Video Stream Retrieval of Unseen Queries using Semantic Memory

Spencer Cappallo
cappallo@uva.nl

Thomas Mensink
tmensink@uva.nl

Cees G. M. Snoek
cgmsnoek@uva.nl

University of Amsterdam
Science Park 904
Amsterdam
The Netherlands

Abstract

Retrieval of live, user-broadcast video streams is an under-addressed and increasingly relevant challenge. The on-line nature of the problem requires temporal evaluation and the unforeseeable scope of potential queries motivates an approach which can accommodate arbitrary search queries. To account for the breadth of possible queries, we adopt a no-example approach to query retrieval, which uses a query's semantic relatedness to pre-trained concept classifiers. To adapt to shifting video content, we propose memory pooling and memory welling methods that favor recent information over long past content. We identify two stream retrieval tasks, instantaneous retrieval at any particular time and continuous retrieval over a prolonged duration, and propose means for evaluating them. Three large scale video datasets are adapted to the challenge of stream retrieval. We report results for our search methods on the new stream retrieval tasks, as well as demonstrate their efficacy in a traditional, non-streaming video task.

1 Introduction

This paper targets the challenge of searching among live streaming videos. This is a problem of increasing importance as more video content is streamed via services like Meerkat, Periscope, and Twitch. Despite the popularity of live streaming video, searching in its content with state-of-the-art video search methods, *e.g.* [11, 18, 25, 34], is nearly impossible as these typically assume the *whole* video is available for analysis before retrieval. We propose a new method that can search across live video streams, for any query, without analyzing the entire video.

In live video, the future is unknowable thus one only has access to the past and present. It is therefore crucial to leverage knowledge of the (recent) past appropriately. Memory can be modeled with the aid of hidden Markov models or recurrent neural networks with long-short term memory. Through the ability to selectively remember and forget, recurrent neural networks have recently shown great potential for search in videos *e.g.* [5, 22]. Inspired by the success of supervised memory models, we propose a mechanism to incorporate memory and forgetting in video stream retrieval without learning from examples.

© 2016. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

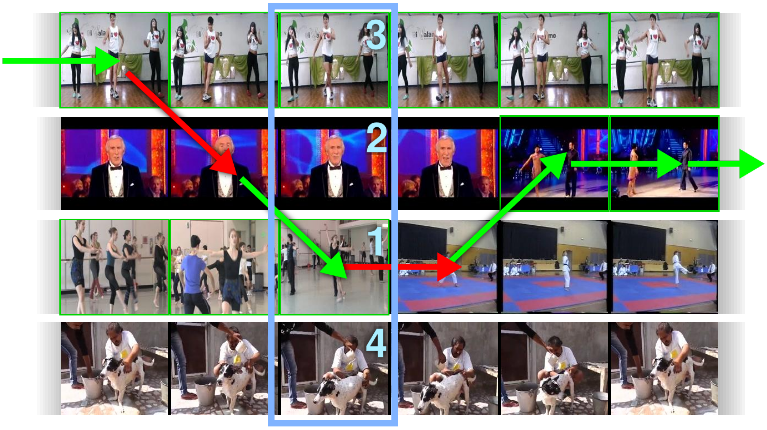


Figure 1: Two video stream retrieval tasks: Instantaneous retrieval returns a ranked list of currently relevant streams, while continuous retrieval seeks to maximize the time spent on relevant streams, while minimizing the number of changes between streams.

Our search mechanism is founded on recent work in zero-shot classification, *e.g.* [10, 23, 33]. These methods all use an external linguistic corpus to pre-train a semantic embedding such that terms which are used in similar contexts have similar vectors within the embedding [19]. Video frames are fed to a pre-trained convolutional neural network that outputs concept classification scores, which, after pooling over the entire video, can be projected into the embedding. An incoming text query utilizes the embedding to find the best matching videos. We adopt this established approach from the zero-shot community and re-purpose it for the new problem of live video stream retrieval.

We make three contributions. First, we establish the new problem of video stream retrieval and introduce a solution based on a framework popularized for zero-shot classification. Second, we introduce several methods to base retrieval on only the recent memory of the streams. Finally, in absence of any stream retrieval tasks in leading benchmarks such as ActivityNet [8] and NIST TRECVID [25], we propose two evaluation settings (see Fig. 1) based on publicly available datasets [8, 13] and compare against established baselines. We also demonstrate that our method excels in more traditional whole-video retrieval scenarios.

2 Related Work

The setting of live stream retrieval is inherently related to a wide gamut of video tasks. In this section we discuss some of the most relevant related work.

Video Concept Detection Video retrieval is aided by knowledge of the visual concepts which compose a scene and whose interaction through time define actions and events [21, 28, 29]. Concept detection in video has been primarily addressed in the context of supervised classification tasks to detect objects, actions, and events, where the entire video is available for processing. Most state-of-the-art approaches represent a video by pooling per-frame features extracted using a pre-trained convolutional neural network (CNN) [12, 20, 34]. Such an approach is good for shorter, single-topic videos where the semantics of all frames are important for the final prediction, such as for events, actions, and activities [8, 25]. In a stream retrieval setting, however, the gestalt of the video in its totality becomes less important than what is currently happening in the stream. Our approach instead emphasizes only recent information, by discarding past information which cannot be guaranteed to be relevant to the current stream content.

More explicit modelling of the temporal qualities of video has taken several forms, from temporal features, such as motion boundary histograms [32], to learning recurrent neural networks [2, 5, 14, 22], and to localising the temporal extent of actions [7, 9]. The idea of temporal windows have also been used to perform temporal action localisation in a hierarchical manner [24], which is orthogonal to any on-line stream processing concerns.

Zero-shot prediction Live stream retrieval is a compelling use case for zero-shot prediction, given that the future content can not be predicted and accompanying descriptions can not be guaranteed. Zero-shot classification seeks to transfer the models learned on one set of classes to another, related class through some intermediary knowledge source [1, 10, 15, 16, 17, 23]. Most of these methods focus on a limited semantic transfer, *e.g*. from a known set of animals to another set of animals; and trained on images and tested on images. The yearly TRECVID MED zero-example benchmark [25] has transferred this problem to event retrieval among videos, where a long, unseen textual event description is used to retrieve web videos. The scope of potential class types is also restricted in this case, and most participants use a fusion of aggregated event-related video features [4, 27, 33].

One example of wider semantic transfer is the work of Jain *et al*. [10]. Jain *et al*. exploit pre-trained ImageNet object detectors and an externally trained semantic embedding to transfer knowledge of ImageNet objects to actions and events in videos. This semantic embedding is constructed such that terms which are used in similar contexts have similar vectors within the embedding. Concept scores from a deep neural network trained to predict ImageNet classes are related to unseen concepts within the embedding space. Due to the encompassing nature of a broad linguistic corpus, this particular approach has been demonstrated to be useful for classifying a wide range of class abstractions, including objects [23], actions, events [10], and emoji [3]. Such an approach is well-suited to the problem of live video stream retrieval, where possible queries may include these and many other class types.

3 Video Stream Retrieval

We focus on the novel problem of video stream retrieval. The nature of live, user-broadcast video has two major implications. First, the full range of potential future queries cannot be known, necessitating the ability to respond to unanticipated queries. Second, the future content of live video is unknown, and might not relate to prior content within the same stream, therefore we propose several methods to emphasize recent stream content.

3.1 Ranking Unanticipated Queries

The goal is to retrieve relevant streams for a provided textual query q. To be robust against unanticipated queries, we follow a zero-shot classification paradigm [3, 10, 23]. A deep neural network trained to predict image classes is applied to the frames of the video stream as a feature extractor. x_t represents the softmax output of the deep network across the output classes C for a frame at time t. Some $\phi(x_t)$ encodes these concepts in a sparse manner. Both the concepts C as well as the query q are placed in a mutual embedding space (in our case,

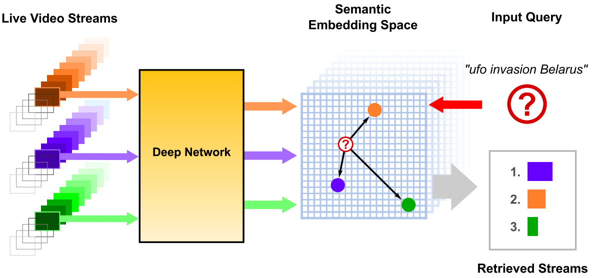


Figure 2: Stream retrieval for any query: Live streams are encoded by concept confidence scores, using a deep network. Streams are ranked based on the similarity between the query and these scores in a semantic space.

we use word2vec [19]), and video steams are scored based on the cosine similarities, using:

$$\text{score}(q, x_t) = s(q)^\intercal \phi(x_t) \tag{1}$$

where $s(q)$ returns a vector containing the cosine similarities between the embedding representation of the query q and those of the concepts C. If the query q comprises multiple terms, we use the mean of the per term scores, $s(q) = \frac{1}{N}\sum_{i=0}^{N} s(q_i)$, which has been shown to hold semantic relevance [19]. Fig. 2 shows the retrieval process.

3.2 Memory for Stream Retrieval

We introduce the notion of a "memory" for the problem of video stream retrieval, which aims to exploit recent information while limiting the effect of possibly irrelevant past information. The variable nature of live video means that past information might not be informative for future predictions, as a stream's content can change drastically. It is necessary to balance the utility of past information against the risk that it is no longer pertinent. In this section, we describe three approaches which use such a memory.

3.2.1 Memory Pooling

Temporal pooling of frame-based features or concepts over an entire video is used in state-of-the-art approaches for standard video retrieval tasks [25]. This strategy could be adapted to an on-line setting by pooling among all frames from time $t = 0$ to the present. However, this introduces problems when the content of a stream changes, which is a particular concern with longer streams. For this reason, we pool instead over a fixed temporal memory m, which is tethered to the present and offers a restricted view on the past:

$$MP_{max}(x_t) = \max_{i=t-m}^{t} x_i \qquad\qquad MP_{mean}(x_t) = \frac{1}{m}\sum_{i=t-m}^{t} x_i \tag{2}$$

where x_t denotes the features at time t, and we evaluate max pooling or mean pooling, denoted as MP_{max} and MP_{mean} respectively. At the start of a stream, when $m < t$, we instead use $m = t$. We set the memory duration m through validation on a small set of queries which are disjoint from the test queries. The contribution of low confidence concepts introduces noisy predictions and influences the retrieval performance, therefore we use only the highest-valued pooled concepts, as proposed in [23].

While mean and max pooling can be computed efficiently across all frames since $t = 0$ in an iterative way, the introduction of a memory requires the storage of m previous frames' worth of features for every concurrent stream. In part motivated by this expense, we introduce an alternative method which can be calculated in a stateless manner.

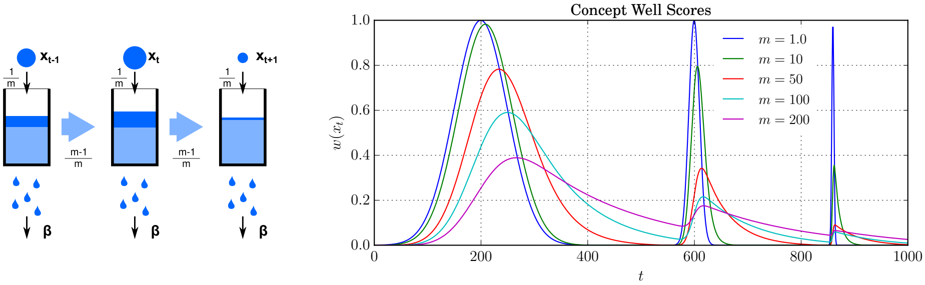


Figure 3: Left: Illustration of a memory well. New concept scores flow into the well, while old information gradually leaks out. Right: The effect of the memory parameter *m* on memory welling. $m = 1$ corresponds to the raw classifier confidence scores. Larger *m* values result in a well which empties more slowly, but which is less responsive to sudden spikes.

3.2.2 Memory Welling

The need to capture both long term trends and short-duration confidence spikes motivates the development of what we term *memory wells*. In these wells, observations flow into the well at every timestep, but the well also leaks at every timestep, as illustrated in Fig. 3. In contrast to the memory pooling, where all observations are weighed equally and observations beyond the memory horizon are lost, the impact of past observations on memory wells instead diminishes steadily over time. Memory wells are defined in the following manner:

$$w(x_t) = \max\left(\frac{m-1}{m} w(x_{t-1}) + \frac{1}{m} x_t - \beta, 0\right), \tag{3}$$

where the current value of a well relies on the well's value at time $t-1$, diminished by a tunable memory parameter m and a fixed constant leaking term β. We illustrate the effect of m in Fig. 3. Note that m in this formulation is somewhat different from that in the memory pooling approach, albeit both aim to tune the contribution of past frames. Memory wells bear a faint resemblance to stacks or queues, but are distinguished by being unordered aggregrations of continuous values rather than ordered collections of discrete items, and by their discarding of stale data over time through leakiness.

The β term creates sparseness in the representation, which ensures that only recent or consistently present concepts are used for prediction. We fix $\beta = \frac{1}{C}$, where C is the number of concepts, as this is a lower bound for a concept being present at time t. This is the value that the classifier would output for every concept if it considered them all equally likely to be present in the current frame. Enforcing sparseness, or rather, enforcing reliability of concept scores, means that the memory well values can be used directly in Equation 1, without the need to arbitrarily select some number of the highest-confidence concepts.

Max Memory Welling In the case of short streams and traditional video processing tasks, which are likely to have more consistent content, the short-term nature of memory welling can be a limitation, even if its properties are still effective for improving temporally local predictions. Memory welling can be adapted to this task through temporal max pooling across the query scores per stream:

$$\text{score}(q, x_t) = \max_{i=0}^{t} \left(s(q)^\intercal w(x_i) \right) \tag{4}$$

This exploits temporally local, high confidence predictions from the welling approach, which might be averaged away in traditional whole video pooling. It is well-suited to single-topic content such as short streams and traditional, full video retrieval tasks.

Computational Complexity The proposed approach for stream retrieval, particularly with memory welling, is comparatively lightweight, which is important for the targeted setting. Semantic similarity values, $s(\cdot)$, can be pre-computed and hashed for the entire query vocabulary, therefore $s(q)$ scales at $\mathcal{O}(l)$, where l is the number of terms in the query q. Calculating $s(\cdot)^\intercal w(x_t)$ depends on the number of concepts, therefore has a complexity of $\mathcal{O}(m)$ for one stream, where m is the number of concepts. Across n streams, this gives a total complexity of $\mathcal{O}(lmn)$. x_t has a constant cost per frame, which on a modern GPU is below 80ms per batch of 128 frames. As l and m are fixed and relatively small constants, real time stream retrieval with the proposed method and a reasonable sampling rate is achievable.

4 Tasks for Video Stream Retrieval

To reflect the on-line nature and the diverse applications of video stream retrieval we propose two evaluation settings: *i*) Instantaneous Retrieval, which measures the retrieval performance at any given time t; and *ii*) Continuous Retrieval, where a succession of streams relevant to a single query are retrieved over a prolonged duration.

4.1 Instantaneous Retrieval

The goal of instantaneous retrieval is to retrieve the most relevant stream for a query q at any arbitrary time t. This temporal assessment is important, given that a model which only performs well when a stream has ended is useless for discovery of live video streams.

To incorporate the temporal domain, we use the mean of the average precision (AP) scores per time step t, which we coin Temporal Average Precision (TAP). Letting AP_t denote the AP score for some query at time t, the TAP then corresponds to the mean AP_t across all times for which there is at least one relevant stream:

$$\text{TAP} = \frac{1}{\sum_t y^t} \sum_t \text{AP}_t \cdot y^t, \tag{5}$$

where y^t indicates whether there is at least one relevant stream for the query at time t.

4.2 Continuous Retrieval

The goal of the continuous retrieval task is to maximize the fraction of time spent watching relevant streams, while minimizing the number of times the stream is changed. Consider a viewer searching for coverage of the Olympics. When one stream stops showing the Olympics, she wants to switch to another stream showing the Olympics. However, switching between two streams every second, even if both relevant, provides a poor viewing experience.

To evaluate this scenario, we consider the number of *zaps*. A zap is any change in the retrieved stream or its relevancy, including the move at time $t = 0$ to the first retrieved stream.

We distinguish good zaps, which is any zap that moves from a currently irrelevant stream to a currently relevant stream, from all other (bad) zaps. The count of good zaps and bad zaps are represented by z_+ and z_-.

The fraction of good zaps to total zaps, $\frac{z_+}{z_+ + z_-}$, describes the average quality of individual changes, but offers an incomplete picture of the system's temporal consistency. Imagine a system which only ever retrieves one stream, which is initially relevant but quickly turns and remains irrelevant. Despite its performance, this would achieve a score of 0.5, as it would have had one good zap over a total of two zaps. To incorporate overall accuracy over time, we also reward an algorithm choosing to correctly remain on a relevant stream. Letting r_+ track the number of times an algorithm remains on relevant stream, the *zap precision* ZP is

$$\text{ZP} = \frac{z_+ + r_+}{\sum_t y^t} \tag{6}$$

where y^t again represents whether or not there is at least one relevant stream at time t.

5 Experiments

5.1 Setup

Datasets We evaluate our methods on three large scale video datasets: i) ActivityNet [8] (AN), a large action recognition dataset with 100 classes and 7200 labeled videos. Performance is evaluated on a test set composed of 60 classes randomly selected from the combined ActivityNet training and validation splits, and a validation set of the other 40 classes is used for parameter search; ii) A subset of the Fudan-Columbia Videos [13] (coined FCVS), composed of 25 videos for each of the 239 classes making up 250 hours of video, which we split into a validation set of 50 classes and a test set of 179 classes. FCVS annotations are more diverse (objects, locations, scenes, and actions), but lack temporal extent, so a class is assumed to be relevant for the duration of a video; iii) TRECVID MED 2013 [25] (MED), an event recognition dataset, used to evaluate the efficacy of our memory-based approach against published results. To facilitate comparison, the setting used by [10] is replicated: whole-video retrieval using only the event name.

In addition to evaluating on short web videos themselves, we introduce AN-L and FCVS-L, which are adaptations to simulate longer streams with varied content. To accomplish this, individual videos are randomly concatenated until the simulated stream is at least 30 minutes long. Annotations from the original videos are propagated to these concatenated videos. Details of the data set splits will be made available to allow future comparison[1].

Features We sample videos at a rate of two frames per second. Each frame is represented by the softmax confidence scores of 13k ImageNet classes, which are extracted using a pre-trained deep neural network from [18]. The network was trained on ImageNet [26] and its structure is based on the GoogLeNet network [30]. Our semantic embedding is a 500-dimensional skip-gram word2vec [19] model trained on the text accompanying 100M Flickr images [31], similar to the one used in [3, 10].

Evaluation and Baselines To simulate the streaming setting, performance is evaluated sequentially across all videos, using only the present and past frames. Results are reported in the previously described TAP and ZP metrics averaged over all test classes. For the memory

[1] http://staff.science.uva.nl/s.h.cappallo/data.html

Table 1: Results of instantaneous and continuous retrieval across all datasets and tasks. $m = 1$ corresponds to using only the current frame, while $m = t$ means that pooling is performed over all past and present frames. Memory welling offers the best performance flexibility.

	Instantaneous (% TAP)				**Continuous (% ZP)**	
	AN	**FCVS**	**AN-L**	**FCVS-L**	**AN-L**	**FCVS-L**
Random	1.4	4.9	3.6	2.9	1.3	1.1
Mean Memory Pooling						
$m = 1$	16.9	21.4	25.1	24.8	21.9	21.6
$m = t$	18.4	30.7	8.5	9.3	5.9	6.3
$m = m^*$	21.7	28.8	29.3	30.0	27.5	27.7
Max Memory Pooling						
$m = t$	20.0	27.4	9.0	9.5	5.9	6.0
$m = m^*$	21.0	27.5	29.7	30.3	27.3	27.5
Memory Welling	22.5	30.5	**30.1**	**30.6**	**28.3**	**28.4**
Max Memory Welling	**24.6**	**35.9**	11.0	15.9	5.6	10.9

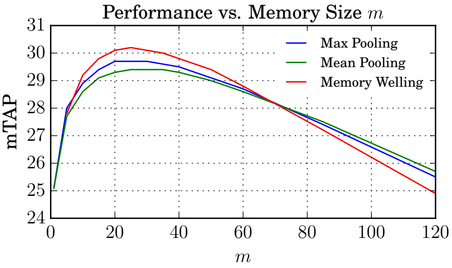


Figure 4: Effect of m parameter on the ActivityNet-Long (AN-L) dataset. All approaches share a similar dependence, with a peak around $m = 25$, which corresponds to 12.5 seconds at our sampling rate. Past this point, the irrelevancy of past information becomes overpowering.

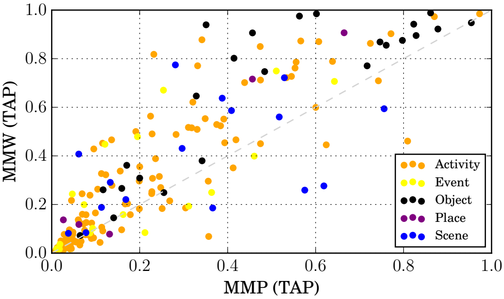
based methods, the optimal value of $m = m^*$ is determined on the validation set containing videos of classes not present in the test set. The two extremes of memory pooling are used as baselines: $m = 1$, which simply relies on the current frame of a video to make a prediction; and $m = t$, which corresponds to pooling over the entirety of the stream up to the present time, similar to whole-video pooling used in video retrieval scenarios [25]. This approach has also been explored as a basis for whole video action recognition [6].

5.2 Instantaneous and Continuous Stream Retrieval

We first compare our proposed methods and baselines on the instantaneous and continuous stream retrieval tasks. Table 1 shows the results for the two tasks. In general, we observe that memory-based approaches shine when query relevance is temporally limited, as in the AN, AN-L, and FCVS-L datasets. For a setting like FCVS, where a single annotation covers an entire stream, the baselines become more competitive. In a scenario where streams are guaranteed to be short in duration and focus on a single topic, then a max memory welling approach makes the most sense. For streams of indeterminate length and content, the memory welling approach offers the best results and flexibility to cover any situations that may arise. In the continuous retrieval setting, the $m = t$ baselines and max memory welling perform poorly, likely due to their inability to respond quickly to changes in stream content.

Category	**MMW**	**MMP**
Art	20.4	14.7
Leisure & Tricks	34.0	24.5
Nature	64.6	55.2
Travel	31.3	30.0
Everyday Life	31.2	21.0
Sports	48.5	32.6
Beauty & Fashion	24.3	17.1
Music	35.7	28.8
DIY	16.9	13.1
Education & Tech	67.8	51.4
Cooking & Health	27.7	20.9
Annotation Type	**MMW**	**MMP**
Place - Particular location	39.1	26.8
Object - Thing or creature	67.1	50.0
Scene - Generic scene setting	39.4	33.3
Event - Particular occurrence	28.5	21.1
Activity - Human activities	30.3	22.2

Table 2: Instantaneous retrieval on FCVS by annotation category and type for Max Memory Welling and Max Memory Pooling. The query type significantly affects the retrieval quality, but welling yields improvement in all cases. Below: Per-class scatterplot comparison of the results.



Method	**mAP (%)**	**mTAP**
Chen *et al.* [4]	2.4	
Wu *et al.* [33]	3.5	
Jain *et al.* [10]	3.5	
Jain *et al.* (our features)	3.5	9.2
Max Memory Welling	**4.7**	**17.8**

Table 3: Performance of Max Memory Welling on the TRECVID2013 MED task. MMW outperforms the state-of-the-art on this task. As [10] uses a different deep network, we also verify their results with our features. Further, we compare the performance of such an approach on instantaneous retrieval.

Impact of Memory Length The impact of the m parameter on instantaneous retrieval is shown in Figure 4. The response of memory-based approaches to changing content degrades if m is too large, and its resistance to noisy spikes suffers if m is too small. m values between 15 and 35 appear to be most adequate for the identification of current content.

Per-Category Performance For the FCVS dataset, we report the performance per category and per annotation type in Table 2. The categories are provided within the annotation hierarchy, while we have manually assigned the FCVS test classes to one of five types. The Nature, Education & Tech, and Sports categories perform strongly, likely due to their domain similarity with the ImageNet concepts used to train the deep network. This is also illustrated by the strong performance of the Object type classes. Meanwhile, the Art and DIY categories perform very poorly. The videos within these categories depict many hard-to-distinguish activities. For example, DIY contains four different classes which are composed primarily of video of hands manipulating paper. This very similar visual content is challenging. Furthermore, Events and Activities prove difficult to retrieve, likely due to their reliance on time. This highlights the difficulty of representing queries with an intrinsic temporal element through constituent static image concepts (such as ImageNet concepts).

5.3 Whole Video Retrieval

To compare our method against published results, we report mAP results on the MED dataset, following the setting from [10]: multimedia event retrieval based solely on the event-name. The results are shown in Table 3, where we also compare to the visual-only results of [4, 33]. Our Max Memory Welling outperforms these methods, while being on par with the more advanced Fisher Vector event-name encoding (4.2 % mAP) of [10]. Note, such an event-name encoding could also be used alongside our method. The Max Memory Welling approach is able to leverage short-term, high-confidence predictions generated through memory welling, which is useful for whole video retrieval.

6 Conclusion

The retrieval of live video streams requires approaches which can respond to unanticipated queries. We present such an approach, and demonstrate the importance and utility of memory-based methods, such as memory welling, for both on-line stream retrieval and other zero-example video tasks. We explore two scenarios, instantaneous and continuous retrieval, that follow naturally from the problem of stream retrieval, and offer an approach for evaluating these scenarios on existing, abundant large scale video datasets.

Acknowledgements

This research is supported by the STW STORY project and the NWO VENI What&Where project.

References

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

[2] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *ICANN*. 2010.

[3] S. Cappallo, T. Mensink, and C. G. M. Snoek. Query-by-emoji video search. In *MM*, 2015.

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

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[6] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.

[7] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal Localization of Actions with Actoms. *TPAMI*, 35(11), 2013.

[8] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

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

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

[11] L. Jiang, S.-I. Yu, D. Meng, Y. Yang, T. Mitamura, and A. G. Hauptmann. Fast and accurate content-based semantic search in 100m internet videos. In *MM*, 2015.

[12] Y.-G. Jiang, X. Zeng, G. Ye, S. Bhattacharya, D. Ellis, M. Shah, and S.-F. Chang. Columbia-ucf trecvid2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching. In *TRECVID*, 2010.

[13] Y. G. Jiang, Z. Wu, J. Wang, X. Xue, and S. F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *arXiv preprint arXiv:1502.07209*, 2015.

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

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

[16] D. Mahajan, S. Sellamanickam, and V. Nair. A joint learning framework for attribute models and object descriptions. In *ICCV*, 2011.

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

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

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

[20] M. Nagel, T. Mensink, and C. G. M. Snoek. Event fisher vectors: Robust encoding visual diversity of visual streams. In *BMVC*, 2015.

[21] A. P. Natsev, M. R. Naphade, and J. Tešić. Learning the semantics of multimedia queries and concepts from a small number of examples. In *MM*, 2005.

[22] J. Y. H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

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

[24] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013.

[25] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, G. Quénot, and R. Ordelman. Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2015.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.F. Li. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3), 2015.

[27] B. Singh, X. Han, Z. Wu, V. I. Morariu, and L. S. Davis. Selecting relevant web trained concepts for automated event retrieval. In *ICCV*, 2015.

[28] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR*, 2006.

[29] C. G. M. Snoek and M. Worring. Concept-based video retrieval. *FnTIR*, 4(2), 2009.

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

[31] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. J. Li. YFCC100M: The new data in multimedia research. *CACM*, 59(2), 2016.

[32] H. Wang, A. Kläser, C. Schmid, and C. L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1), 2013.

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

[34] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *CVPR*, 2015.