we exploit web images and videos for action recognition and localization in a fully webly-supervised fashion. Figure 2 shows our model structure. An important observation is that different data sources provide varied types of data and labels. For example, images retrieved from Google image search are usually clean. *Clean* in this case means that usually the object or action of interest is in the center of the image, no other object is present and the background is white or very simple. This differs greatly from the way the actions/objects appear *in the wild*, where there is clutter, defocus, complex backgrounds, etc. Web videos usually are closer to the real data since it is much more work to create a clean video. Thus the RGB frames from web videos usually depict the actions in real scenarios, which closes the semantic gap and provides training data closer to the test sequences. Compared to Google, images from web sources like Flickr are more natural, in the sense that the object is usually depicted *in the wild*. This discrepancy between different sources produces an effect that we dub as the *source bias*. Another biasing factor arises when using a supervised classifier to filter outliers to determine a training set for a successive classifier (like in [17, 18, 19, 15]). In this case, the bias is introduced by the fact that the “inliers” determined

by the first classifier will be only samples that were well understood by the first classifier. We refer to this problem as *filter bias*.

We propose a methodology that avoids both biasing factors, respectively by employing the following strategies:

- **Source Bias:** mix data from different web sources to reduce the source bias introduced by prevailing image structures typical for a single data provider.
- **Filter Bias:** avoid using a supervised procedure to filter outliers so not to introduce bias through a specific training set.

Our action recognition pipeline consists of two main components: *i*) data collection and filtering (Section 3.1 and 3.2). *ii*) training a two-stream CNN architecture, one stream based on RGB data and the other based on optical flow, similar to [37] (Sections 3.3 and 3.4).

We summarize the proposed pipeline in four steps: *i*) download a set of web images  $\mathcal{I}$  and a set of web videos  $\mathcal{V}$  for each action label; *ii*) filter  $\mathcal{I}$  and  $\mathcal{V}$  to obtain  $\hat{\mathcal{I}}$  and  $\hat{\mathcal{V}}$ ; *iii*) train a network on RGB frames from the filtered sets  $\hat{\mathcal{I}}$  and  $\hat{\mathcal{V}}$ ; *iv*) train a network on stacks of optical flow maps from the set  $\hat{\mathcal{V}}$ . We detail the individual steps of our pipeline in the following subsections.

#### 3.1. Data Collection

Google search often returns images in which the human actor is located in the image center in front of a uniform background. Training with this data can lead the model to underperform in cluttered scenes. To resolve this, we add real world images chosen from Flickr, which are more diverse in terms of actor position and background scene. Relevantly, Flickr has been used for many benchmark datasets like Pascal VOC [48]. Additionally, web searches for some classes return results which do not represent an action in the wild. There is a so called *semantic gap* [49, 43] in how the action appears in real world videos and how they are depicted in web images. This semantic gap can be reduced by using RGB frames from web videos which we collect from YouTube. As in the case of image data,

YouTube is queried with the action names. Finally, since web videos are often rather long and the action can appear in any part of the sequence, we add action videos as animated GIF files by querying Google search by activating the “animation result only” filter, as well as Giphy<sup>1</sup>, an online GIF database to download GIF files for the action. Due to poor compression inherent in the file format, GIFs are usually very short, this in turn increasing the probability that the majority of a sequence will represent the desired action. Please note that even GIFs need to be filtered, since although the chance is high that they display a well trimmed action, it can still be the wrong one.

Similar to Gan *et al.* [16], we perform small changes to the category labels for querying the web. For example for the label `nun chucks`, the search term is modified to `doing nun chucks` to avoid retrieving pictures and videos of the object instead of the corresponding action.

### 3.2. Filtering

The images collected from the web contain outliers that do not belong to the query, such as different action classes, animations, or just text. To minimize the influence of these samples it is necessary to filter the retrieved images before training a classifier. To portray an example, Figure 3 shows the image search results given the query `doing archery` from Google image search. Clearly, the images highlighted in the red box are outliers since they do not depict any human action, and will confuse the model being learned if included in the training set.

As previously mentioned, we aim at filtering the image set in a way that prevents the aforementioned filter bias. Filter bias can formally be described as a set  $C \subset \mathcal{X}$  of correctly labeled samples being corrupted by outliers  $O \subset \mathcal{X}$ . We are given the superset  $\mathcal{S} = C \cup O$  and need to find a filter function  $f : \mathcal{X} \rightarrow \{-1, 1\}$  that classifies elements of  $C$  as positive and detects outliers from  $O$  as negative such that the selected set  $\hat{C} = \{x | f(x) = 1, x \in \mathcal{S}\}$  equals the clean samples  $\hat{C} = C$ . Hence, filter bias occurs when  $f$  selects a strict subset of  $\hat{C} \subset \mathcal{S}$  which changes the distribution of samples and thus biases any classifier trained on it.

A common strategy deployed by state-of-the-art webly supervised approaches for action recognition [17, 14, 16] is to train a classifier that is then used to filter out noise on the remaining set. We will hereinafter refer to this paradigm as *supervised filtering*. As we will show in the experimental section, such supervised filtering approaches easily lead to filter bias, as they tend to filter out difficult but correct samples, thus biasing the resulting set  $\hat{C}$  to contain many simple examples and only few difficult ones.

Hence, we suggest to use an approach for filtering web data outliers that does not rely on a training set. We refer to this filtering paradigm as *independent filtering*. Specifically, we rely on a filtering algorithm based on random walk similarly to [50, 47]. To this end, we define a fully connected graph  $Z(N, E)$ , where  $N$  is the set of all  $n$  images and  $E$  represents the set of edges between them. We map an image  $N_i$  to a feature vector using  $\phi(N_i)$ . The Euclidean distance in feature space

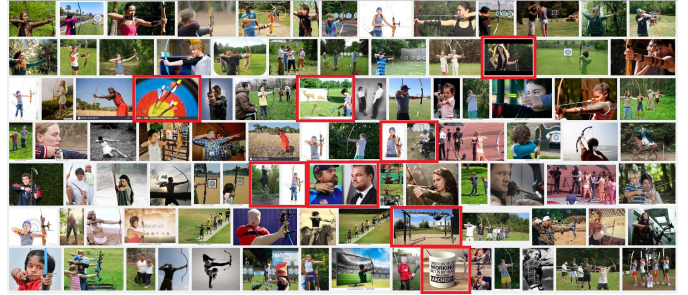


Figure 3: **Query:** `doing archery` (from `images.google.com`). We highlight all the frames that we do not expect to encounter in a video depicting a human doing `archery` with a red border.



Figure 4: Images of action `archery` after applying random walk filtering. **Left:** image subset obtained by removing the noisy images. **Right:** images filtered out by random walk.

of a pair of nodes  $\phi(N_i)$  and  $\phi(N_j)$  is a measure of similarity. A small distance implies similar images. The transition probability between any two nodes  $N_i$  and  $N_j$  is given by:

$$P_{i,j} = \frac{e^{-\gamma \|\phi(N_i) - \phi(N_j)\|_2}}{\sum_{m=1}^n e^{-\gamma \|\phi(N_i) - \phi(N_m)\|_2}} \quad (1)$$

We will compute the relevance  $r_k(N_j)$  of image  $N_j$  iteratively over iterations  $k$ . Let  $v_j = \frac{1}{n}$  be the initial probabilistic score. The update rule can then be written as:

$$r_k(j) = \beta \sum_i^n r_{k-1}(i) p_{i,j} + (1 - \beta) v_j \quad (2)$$

where  $\beta$  controls the contribution of both terms to the final score. In all our experiments we set  $\beta = 0.99$  and  $\gamma = 0.01$ . The filtered image set for the `archery` action are shown in Figure 4. Most images which do not contain any human action such as target boards, arrow set, clip-art images are sorted out.

### 3.3. RGB Network

As for the RGB network, four different models were trained on color frames to evaluate the contribution provided by different web sources:

- **Web images:** trained with only web images collected from Google and Flickr. After filtering with random walk, the best 450 images are selected for training.

<sup>1</sup><http://giphy.com/>

- **Web videos:** per class, we sample 50 videos from YouTube to obtain 1500 frames. After random walk, the 500 highest ranked frames were chosen for the training set.
- **Web images + videos:** the training set includes 400 web images, 500 frames from videos and 100 additionally frames from animated GIFs, for a total of 1000 training samples per class. We train two variants: one’s training set is filtered by random walk, the other’s by the network trained only on web images.

For all networks, the images are randomly separated into a training (80%) and a validation set (20%). The network, a 50-layer ResNet [51], is trained using Caffe [52] with pre-trained weights from ImageNet [1]. Stochastic Gradient Descent (SGD) with a batch-size of 10 and a learning rate of  $10^{-5}$  (decreased 10 times after every 100k iterations) is used. After 200k iterations the optimization converges.

### 3.4. Optical Flow Network

Motion is an important cue for the identification of actions. When the models are trained only with still RGB images, the lack of temporal information affects performance. In this section, the focus is on adding temporal motion information to the model. We train a CNN using optical flow images as input to predict the action class. Optical flow between pairs of consecutive video frames represents a short motion. To capture longer temporal dependencies, the optical flow images are stacked for a sequence of frames. Such inputs implicitly describe the motion in a sequence, which makes recognition easier.

Similar to RGB images, the input for the network in case of optical flow is also in the form of a volume. The flow images are split into horizontal ( $x$ ) and vertical ( $y$ ) displacements. These displacements are stacked one after another as multiple channels, with the goal of modeling longer temporal dependencies. For frames of size  $w \times h$ , the input volume for the network will be  $w \times h \times 2D$ .  $D$  represents the number of stacked optical flow frames.

For the flow classification model, the feature map of the first convolutional layer has size  $64 \times 3 \times 7 \times 7$  since it was trained on 3-channel RGB images. The mean of the weights across the 3 channels can be replicated  $2D$  times to match the new input dimensions. The resulting first layer will thus have a size of  $64 \times 2D \times 7 \times 7$ . The weights of all remaining layers are kept the same for initialization to train a Resnet-50 model [51]. Again, SGD is used as a solver with batch size of 10. The training is performed over 60 epochs and the initial learning rate is set to  $10^{-3}$  which is reduced by a factor of 10 after every 20 epochs. The optical flow between two consecutive frames is computed using Brox’s method [53].

Each frame sequence is run separately through the RGB and the flow network to obtain video level probabilities. To get a final score, the probabilities from the two networks are combined by two different schemes, namely *fusion-by-averaging* and *fusion-by-product*. In the first case, we compute the element-wise average of the two probability vectors from the RGB and the flow CNN. In the second case, instead, we compute the

element-wise product of the two probability vectors from the two networks. The class that corresponds to the entry with maximal probability forms the final prediction.

### 3.5. Action Understanding Tasks

A major benefit of the proposed pipeline is that the trained networks can be used for three different action understanding tasks. We show how the networks, trained on the same set of action labels, can be used for trimmed action classification, untrimmed action classification and action localization. To the best of our knowledge, this is the first work that employs the same fully webly-supervised method for all three tasks.

Action classification in trimmed videos is arguably the easiest of the three tasks, since at test time the algorithm is given a short video that fully contains a single action to be classified. Since the same action covers the whole clip, there is no presence of background actions which might confuse the network. To compute a label for each video, the probability vectors corresponding to each frame, obtained by forward pass through the trained CNNs, are averaged along the temporal axis to get the final score for the video. Differently, action classification in untrimmed videos includes the additional challenge that the video is not cut around the action. Videos are typically longer and the action to be recognized spans only a short portion of the whole clip. In addition, the action to be recognized could be present in one or multiple instances. Although the system is trained only on actions, it needs to be robust to long background sequences in between actions. Here, we also average all frame-wise probability scores temporally to yield the score for each video.

Finally, untrimmed action localization is the task where, in addition to the action label, temporal action boundaries defining the start and end moments of the action in the video have to be estimated. This problem is particularly challenging since the background frames often bear some resemblance to the action. For instance, most videos with activity labels such as "diving", "breast stroke", "front crawl" are all similar to each other, as they contain a swimming pool. We employ two different techniques to perform the temporal localization of actions. As for the first one, referred to as *frame-by-frame* localization scheme, the global level prediction for the whole video is first obtained from the network. Then, the action is localized on a frame-by-frame basis. Specifically, all video frames conforming to the global prediction above a certain probability threshold are grouped together along the temporal axis. Such sequences that are longer than 0.1 seconds count as positive localizations. The second localization scheme, denoted as *sliding window*, uses a small sub-window of the full video and tries to localize the action therein. Differently to *frame-by-frame* localization, no global level predictions or thresholds are taken into account. If an action is predicted in the sub-window by averaging its single predictions, the whole window’s temporal bound is reported.

## 4. Results

We benchmark our framework on two publicly available large-scale datasets, UCF-101 [20] and Thumos’14 [21], and

compare to the state of the art.

#### 4.1. Datasets and Metrics

**UCF-101** [20] is a large-scale dataset consisting of 13,320 trimmed videos from 101 action classes. We test our webly-supervised model on the provided three test splits consisting of 3783, 3734 and 3696 video respectively. The metric used to measure performance is classification accuracy (Acc.) averaged across the three test splits.

**Thumos'14** [21] is a large scale dataset consisting of only untrimmed videos from the same 101 action classes as of UCF-101. An untrimmed video may contain one or multiple instances of same or different actions within its temporal bounds. This dataset presents two tasks: (1) action recognition in video, as well as (2) localizing the action temporally among the videos. The test set consists of 1574 videos from 101 action classes, but the localization task is only applicable for 20 selected classes. The metric used for evaluation as per official Thumos'14 protocol is mean Average Precision (mAP).

#### 4.2. Analysis of Filtering Method

In this section we compare the random walks filtering technique to the E-LDA [54] based filtering of [12]. To be able to use ground truth for this task, we use images from the PASCAL VOC 2012 dataset, which in turn was created from web (flickr) images. For all our experiments we take all images of one of the 20 classes and subsequently add more and more 'noisy samples' in the form of images from the other classes. We then run E-LDA and RW filtering on the corrupted dataset and can precisely measure how well the outliers are removed by the filtering procedure. The experiment is repeated and averaged over 19 classes<sup>2</sup>. To the original image we add 1%, 2%, 3%, 4%, 5%, 10%, 15% and 20% noise samples and then filter with the same thresholds in all noise levels. As expected the best performance is achieved when the noise level matches the filtering amount. When more is filtered than actual noise, naturally the recall decreases while precision stays maximal.

The results can be found in Figure 5. The Figure clearly shows that RW filtering achieves a better performance than E-LDA at all noise levels. Furthermore, it is noteworthy that RW-filtering with up to 15% noise always reaches 100% precision with over 90% recall. This ensures that no false positives are left in the data and the network can be trained with only clean images. It is also interesting to note, that, especially for higher noise levels, the recall first drops and then the precision increases. This indicates that there are some outliers that are difficult to detect for both filtering techniques and are only removed by RW with a higher threshold. E-LDA is not able to identify the outliers especially with noise greater than 5%.

#### 4.3. Evaluation of Bias Removal

We will analyze the effects of the two different types of biases on our model. Firstly, we analyze the effects of source

<sup>2</sup>Class *person* was excluded due to the much larger amount of images which would take more than one week for the noise analysis for E-LDA.

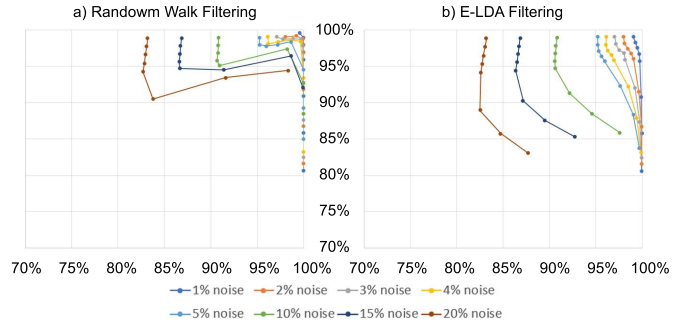


Figure 5: Analysis of filtering methods on the UCF-101 dataset. a) shows precision-recall curves filtering results for the Random Walk based method while b) depicts the results for E-LDA. In all noise levels RW filtering performs better than E-LDA.

Network trained on	Accuracy %
web video frames	52.40
web images	62.44
video frames + images	<b>65.60</b>

Table 1: Source Bias experiment: performance on the UCF-101 dataset of different networks trained on RGB data only.

bias on UCF-101. Table 1 shows the benchmark results of training three networks. All training sets are filtered by random walk. When using web video frames only, we achieve the lowest performance since web videos are untrimmed and the desired action frames are often surrounded by large number of background frames leading to difficulties in removing outliers. On the other hand, using web images alone, we gain 10% accuracy since there are usually less outliers in the search results. Finally, by combining the data of the two sources we achieve the best performance. This result clearly shows that combining different data sources improves the generalization ability of the trained classifier, as the source bias decreases.

In a second experiment, we investigate the effects of filter bias. In Table 2 we compare the recognition accuracy for optical flow maps by comparing two filtering methods. We select RGB frames from the videos, compute its optical flow map and add it to the training set. We show two methods for RGB frame selection. One is by using the independent Random Walk filtering, the other uses RGB network from the previous experiments to select frames with high confidence. In the results one can clearly observe the degraded performance induced by filtering with the classifier. This is caused by two factors: the RGB network works well for video frames of actions that can be classified easily by a single frame. This is not necessarily true for optical flow maps. In fact, the whole idea behind using flow additionally to the images was to create two complementary classifiers. Pre-selecting the training data with the RGB network defeats this purpose. Secondly, for some classes the RGB network is almost never confident enough to generate sufficient training data for the flow network.

To show that the filter bias is not specific to trimmed video flow classification only, we show an additional experiment, this time evaluating on the untrimmed Thumos'14 dataset using an

Input	Filter	Accuracy %
Flow only	RGB net	15.46
Flow only	Random Walk	<b>50.67</b>

Table 2: Filter Bias experiment: optical flow classification performance by filtering the training set with random walk and the RGB frame classifier on UCF-101 (split 1).

Training Set	Filter	mAP %
web images + videos	filtered videos	58.11
web images + videos	Random Walk	<b>60.89</b>

Table 3: Untrimmed video classification results on Thumos’ 14 dataset.

RGB network. Here we train on web images and video frames jointly but we compare two methods of selecting frames from the video. One is by using the RGB image network to identify confident frames, the other performs a random walk on images and video frames jointly. Again, we observe degraded performance due to filter bias.

#### 4.4. Evaluation of Fusion Schemes

In addition, we analyze the performance of the introduced fusion schemes, *i. e.* fusion-by-average and fusion-by-product, and the influence of the number of stacked flow maps  $D$  input into the flow network in Table 4.

We show that in the end combining all three networks - both flow and the RGB one - yields the best results. Combining both flow networks is beneficial since short ( $D = 1$ ) and longer ( $D = 10$ ) temporal dependencies can be captured. When adding the RGB network, product fusion between different modalities seems to emphasize their synergies. The overall impact of adding temporal information improves the recognition accuracy for 88 out of the 101 classes of UCF-101.

In the past two subsections we have analyzed the influence of bias and shown that independent filtering is superior to filtering by classifying. Additionally, we analyzed performance under different forms of fusion and the hyper parameter  $D$ . In the following, we will compare our approach to the state of the art.

#### 4.5. Action Classification of Trimmed and Untrimmed Videos

We compare our approach to state-of-the-art methods that use training data (Table 5) as well as those being purely webly-supervised (Table 6). Relevantly, without using even a single humanly annotated training sample, our approach performs better than [35, 55] who use annotations for training. In comparison to webly-supervised approaches, our CNN approach works slightly better than Lead-Exceed network without LSTM [16].

For the case of action recognition in un-trimmed video on Thumos’ 14, to the best of our knowledge, we do not have a direct comparison in webly-supervised methods doing un-trimmed recognition. Table 7 shows a comparison of our approach against the methods, both trained on the annotated dataset from the Thumos’ 14 challenge.

Input	Fusion	Accuracy %	
Flow	D=1	-	50.61
Flow	D=10	-	51.52
Flow	(D=1) + (D=10)	average	52.62
RGB + Flow	D=1	average	71.24
RGB + Flow	D=1	product	72.75
RGB + Flow	(D=1) + (D=10)	product	<b>74.7</b>

Table 4: Analyzing the influence of the number of stacked flow maps  $D$  and the two fusion methods on UCF-101.

Method	Accuracy %
Karpathy <i>et al.</i> [35]	65.4
LRCN [55]	71.1
LSTM composite model [56]	75.8
C3D [36]	82.3
Two-stream network [37]	88.0
ST-VLMPF [39]	<b>93.6</b>
RGB+Flow (ours)	74.7

Table 5: Comparison with state-of-the-art methods that use labeled UCF-101 dataset for training.

#### 4.6. Action Localization Results

As previously described, the aim of the action localization task is to recognize and localize an action temporally in a given untrimmed video, *i. e.* the output is represented by a real-valued score indicating the confidence of the prediction, together with the starting and ending frame for the given action. We report experiments for both introduced localization schemes, *i. e.* *Frame-by-frame* and *Sliding window*. For this latter, results for different window sizes are reported.

It can be seen that the frame-by-frame method of prediction works better than the sliding window approach in these experiments. This may be due to over-estimating the temporal bounds with sliding window which causes the intersection over union score to deteriorate. The shorter actions are detected better in a frame-by-frame manner. The proposed method works better than the webly state-of-the-art for smaller overlap ratios. This can be due to detecting only the highlights, *e. g.* actions present in web images, but not the full activity from beginning to end. In comparison to the state-of-the-art methods that use the manually annotated data for training, the webly supervised approach still lags behind in performance.

#### 4.7. Qualitative Analysis

In general, as can be seen in Figure 7, actions that are related to a characteristic scenario or that involve a specific object tend to be accurately recognized, as they can directly be identified from individual RGB frames, which are abundantly available on the web to learn from. As an example, actions like billiards, bowling and actions associated to playing a musical instrument are all identified correctly with almost 100% accuracy by means of RGB nets only. Recognition becomes more difficult with actions which have intricate motion, for which optical flow helps (*e. g.*, the knitting action Figure 6b). However some actions, especially those which involve body move-



Method	Accuracy %
Gan <i>et al.</i> [15]	69.3
Lead-Exceed (w/o LSTM) [16]	74.4
RGB+Flow (ours)	<b>74.7</b>

Table 6: Comparison with webly-supervised state-of-the-art methods tested on UCF-101.

Method	mAP %
Jain <i>et al.</i> [57]	<b>69.3</b>
INRIA Lear [58]	64.4
ours	60.8

Table 7: Comparison of Thumos' 14 results against the state of the art.

ments only, are very difficult for the model to recognize. For instance, Jumping Jacks, Jump Rope and Lunges (0% recall rate) have a limited number of real world images to be trained on, and, as said before, images from Google suffer from source bias. In these cases, optical flow also does not help much due to lack of descriptive samples in the training set, often being confused by similar classes characterized by similar body movements. For instance Lunges is often confused with Body weight squats and Jump Rope (all characterized by similar up-and-down movements).

## 5. Conclusion

Webly-supervised methods open a new direction to extend video analysis tasks to a larger scale and at a lower cost compared to current systems. Our experiments on two large-scale datasets demonstrate that data collected from the web is effective in training powerful models for human activity recognition in videos. Our method is even comparable to some existing approaches that rely on manually-labeled supervised training sets. Additionally, we improve over the state of the art for webly-supervised methods. We identify two biases: source bias and filter bias, that can occur in webly trained models at different stages of the pipeline, and show how they can be effectively reduced. The results further encourage the research into the direction of webly-supervised methods, as it holds the potential to save both time and money associated with data annotation. As analyzed in details at the end of our experimental section, future research work should be targeted at improving webly recognition for actions with limited real world training data and characterized by similar body movements. Further, we believe that the localization task can be improved with typical action proposal methods, however the networks need to be trained to recognize background classes to discard proposals effectively.

## Acknowledgments

L. Ballan was supported by the ECs FP7 under the grant agreement No. 623930 (Marie Curie IOF). C. Rupprecht was supported by the TUM - Institute for Advanced Study (German Excellence Initiative - FP7 Grant 291763). This work was

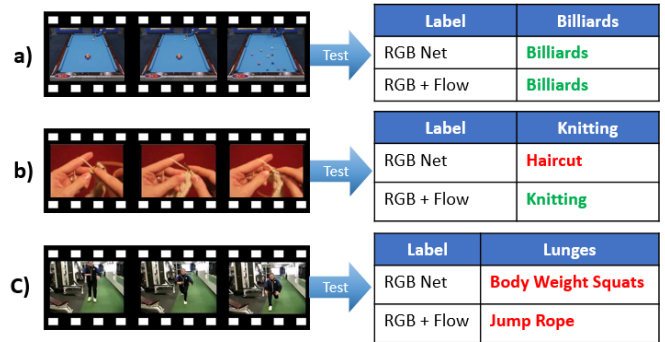


Figure 6: Some examples of test results on UCF-101 dataset

partially funded by the Bavaria California Technology Center (BaCaTec grant No. 12[2016-1]).

## References

- [1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR, 2009.
- [2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.
- [3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: ECCV, 2014.
- [4] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, L. Fei-Fei, Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations, International Journal of Computer Vision in press.
- [5] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, C. K. I. Williams, J. Zhang, A. Zisserman, Dataset issues in object recognition, in: Toward Category-Level Object Recognition, 2006, pp. 29–48.
- [6] A. Torralba, A. Efros, Unbiased look at dataset bias, in: CVPR, 2011.
- [7] L. Li-Jia, L. Fei-Fei, OPTIMOL: Automatic Online Picture Collection via Incremental Model Learning, International Journal of Computer Vision 88 (2010) 147–168.
- [8] X. Chen, A. Shrivastava, A. Gupta, Neil: Extracting visual knowledge from web data, in: ICCV, 2013.
- [9] S. K. Divvala, A. Farhadi, C. Guestrin, Learning everything about anything: Webly-supervised visual concept learning, in: CVPR, 2014.
- [10] B. Zhou, V. Jagadeesh, R. Piramuthu, ConceptLearner: Discovering Visual Concepts from Weakly Labeled Image Collections, in: CVPR, 2015.
- [11] A. Joulin, L. van der Maaten, A. Jabri, N. Vasilache, Learning visual features from large weakly supervised data, in: ECCV, 2016.
- [12] X. Chen, A. Gupta, Webly supervised learning of convolutional networks, in: ICCV, 2015.
- [13] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, L. Li-Jia, YFCC100M: The New Data in Multimedia Research, Communications of the ACM 59 (2) (2016) 64–73.
- [14] C. Sun, S. Shetty, R. Sukthankar, R. Nevatia, Temporal localization of fine-grained actions in videos by domain transfer from web images, in: ACM-MM, 2015.
- [15] C. Gan, C. Sun, L. Duan, B. Gong, Webly-supervised video recognition by mutually voting for relevant web images and web video frames, in: ECCV, 2016.
- [16] C. Gan, T. Yao, K. Yang, Y. Yang, T. Mei, You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images, in: CVPR, 2016.
- [17] N. Ikizler-Cinbis, R. G. Cinbis, S. Sclaroff, Learning actions from the web, in: ICCV, 2009.

Overlap Ratio	0.1	0.2	0.3	0.4	0.5
<b>State-of-the-art</b>					
Wang <i>et al.</i> [59]	18.1	17.0	14.1	11.7	8.3
Oneata <i>et al.</i> [58]	36.6	33.6	27.0	20.8	14.4
<b>Webly state-of-the-art</b>					
Sun <i>et al.</i> [14]	12.4	11.0	8.5	5.2	4.4
<b>Ours</b>					
Frame-by-frame	20.5	16.5	11.2	6.8	3.7
Sliding window	15.4	13.1	9.1	5.4	2.6

Table 8: Results on Thumos'14 test set for temporal localization in untrimmed videos and comparison with the state-of-the-art.

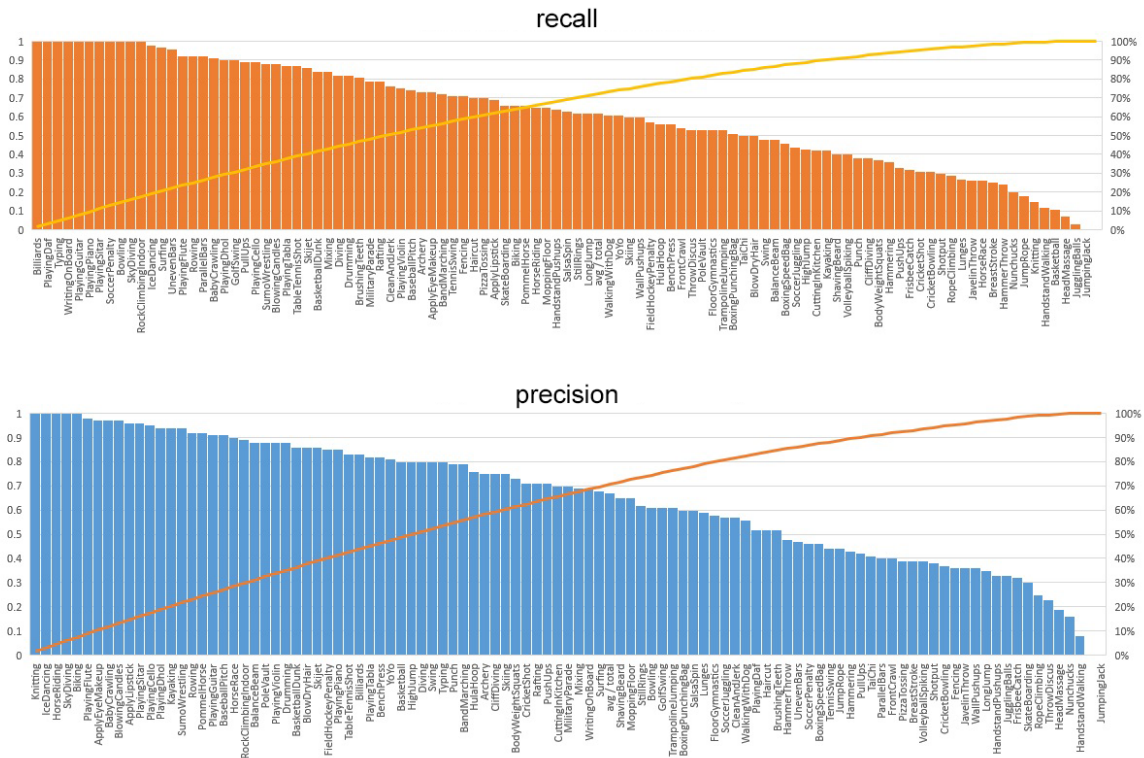


Figure 7: Histogram of class-wise precision and recall rates on UCF-101.

- [18] J. Chen, Y. Cui, G. Ye, D. Liu, S.-F. Chang, Event-driven semantic concept discovery by exploiting weakly tagged internet images, in: ACM ICMR, 2014.
- [19] B. Singh, X. Han, Z. Wu, V. I. Morariu, L. S. Davis, Selecting relevant web trained concepts for automated event retrieval, in: ICCV, 2015.
- [20] K. Soomro, A. R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, in: CRCV-TR-12-01, 2012.
- [21] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, M. Shah, The THUMOS challenge on action recognition for videos “in the wild”, *Computer Vision and Image Understanding* 155 (2017) 1–23.
- [22] R. Poppe, A survey on vision-based human action recognition, *Image and Vision Computing* 28 (6) (2010) 976–990.
- [23] D. Weinland, R. Ronfard, E. Boyer, A survey of visionbased methods for action representation, segmentation and recognition, *Computer Vision and Image Understanding* 115 (2) (2010) 224–241.
- [24] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, G. Serra, Event detection and recognition for semantic annotation of video, *Multimedia Tools and Applications* 51 (1) (2011) 279–302.
- [25] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: ICCV Workshops, VS-PETS, 2005.
- [26] A. Klaser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: BMVC, 2008.
- [27] I. Laptev, M. Marszałek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: CVPR, 2008.
- [28] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, *International Journal of Computer Vision* 103 (1) (2013) 60–79.
- [29] X. Peng, C. Zou, Y. Qiao, Q. Peng, Action recognition with stacked fisher vectors, in: ECCV, 2014.
- [30] H. Wang, C. Schmid, Action recognition with improved trajectories, in: ICCV, 2013.
- [31] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, B. Raj, Beyond gaussian pyramid: Multi-skip feature stacking for action recognition, in: CVPR, 2015.
- [32] C. R. de Souza, A. Gaidon, E. Vig, A. M. López, Sympathy for the details: Dense trajectories and hybrid classification architectures for action recognition, in: ECCV, 2016.
- [33] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012.
- [34] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (1) (2013) 221–231.
- [35] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei,

- Large-scale video classification with convolutional neural networks, in: CVPR, 2014.
- [36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in: ICCV, 2015.
  - [37] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: NIPS, 2014.
  - [38] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: towards good practices for deep action recognition, in: ECCV, 2016.
  - [39] I. C. Duta, B. Ionescu, K. Aizawa, N. Sebe, Spatio-temporal vector of locally max pooled features for action recognition in videos.
  - [40] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, L. Fei-Fei, The unreasonable effectiveness of noisy data for fine-grained recognition, in: ECCV, 2016.
  - [41] Q. Li, J. Wu, Z. Tu, Harvesting mid-level visual concepts from large-scale internet images, in: CVPR, 2013.
  - [42] C. Sun, C. Gan, R. Nevatia, Automatic concept discovery from parallel text and visual corpora, in: ICCV, 2015.
  - [43] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. M. Snoek, A. Del Bimbo, Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval, *ACM Computing Surveys* 49 (1) (2016) 14.
  - [44] W. Yang, G. Toderici, Discriminative tag learning on youtube videos with latent sub-tags, in: CVPR, 2011.
  - [45] L. Ballan, M. Bertini, G. Serra, A. Del Bimbo, A data-driven approach for tag refinement and localization in web videos, *Computer Vision and Image Understanding* 140 (2015) 58–67.
  - [46] S. Ma, S. A. Bargal, J. Zhang, L. Sigal, S. Sclaroff, Do less and achieve more: Training cnns for action recognition utilizing action images from the web, *Pattern Recognition* 68 (2017) 334–345.
  - [47] W. Sultani, M. Shah, What if we do not have multiple videos of the same action? – video action localization using web images, in: CVPR, 2016.
  - [48] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, *International Journal of Computer Vision* 88 (2) (2010) 303–338.
  - [49] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (12) (2000) 1349–1380.
  - [50] H. D. K. Moonesignhe, P.-N. Tan, Outlier detection using random walks, in: ICTAI, 2006.
  - [51] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016.
  - [52] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACM-MM, 2014.
  - [53] T. Brox, A. Bruhn, N. Papenberger, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: ECCV, 2004.
  - [54] B. Hariharan, J. Malik, D. Ramanan, Discriminative decorrelation for clustering and classification, in: ECCV, 2012, pp. 459–472.
  - [55] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: CVPR, 2015.
  - [56] N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised learning of video representations using lstms, *CoRR*, abs/1502.04681 2.
  - [57] M. Jain, J. van Gemert, C. G. M. Snoek, et al., University of Amsterdam at Thumos challenge 2014, in: THUMOS14 Action Recognition Challenge, 2014.
  - [58] D. Oneata, J. Verbeek, C. Schmid, The LEAR submission at Thumos 2014, in: THUMOS14 Action Recognition Challenge, 2014.
  - [59] L. Wang, Y. Qiao, X. Tang, Action recognition and detection by combining motion and appearance features, in: THUMOS14 Action Recognition Challenge, 2014.