Bag-of-Fragments: Selecting and Encoding Video Fragments for Event Detection and Recounting

Pascal Mettes[†], Jan C. van Gemert[†], Spencer Cappallo[†], Thomas Mensink[†], Cees G. M. Snoek[†⋆]
[†]University of Amsterdam [⋆]Qualcomm Research Netherlands

ABSTRACT

The goal of this paper is event detection and recounting using a representation of concept detector scores. Different from existing work, which encodes videos by averaging concept scores over all frames, we propose to encode videos using fragments that are discriminatively learned per event. Our *bag-of-fragments* split a video into semantically coherent fragment proposals. From training video proposals we show how to select the most discriminative fragment for an event. An encoding of a video is in turn generated by matching and pooling these discriminative fragments to the fragment proposals of the video. The bag-of-fragments forms an effective encoding for event detection and is able to provide a precise temporally localized event recounting. Furthermore, we show how bag-of-fragments can be extended to deal with irrelevant concepts in the event recounting. Experiments on challenging web videos show that i) our modest number of fragment proposals give a high sub-event recall, ii) bag-of-fragments is complementary to global averaging and provides better event detection, iii) bag-of-fragments with concept filtering yields a desirable event recounting. We conclude that fragments matter for video event detection and recounting.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: Video analysis

Keywords

Event detection; event recounting; bag-of-fragments; discriminative fragments

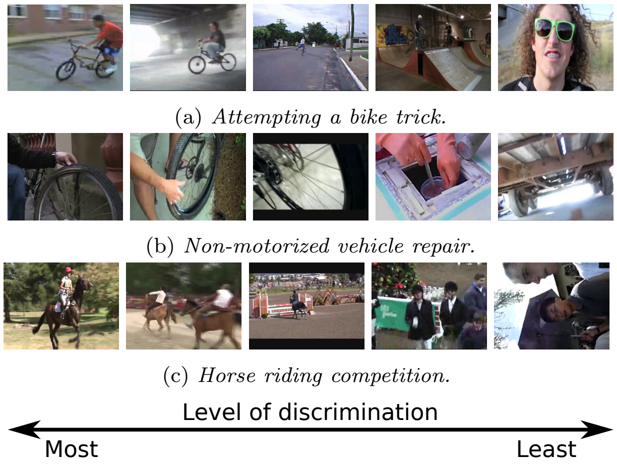

(a) *Attempting a bike trick.*
(b) *Non-motorized vehicle repair.*
(c) *Horse riding competition.*

Figure 1: We propose bag-of-fragments, a video representation that finds and encodes the most discriminative fragments for event detection and recounting. The figure shows the middle frame of five fragments for three events, ordered by level of discrimination. The most discriminative fragments are exemplary for the event and will be included in the bag-of-fragments, while the more ambiguous and less discriminative fragments are ignored.

1. INTRODUCTION

In this work, we focus on detecting events in videos and recounting why an event is relevant by providing the most relevant semantic concepts. This problem is typically addressed by globally aggregating concept detector scores over the whole video [15, 16, 18]. Global aggregation of detector scores poses two problems to event detection and recounting. First, event videos are a complex interplay of various sub-events with a varying degree of relevance [1] which are blended into a single representation. Second, in a global aggregation, the event recounting is unable to state where in the video relevant concepts occur. Similar to related work, we compute concepts scores for frames in a video as the semantic representation, but we aim to perform event detection and recounting on the level of video fragments.

We propose a pipeline to encode a video using fragments that form discriminative sub-events for a complex event, which we call *bag-of-fragments*. For such an encoding, we first need to generate fragments from a video. As the search space of all possible fragments in a video is vast, we propose a hierarchical clustering algorithm to yield a concise set of semantically coherent fragment proposals. The algorithm, inspired by object proposals in images [10, 24], iteratively merges only the most informative fragments. As a result, fragments are generated across all temporal locations and scales of a video, without exhaustively preserving the full search space of fragments.

ICMR'15, June 23–26, 2015, Shanghai, China.
Copyright © 2015 ACM 978-1-4503-3274-3/15/06:.$15.00.
http://dx.doi.org/10.1145/2671188.2749404.

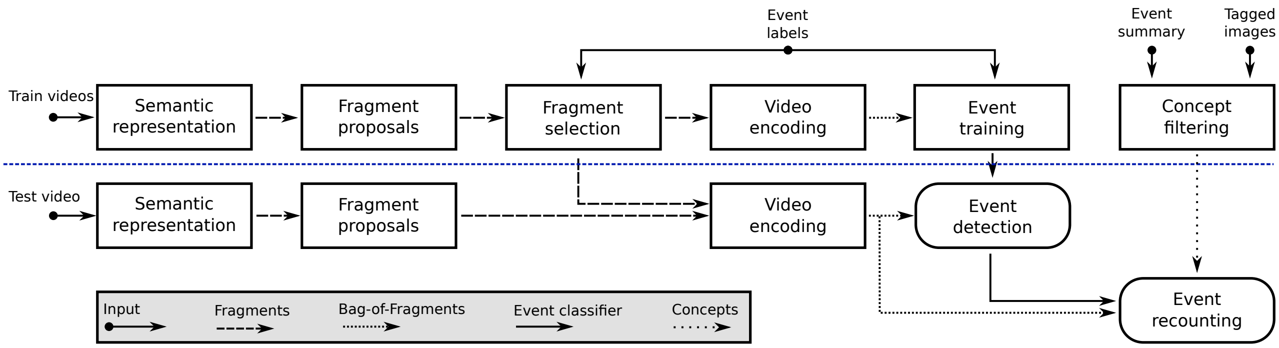


Figure 2: The pipeline of our bag-of-fragments for event detection and recounting.

Based on the fragments proposals of a set of training videos, we select the most discriminative ones of an event. Fig. 1 highlights a number of fragments with various levels of discrimination according to our selection. The selected discriminative fragments form the basis of our bag-of-fragments encoding. The discriminative fragments of an event are matched and pooled over the fragment proposals of a single video, resulting in an effective encoding for event detection. What is more, as the encoding is performed per fragment, information regarding the most informative fragments of a video is retained. Event recounting can therefore be performed by providing the most relevant concepts within the most informative fragments of a video. Lastly, we show how co-occurrence statistics from social tagged images can aid our event recounting by filtering irrelevant concepts.

Experimental evaluation on challenging web videos from the THUMOS [12] and TRECVID benchmarks [20] highlights the effectiveness of our bag-of-fragments for event detection and recounting. First, we show that our fragment proposals are able to retain the most informative fragments at a fraction of full search space of fragments. Second, we show the effectiveness and complementary nature of our bag-of-fragments encoding for event detection compared to a global aggregation of concept scores. Third, we show qualitatively that our bag-of-fragments with concept filtering yields desirable event recounting results.

2. RELATED WORK

Video encodings for event and action detection typically use low-level features that describe the spatio-temporal signal [7, 14, 26]. While these features are well-suited for recognition, they lack any semantic interpretation which complicates recounting why a video is recognized. A solution is offered by Videostory [8], that learns a representation to jointly embeds user tags with video features. The features of a new video can be mapped to this joint feature-tag space and the embedded tags allow recounting the detection evidence. Instead of user tags, which may be noisy, we use high-quality concept classifiers to allow recounting. We will experimentally compare against Videostory [8].

Instead of low-level features, recent video encodings apply a bank of concept classifiers to individual frames and average the frame-based responses for forming a video representation [15, 16, 18]. Such a representation has shown excellent accuracy for event recognition [16, 18] with the added benefit that the concept classification scores provide valuable clues for recounting why the whole video is relevant [8, 15]. Where these works recount a complete video, we instead recount on video fragments, which offer a more precise fine-grained temporal granularity. Rather than averaging concept scores over the whole video, we aggregate scores on coherent video fragments and use them for event detection and recounting.

Although a complex event consists of various sub-events, it can be recognized by a human after seeing only a few well-chosen discriminative video fragments [1]. In automatic event recognition, fragments have been used as latent variables in an SVM optimization [23, 25]. Since latent-SVM is computationally expensive, only a limited number of latent fragments can be used. Instead, our method can exploit a larger set of possible discriminative fragments, which increases the likelihood of finding the most discriminative ones. We draw inspiration from mid-level parts as used in image classification [5, 13]; we automatically discover discriminative video fragments and use them to encode full videos as a bag of their best matching fragments.

To perform a bag-of-fragments encoding of a video, we need to first split a video into coherent fragments that are likely to contain a discriminative sub-event. Instead of a brute-force sliding window [19] or detecting shot boundaries [27] we base our fragments on proposal methods [10, 11, 24]. We propose a fast clustering method to generate a small set of fragment proposals with a high sub-event recall. We will experimentally evaluate our proposals against sliding windows and shot boundary detection.

For event recounting, the highest scoring concept scores in a video are typically used as evidence [4, 15]. Because the highest score is sensitive to noise, Sun et al. [22] propose a manually defined white-list of acceptable concepts per event. We extend this work by replacing the manual white-list with an automatically found list based on co-occurrence statistics using a high-level event description and tagged images, as recently proposed for zero-shot image classification [17]. Automatic white-listing eliminates any manual effort, which is labor intensive and may be prone to errors or subjectivity.

3. BAG-OF-FRAGMENTS

The key contribution of this paper is an encoding of discriminative fragments, which we call bag-of-fragments, for event detection and recounting. The pipeline consists of four major stages. The first stage generates fragment proposals by splitting a video into a set of fragments. The second stage performs fragment selection, by identifying the most discriminative fragment proposals for provided event exam-

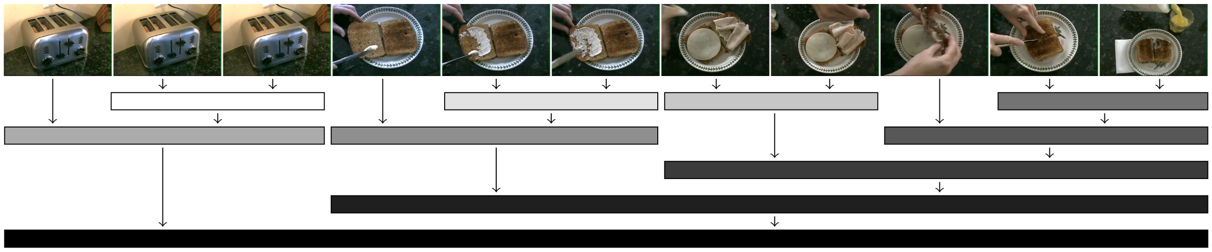


Figure 3: A video of *Making a sandwich*, where the fragment proposals are hierarchically merged using the combined similarity. Note how semantically more correlated proposals of the video are merged earlier (indicated by the colors of the bars).

ples. In the third stage we utilize the discriminative fragments to generate a bag-of-fragments encoding of a video. The encoding forms the basis for both event detection and recounting. The fourth component, concept filtering, is introduced to generate a relevant event recounting. We summarize the pipeline in Fig. 2 and detail the stages next.

3.1 Fragment proposals

To generate a set of fragment proposals for a video, we employ a hierarchical clustering algorithm to cluster fragments into proposals. The main idea behind the hierarchical clustering is to iteratively merge only the most informative fragments, rather than considering all fragment merges. For the clustering, we employ two similarity measures, a semantic and syntactic similarity. The two similarity measures aim to merge the most semantically similar fragments (semantic similarity), while maintaining a balanced cluster tree (syntactic similarity).

Semantic fragment similarity. Let $\mathbf{f}_i \in \mathbb{R}^d$ denote the semantic representation of fragment i containing the scores of d concepts. The semantic similarity between two fragments i and j is then estimated as:

$$S_c(\mathbf{f}_i, \mathbf{f}_j) = \sum_{k=1}^{d} |f_i(k) - f_j(k)|, \tag{1}$$

where $f_i(k)$ denotes the k^{th} concept of $\mathbf{f}_i$. Using this equation as a similarity measure, the two consecutive fragments to be merged at each iteration are the fragments for which Eq. 1 is minimized. Such a clustering algorithm combines the semantically most similar fragments at each iteration, generating semantically coherent fragments. Updating fragments i and j into fragment t can be done efficiently, as we apply average pooling of the concepts within a fragment:

$$\begin{aligned} f_t(k) &= \tfrac{r(\mathbf{f}_i)\cdot f_i(k) + r(\mathbf{f}_j)\cdot f_j(k)}{r(\mathbf{f}_i)+r(\mathbf{f}_j)}, \qquad k = \{1,..,d\}, \\ r(\mathbf{f}_t) &= r(\mathbf{f}_i) + r(\mathbf{f}_j), \end{aligned} \tag{2}$$

where $r(f_i)$ denotes the number of frames in fragment i.

Syntactic fragment similarity. To prevent that a single video fragment gobbles up small fragments one by one, we add another similarity measure that enforces a more balanced cluster tree:

$$S_s(\mathbf{f}_i, \mathbf{f}_j) = r(\mathbf{f}_i) + r(\mathbf{f}_j). \tag{3}$$

The idea behind the similarity measure of Eq. 3 is to penalize large fragments from merging in favor of smaller fragments.

Combined similarity. A combination of semantic and syntactic similarities is given by a linear combination of the terms,

$$S(\mathbf{f}_i, \mathbf{f}_j) = S_c(\mathbf{f}_i, \mathbf{f}_j) + \alpha \cdot S_s(\mathbf{f}_i, \mathbf{f}_j). \tag{4}$$

As the ranges of the two similarity measures are different, the variable α is set to the sum of the concept scores divided by the sum of the sizes for all consecutive fragments at each iteration. This makes both similarity measures of equal importance.

The hierarchical clustering with the combined similarity measure results in a concise set of fragment proposals, ideally retaining those fragment proposals that are semantically coherent. An example of the fragment proposals generated for a video with eleven sampled frames is shown in Fig. 3.

3.2 Fragment selection

From the set of fragment proposals P generated for a set of event training videos, we aim to select the most discriminative ones. We utilize the training videos in two stages. In the first stage, the training videos are used to select which fragment proposals are most discriminative for a given event. In the second stage, a bag-of-fragments encoding is generated for each training and test video based on the discriminative fragments. The encodings are then used to train an event classifier with an off-the-shelf SVM classifier. During training, we are given N training videos $X = [\mathbf{x}_1, .., \mathbf{x}_N]$, where $\mathbf{x}_i \in \mathbb{R}^{s_i \times d}$ denotes the i^{th} training video containing s_i fragment proposals. Furthermore, video labels are provided as $Y = [y_1, .., y_N]$, where $y_i \in \{-1, +1\}$ states whether training video i contains the event. We outline a three step procedure for selecting the discriminative fragments.

1) Generating event fragment classifiers. We first compute an event classifier for each fragment proposal of the positive event training videos. As negative examples we simply use the fragment proposals of negative videos. Rather than explicitly training an SVM classifier for each (positive) proposal separately, we prefer a faster alternative using discriminative decorrelation [9]. We assume that the maximum likelihood estimate of the covariance matrix Σ used in linear discriminant analysis is the sample covariance over all the fragments in the training set X, ignoring class labels. As a result the linear discriminant analysis parameters μ and Σ only need to be computed once over the whole training set. Then, a classifier $\mathbf{w}_{ij}$ for $\mathbf{x}_{ij}$, proposal j of training video i, is efficiently computed as:

$$\mathbf{w}_{ij} = \Sigma^{-1}(\mathbf{x}_{ij} - \mu). \tag{5}$$

Eq. 5 results in a classifier for each fragment proposal, ready to be evaluated on all N training videos.

2) Matching fragment classifiers for video pooling. For each fragment proposal $p \in P$, we perform a matching to all the N training videos by computing the dot-product between the event fragment classifier of p and the fragment proposals of the training video. After the matching we perform a max-pooling operation, which simply retains the maximum dot-product value of the matching for the entire video. The pooling value expresses how much the training video is related to proposal p. By ranking the training videos according to their max-pooled values and comparing the ranking to labels Y, we are able to determine how well the fragment proposal is able to distinguish positive from negative videos containing a specific event. We note that each fragment proposal has a bias in the matching and pooling, namely to the video from which it has originally been retrieved. However, as the bias is equal among all the proposals, it does not lead to overfitting towards specific fragment proposals.

3) Selecting discriminative classifiers. From the set of all fragment proposals P, we aim to select a discriminative subset $F \subset P$. This is performed by selecting the fragment proposals with the best ranking scores. To avoid inclusion of visually similar and therefore redundant proposals, we enforce a constraint on each fragment proposal. The event fragment classifier of each fragment proposal should have a cosine distance of at least 0.5 with respect to the better performing proposals, otherwise, it is removed. Such a constraint results in a diverse set of discriminative fragments [13].

3.3 Video encoding

Now that we have the discriminative fragments for the event, we utilize them to perform a fragment encoding for both the training and test videos. The encoding is performed with the same matching and pooling operation as in step 2 above. Let f denote the number of selected discriminative fragments, i.e. $f = |F|$, then the fragment encoding results in an f-dimensional feature vector for a video. The number of discriminative fragments f is a hyperparameter. In the experiments, we evaluate the influence of the number of selected discriminative fragments per event on the detection performance. The encoding is performed over all training and test videos.

3.4 Concept filtering

Apart from a video representation for event detection, our encoding is also able to perform event recounting. To that end, the fragment proposals of the test video that have resulted in the highest max-pooled values are selected as the most informative fragments. For each of these informative fragments we select the concepts that have contributed most to the corresponding dot-product of the max-pooled value as the informative concepts. For each test video this results in a list of informative fragments and their corresponding concepts.

As indicated by Sun et al. [22], directly selecting the top scoring concepts as the recounting for each selected fragment leads to noisy results, mostly because of concept detector noise. The noise results in incorrect or irrelevant concepts in the recounting, as they were erroneously given a high score. Rather than manually determining which concepts are relevant for an event [22], we propose to automatically filter concepts. More specifically, we use co-occurrence statistics [17] to compare the concepts used in our representation to a textual summary of the event.

As shown in Fig. 2, the concept filtering takes as input a textual summary of the event and a collection of social-tagged images from which we compute the co-occurrence statistics. Each concept in the semantic representation is compared to each concept in the textual event summary using a co-occurrence score. Intuitively, the co-occurrence statistic states that two concepts are related if they occur together relatively often as tags in the same images. As such, concepts with high co-occurrence scores are deemed relevant with respect to the event.

For the social tag dataset, we have retrieved the available subset of 4,770,156 Flickr images in ImageNet [3]. For the set of Flickr images, we have in turn collected the corresponding meta-data, in the form of 14,088,893 unique tags. Roughly 95% of the images contain multiple tags, which make the co-occurrence practical. We have made the meta-data of the Flickr images available online[1].

Let $\mathcal{Z}$ denote the set of concepts from the textual summary of the event. Then for a concept i in the semantic representation, we compute the co-occurrence score to all concepts in $\mathcal{Z}$ using the Dice coefficient [17] and keep the maximum score:

$$C_i = \max_{z \in \mathcal{Z}} \left[2 \frac{c_{iz}}{c_i + c_z} \right], \tag{6}$$

where c_i and denotes the number of images with tag i and c_{iz} the number of images with both tag i and z.

The final score C_i is used here as the relevancy score of concept i with respect to the event. For the concept filtering, we select the concepts with the highest scores according to Eq. 6, where the number of concepts to retain is a parameter. Finally, the event recounting is altered by only recounting the most informative concepts that are also relevant according to the concept filtering. This makes for an event recounting that is less sensitive to the noise in concept detectors and more relevant for the event, as we will show in the experiments.

4. EXPERIMENTAL SETUP

4.1 Experiments

4.1.1 *Fragment proposal quality*

In the first experiment, we evaluate the fragments generated by our fragment proposal algorithm. This evaluation is performed on the THUMOS'14 temporal localization dataset [12]. The dataset consists of 1010 validation videos, each containing several semantically different action-related events at different time intervals. The annotations of the events and their time intervals are provided.

Evaluation. The quality of the fragment proposals is evaluated by examining the recall ratio. For each video, we compare our fragment proposals to the ground truth fragments using the intersection-over-union [6]. Sufficient overlap is achieved if the maximum intersection-over-union is at least 0.5. The ratio of the annotated fragments with sufficient overlap forms the final score.

Baselines. We compare our fragment proposals to two baseline strategies. The first is a shot boundary detection algorithm, where a video is split into a number of non-overlapping fragments based on detected shots within the

[1] https://staff.fnwi.uva.nl/s.h.cappallo/flickrnet.html

video. More specifically, we employ a graph partition model, as proposed in [27]. Here, opponent color histograms are used to represent frames and the continuity signal is thresholded to get shot detections [27]. The second baseline is a sliding window procedure, where a video is split into a number of fragments by sliding a temporal chunk across a video at all temporal positions and scales.

4.1.2 *Event detection*

In the second experiment, we investigate the potential of the bag-of-fragments in the context of event detection. We rely on the TRECVID MED 2014 dataset [20]. This dataset consists of 20 events, where 100 positive videos are provided per event. A general set of 4991 background videos is provided as negative set.

Evaluation. We focus on two evaluations. First, we examine the effect of the number of selected discriminative fragments in our encoding and compare it to a bag-of-fragment encoding using all fragment proposals, instead of the discriminative fragments. Second, we compare our bag-of-fragments to baseline encodings and we evaluate their fusion. The performance of a single event is measured using the Average Precision (AP). At test time, roughly 24,000 videos are given for an event, where the goal is to rank the videos displaying the actual event higher than the negative videos. The quality measure on the whole dataset is in turn measured by the mean Average Precision (mAP) across all the events.

Methods. For the second evaluation, a total of four different methods are applied to the TRECVID MED 2014 dataset. The first is a global model, where the concept scores of a video are averaged over the frames [15, 16, 18]. The second is the state-of-the-art VideoStory, which uses a global model as the visual representation [8]. In addition to these two baselines we evaluate our bag-of-fragments. Finally, we consider a fusion to show the complementary power of bag-of-fragments. For each of the individual methods, the corresponding representations are ℓ_2 normalized and fed to a linear SVM [2]. The SVM outputs are in turn converted to probability values using Platt scaling [21] and the ranking of the videos is performed on these probability values. The fusion is simply performed by computing the product of the probability values for each video.

4.1.3 *Event recounting*

The third experiment focuses on the recounting using the same dataset as used for event detection. We use the model trained for event detection to recount the semantic evidence of a test video, in combination with our concept filtering. We use the textual summary provided by TRECVID for each event to compute the co-occurrence scores.

Evaluation. The evaluation of the event recounting is completed on a qualitative basis. First, we examine the quality of the co-occurrence method for concept selection for 25 concepts in the semantic representation, compared to all 20 events. Second, we show the event recounting results for three test videos from different events, both with and without concept selection.

4.2 Implementation details

Semantic representation. We apply convolutional neural networks to provide the frame-based semantic representations. More specifically, we employ an in-house implementation of [28] trained on 15,293 ImageNet [3] concepts. The frames in each video are sampled once every second. For each frame, we extract the features from the third fully connected layer, the layer before the soft-max, such that the frame is represented by a 15,293-dimensional semantic vector. The aggregation of the frames in a single fragment is performed by averaging the scores per concept [16, 18].

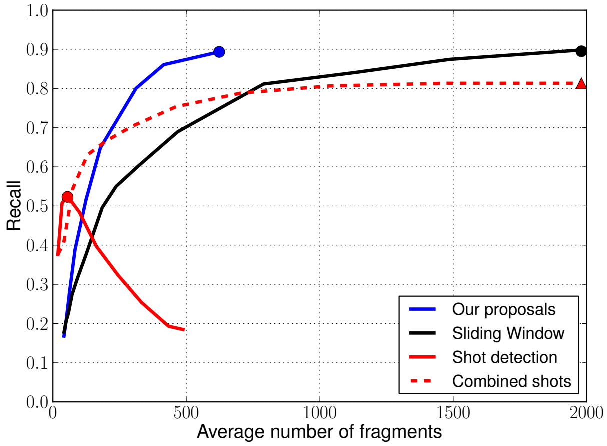


Figure 4: Achieved recall scores as a function of the number of video fragments. Our fragment proposals yield improved recall scores using a fraction of the fragments.

Bag-of-Fragments. For our discriminative fragments, we first extract all the fragment proposals from the positive videos, apply ℓ_2 normalization on each proposal, and compute the efficient event fragment classifier. All the event fragments classifiers are max-pooled over the train videos and the top discriminative fragments are selected. For the discriminative fragment selection, the Average Precision score is used to evaluate the ranking of the max-pooled values per fragment proposal.

5. EXPERIMENTAL RESULTS

5.1 Fragment proposal quality

An overview of the recall as a function of the number of fragments is shown in Fig. 4. For our proposal algorithm and for the sliding window, the number of fragments is varied by varying the size of the initial temporal chunk. For the shot boundary detection, the number of fragments is a function of the shot threshold; the stricter the threshold, the more fragments. We have also added a shot boundary baseline that combines the fragments from multiple thresholds.

As the graph of Fig. 4 shows, our fragment proposal algorithm compares favorably in terms of recall to the two shot detection baselines. Our algorithm yields a peak recall of 0.89, while the shot boundary detection yields a peak recall of 0.52. The limited recall scores of the shot boundary detection are caused by its non-hierarchical nature. By splitting videos solely into non-overlapping fragments, important information is missed. This is further confirmed by the combined shot boundary detection baseline, which yields higher recall scores. The peak recall is however not only 8% lower (at 0.81), but also requires roughly three times as many fragments as our algorithm.

	Global		*This paper*	
Event	**Average** [16, 18]	**VideoStory** [8]	**Bag-of-Fragments**	**Combination**
Attempting a bike trick	5.6%	12.0%	10.6%	20.9%
Cleaning an appliance	5.3%	15.6%	13.2%	25.0%
Dog show	56.6%	74.5%	73.8%	76.5%
Giving directions to a location	6.8%	5.6%	2.4%	9.9%
Marriage proposal	0.6%	0.8%	0.9%	1.4%
Renovating a home	4.3%	10.8%	11.5%	16.7%
Rock climbing	7.3%	14.9%	8.6%	19.7%
Town hall meeting	53.8%	42.2%	37.9%	52.8%
Winning a race without a vehicle	19.3%	17.7%	15.2%	30.4%
Working on a metal crafts project	8.1%	9.5%	18.2%	23.0%
Beekeeping	72.1%	70.3%	78.1%	91.7%
Wedding shower	25.1%	20.4%	30.4%	45.6%
Non-motorized vehicle repair	43.0%	42.6%	52.9%	60.9%
Fixing musical instrument	38.3%	44.2%	58.5%	66.0%
Horse riding competition	50.2%	50.1%	45.0%	64.7%
Felling a tree	12.3%	11.4%	22.5%	28.7%
Parking a vehicle	17.8%	24.1%	16.8%	30.4%
Playing fetch	3.2%	8.0%	4.2%	10.9%
Tailgating	29.6%	30.1%	40.7%	54.3%
Tuning a musical instrument	4.8%	12.2%	10.4%	17.4%
mean	23.2%	25.9%	27.6%	37.3%

Table 1: Event detection results on TRECVID MED 2014 for global averaging, VideoStory, and our bag-of-fragments in average precision. We also report the combination between global averaging and bag-of-fragments. Bag-of-fragments is best and highly complementary to existing encodings.

The peak recall of the sliding window baseline is equal to our algorithm, but sliding window requires far more fragments (3.2x). This result highlights the effectiveness of our method. Rather than going through all possible fragment combinations, we only examine the most promising combinations across the hierarchy. This results in less fragment proposals, without sacrificing recall.

5.2 Event detection

For the event detection, we first evaluate the primary bag-of-fragments parameter, namely the number of selected discriminative fragments. Fig. 5 shows the effect of the number of fragments on the mean Average Precision (mAP) score. Using only two discriminative fragments yields a mAP of 6.0%. The performance increases monotonously as the number of discriminative fragments increases, indicating that a rich set of fragments to represent an event is beneficial. At roughly 2000 discriminative fragments, the performance starts saturating, with an mAP of 27.6% and we will use this setting for the rest of the experiments. Furthermore, we have evaluated the performance using all fragment proposals, which resulted in a mAP of 26.6%. This result indicates that only using discriminative fragments not only results in a more compact encoding, but also leads to improved results.

For the comparative evaluation we show an overview of the results for the four different methods for the 20 events Table 1. Compared to the global averaging baseline, our algorithm yields improved Average Precision scores for 15 of the 20 events, with an absolute increase of 4.4% in mean Average Precision, from 23.2% to 27.6%.

Noteworthy is the difference in performance across different events. Our discriminative fragments improves upon the baseline with 20.2%, 17.2%, and 10.2% (absolute difference) for the events *Fixing a musical instrument*, *Dog show*, and *Felling a tree*. The baseline method performs 15.9% and 5.2% better for the events *Town hall meeting* and *Horse riding competition*. This indicates a complementary nature between the two models. Indeed, a fusion between the two improves the performance significantly with a mean Average Precision of 37.3%, an notable absolute improvement of 14.1% over the baseline model. Note that from the fragment point-of-view, the fusion with the global averaging baseline comes computationally for free, as the global average is always the last merge in our hierarchical clustering.

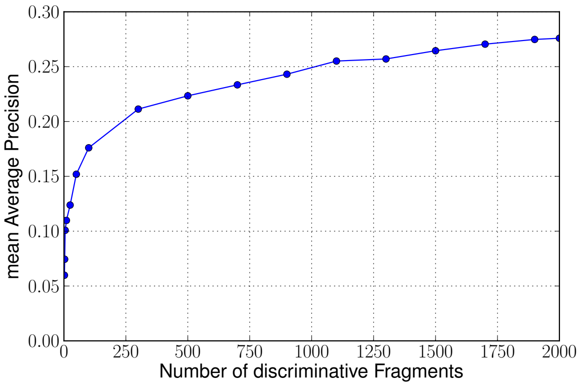


Figure 5: The influence of the number of discriminative fragments for event detection.

Table 1 also shows the result of VideoStory on the same dataset [8]. Although VideoStory similarly improves upon the global baseline, it does not match the bag-of-fragments result, indicating the effectiveness of the bag-of-fragments.

To further highlight the effectiveness of fragment-based event detection, we show a fragment ranking for three events in Fig. 1. The Figure shows that the most discriminative fragments of an event are exemplary snapshots of the event. Examples of this include a person on a BMX for *Attempting a bike trick*, a bicycle tire for *Non-motorized vehicle repair*, and a person on a horse for *Horse riding competition*. Also, the Figure shows that more ambiguous fragments are deemed less discriminative for the event.

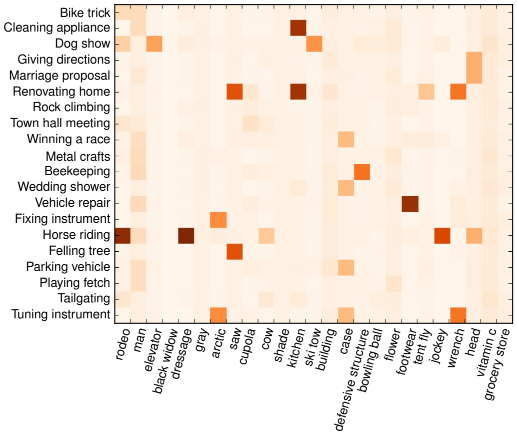


Figure 6: Plot of the maximum co-occurrence value for 25 concepts with respect to the 20 TRECVID MED 2014 events.

5.3 Event recounting

For event recounting, we first highlight the effect of co-occurrence for a number of concepts in the semantic representation. In Fig. 6, the maximum co-occurrence values are shown for 25 concepts with respect to all 20 TRECVID MED 2014 events. For a number of peaks in the plot, the concept-event relation is as expected. Examples include the relationship between rodeo/jockey and *Horse riding competition*, between kitchen and *Cleaning an appliance*, and between saw and *Felling a tree*. These discovered concept-event relationships indicate that visual co-occurrence from social tagged image data may serve as a fruitful proxy for automatic concept selection.

However, Fig. 6 also indicates a limit of our use of co-occurrence. The concept elevator fires on the event *Dog show*, while they are seemingly uncorrelated. However, the textual summary of the event contains the concept lift. In the context of *Dog show*, lift is a verb (to lift; to move upward), but the co-occurrence statistics use the concept as a noun (lift; elevator). As the co-occurrence statistic is oblivious to the ambiguity of concepts, seemingly uncorrelated concepts might yield a high co-occurrence value.

Second, we perform a qualitative evaluation on the effect of concept filtering for event recounting. In Fig. 7, the recounting results of our discriminative fragments are shown for three videos from different events, both with and without concept selection. For each video, the three most informative video fragments are selected and for each selected fragment, the two most informative concepts are shown. The discriminative fragments are able to select the fragments of the test video that are correlated to the event. Without the concept filtering, the recounted concepts are at times incorrect or over-specific. This is exemplified in Fig. 7a and Fig. 7b, with noisy concepts such as millipede and abacus for *Tuning an instrument* and guillotine for *Renovating a home*. These results indicate the negative influence of the noise, as these concepts do not help to convince a user that the corresponding event is in the video.

However, if our concept selection is added to the recounting, the resulting concepts become both more generic and more relevant. This is for example visible in Fig. 7c. Without concept filtering, incorrect concepts such as dulcimer and wildcat are recounted. Upon adding the concept selection, concepts that are more relevant to the event *Beekeeping* are shown, such as honeycomb.

6. CONCLUSIONS

We propose encoding of videos using fragments. We show how to generate a concise set of fragment proposals for a single video. From the set of fragment proposals of the training videos for an event, we select the most discriminative ones. By matching and pooling these discriminative fragments over the fragment proposals of a video, we arrive at our bag-of-fragments encoding. Experimental evaluation shows the effectiveness of the encoding for event detection, as well as its complementary nature to a global aggregation of semantic concepts. Furthermore, we propose an automatic algorithm to filter relevant concepts in the semantic representation with respect to an event by leveraging co-occurrence statistics from a social tag image dataset. Qualitative evaluation highlights the capability of our bag-of-fragments in combination with concept filtering for event recounting.

Acknowledgements

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

7. REFERENCES

[1] S. Bhattacharya, F. X. Yu, and S.-F. Chang. Minimally needed evidence for complex event recognition in unconstrained videos. In *ICMR*, 2014.

[2] C. Cortes and V. Vapnik. Support-vector networks. *ML*, 20(3), 1995.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] D. Ding, F. Metze, S. Rawat, P. F. Schulam, S. Burger, E. Younessian, L. Bao, M. G. Christel, and A. Hauptmann. Beyond audio and video retrieval: towards multimedia summarization. In *ICMR*, 2012.

[5] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, 2013.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2), 2010.

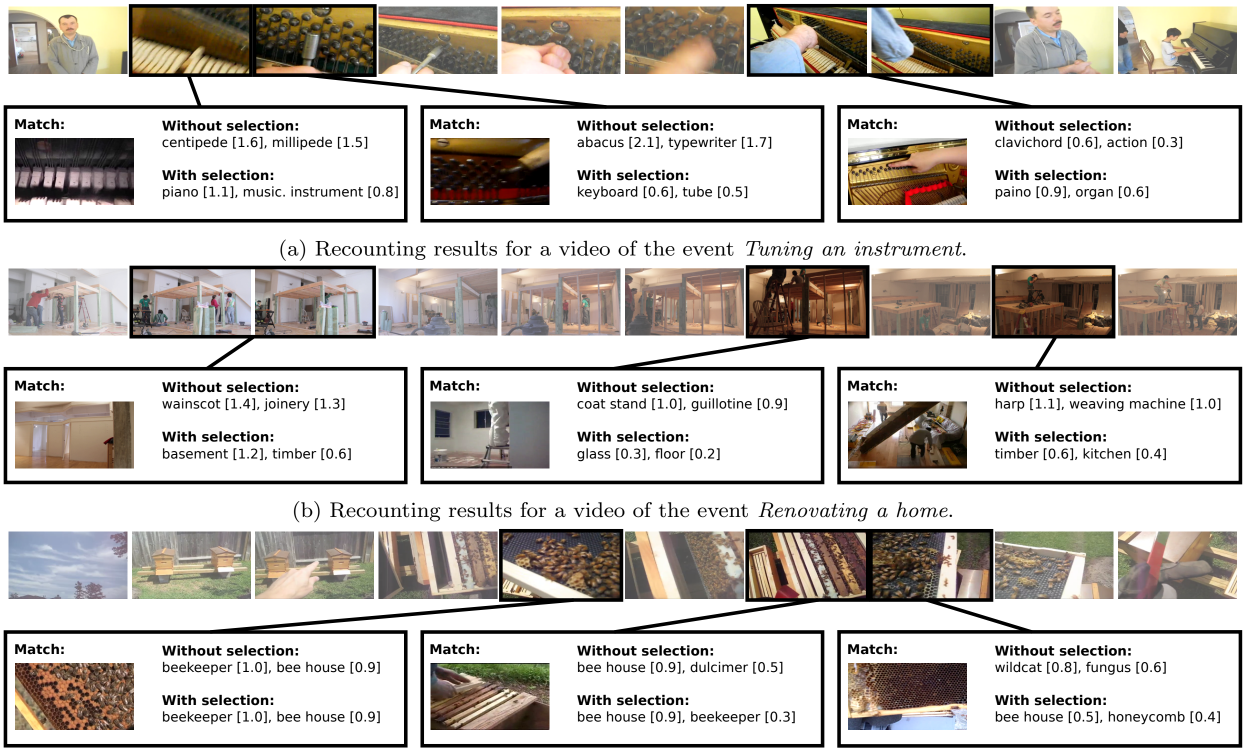
[7] I. Everts, J. C. van Gemert, and T. Gevers. Evaluation of color spatio-temporal interest points for human action recognition. *TIP*, 23(4), 2014.

[8] A. Habibian, T. Mensink, and C. G. M. Snoek. Videostory: A new multimedia embedding for few-example recognition and translation of events. In *ACM MM*, 2014.

[9] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012.

[10] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *BMVC*, 2014.

[11] M. Jain, J. C. Van Gemert, H. Jégou, P. Bouthemy, and C. G. M. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014.


Match:
Without selection:
centipede [1.6], millipede [1.5]
With selection:
piano [1.1], music. instrument [0.8]
Match:
Without selection:
abacus [2.1], typewriter [1.7]
With selection:
keyboard [0.6], tube [0.5]
Match:
Without selection:
clavichord [0.6], action [0.3]
With selection:
paino [0.9], organ [0.6]
Match:
Without selection:
wainscot [1.4], joinery [1.3]
With selection:
basement [1.2], timber [0.6]
Match:
Without selection:
coat stand [1.0], guillotine [0.9]
With selection:
glass [0.3], floor [0.2]
Match:
Without selection:
harp [1.1], weaving machine [1.0]
With selection:
timber [0.6], kitchen [0.4]
Match:
Without selection:
beekeeper [1.0], bee house [0.9]
With selection:
beekeeper [1.0], bee house [0.9]
Match:
Without selection:
bee house [0.9], dulcimer [0.5]
With selection:
bee house [0.9], beekeeper [0.3]
Match:
Without selection:
wildcat [0.8], fungus [0.6]
With selection:
bee house [0.5], honeycomb [0.4]

(a) Recounting results for a video of the event *Tuning an instrument.*

(b) Recounting results for a video of the event *Renovating a home.*

(c) Recounting results for a video of the event *Beekeeping.*

Figure 7: Qualitative event recounting results with and without the concept filtering for three different videos. For each video, the top three fragments are selected and the top two concepts for each selected fragment are shown, along with their fragment-based score. For each recounted fragment, the matching discriminative fragment from the training set is shown.

[12] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. In *ECCV Workshop*, 2014.

[13] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013.

[14] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[15] J. Liu, Q. Yu, O. Javed, S. Ali, A. Tamrakar, A. Divakaran, H. Cheng, and H. Sawhney. Video event recognition using concept attributes. In *WACV*, 2013.

[16] M. Mazloom, A. Habibian, and C. G. M. Snoek. Querying for video events by semantic signatures from few examples. In *ACM MM*, 2013.

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

[18] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev. Semantic model vectors for complex video event recognition. *TMM*, 14(1), 2012.

[19] D. Oneata, J. Verbeek, and C. Schmid. The lear submission at thumos 2014. In *ECCV Workshop*, 2014.

[20] P. Over et al. Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID Workshop*, 2014.

[21] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, 1999.

[22] C. Sun, B. Burns, R. Nevatia, C. G. M. Snoek, B. Bolles, G. Myers, W. Wang, and E. Yeh. Isomer: Informative segment observations for multimedia event recounting. In *ICMR*, 2014.

[23] C. Sun and R. Nevatia. Discover: Discovering important segments for classification of video events and recounting. In *CVPR*, 2014.

[24] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2), 2013.

[25] A. Vahdat, K. Cannons, G. Mori, S. Oh, and I. Kim. Compositional models for video event detection: A multiple kernel learning latent variable approach. In *ICCV*, 2013.

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

[27] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A formal study of shot boundary detection. *TCSVT*, 17(2), 2007.

[28] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.