Lifelog Scene Change Detection Using Cascades of Audio and Video Detectors

Katariina Mahkonen, Joni-Kristian Kämäräinen, Tuomas Virtanen

Department of Signal Processing
Tampere University of Technology
Finland

Abstract. The advent of affordable wearable devices with a video camera has established the new form of social data, lifelogs, where lives of people are captured to video. Enormous amount of lifelog data and need for on-site processing demand new fast video processing methods. In this work, we experimentally investigate seven hours of lifelogs and point out novel findings: 1) audio cues are exceptionally strong for lifelog processing; 2) cascades of audio and video detectors improve accuracy and enable fast (super frame rate) processing speed. We first construct strong detectors using state-of-the-art audio and visual features: Mel-frequency cepstral coefficients (MFCC), colour (RGB) histograms, and local patch descriptors (SIFT). In the second stage, we construct a cascade of the trained detectors and optimise cascade parameters. Separating the detector and cascade optimisation stages simplify training and results to a fast and accurate processing pipeline.

1 Introduction

Wearable devices with a video camera, such as Google Glasses, are becoming commodity hardware and it seems that consumers are willing to push personal blogs even further: video and audio logging of their lives from the first-person view (Figure 1), *lifelogs*. The lifelog applications have recently become under active investigation, but it is still unclear how to adopt and adapt the existing video processing techniques. In addition, a huge number of video streams and on-device processing require computationally economic but fast methods.

One important application for lifeloggers is to automatically annotate important moments which can be quickly stored, indexed and shared. This can be achieved by condensing important moments into a short "skim" and thus research on video skimming (summarisation/abstraction) has recently gained momentum [1, 2]. The best skimming methods are too heavy for on-device processing, but their core components, such as *scene detection* which produces the smallest data pieces for skimming, scenes, may be doable. On-device scene change detection can help fast (on-line) generation of summaries. State-of-the-art scene detection methods rely on visual information, but another important cue, audio, provides an alternative modality with orders of magnitude faster processing. Visual and audio cues are often complementary and therefore many hybrids have

Fig. 1. Video frames from our CASA2 lifelog dataset.

been proposed [3, 4]. However, these works mainly concentrate on maximising accuracy and omit potential for faster computation.

What is the best approach to combine detectors using features of varying importance and even from different modalities? In machine learning literature, a particularly suitable technique for cost (computation time) sensitive learning are detector cascades introduced by Viola and Jones [5]. State-of-the-art cascade construction methods do not operate on stages [5], but simultaneously optimise the whole cascade and its parameters using all training data at once [6–8]. That sets restrictions on used detectors while in our case they can be very different and therefore joint optimisation of the methods, cascade structure and its parameters is too complicated and slow, even impossible. In this work, we take a novel approach: we adopt the cascade structure but cast the problem as a classifier combination [9]: the detectors are trained separately as "strong detectors", then cascaded based on their complexity (audio detectors first) and finally the cascade parameters are optimised similar to expert weights in [9]. Our relaxed design results to simpler cascade construction and training, allows using pre-trained detectors by others, and still our "soft cascades" achieve fast (super frame rate) processing and superior accuracy.

2 Related Work

Detector Cascades - Viola and Jones [5] is the seminal work introducing cascades as a machine learning approach to tackle the real-time requirement in face detection. Their method operated on stages each aiming at high recall and false positives passed to the next more complicated classification stage. The approach is effective but sub-optimal and recent holistic approaches optimising detectors, cascade structure and cascade parameters simultaneously with all training data can provide better cascades [6–8]. That, however, sets requirements for the detectors which we wish to avoid in our work to be able to to exploit the best available detectors. We use all training data to train a set of binary detectors, we combine them using a free-form logical rule (a fixed cascade structure) and then optimise cascade parameters by exhaustive or beam search. In that sense our model is close to combining classifiers theory [9] adapted to cascades.

Combining Audio and Video Cues - Many novel video applications based on visual features have been proposed [10, 11], but combinations of audio and visual features seem always superior [3, 4, 12]. In contrast to the previous works, we do not explicitly engineer a combined audio-video-detector, but take a set of detectors, train them separately for the specific task, and then construct and optimise a cascade structure. That is also justified by the fact that lifelog data is different from the previously used full length movies [13], TV news [14], filmstrips [3] or the mixture TrecVid data [15]. The lifelog data is raw, abruptly moving, unedited, first-person-shot video.

Contributions – Our contributions in this work are two-fold: 1) we investigate how existing visual and audio processing methods work on lifelog data and 2) introduce "soft cascades" of strong detectors for efficient scene change detection in super frame rate. We have collected seven hours of real lifelog data and in our cascades we utilise the best performing cues from various studies: colour (RGB) histograms [16], local image patch descriptor (SIFT) based visual bag-of-words [17] and Mel-frequency cepstral coefficients (MFCCs) [18]. Our focus is on the essential low-level task in video summarising and indexing: *scene detection* [19]. We also report interesting results for shot detection [20]. We point out the following interesting findings:

- In lifelog analysis, audio cues seem to be much more important than in the previous works on movie or broadcast videos (e.g., the TRECVid campaign [20]) or camcorder recorded home videos. In our experiments, visual cues often fail in scene detection.
- Video cues, however, provide partially complementary information to audio, and that can be used to boost the detection accuracy without too much computational increase using the soft decision cascade paradigm proposed in this work.
- The cascades can be constructed easily by separating the detector parameter and cascade parameter optimisation into two separate stages.

3 Decision Cascades

The general goal of constructing cascades is to find a set of detectors (nodes) that minimise a target function consisting of penalties for accuracy loss, (computational) cost of evaluating nodes, and a regularisation term to avoid overfitting [6]. Optimisation of such target function requires inter-operability of the detectors, for example, access to the internal decision tree nodes in [6], i.e. cascade methods operate on "weak classifiers". In our case, we may have N very different type of detectors pre-trained for the same task and we wish to explicitly cascade them into the computationally fastest order. A same detector may appear multiple times but its execution is needed only once. In that sense, our approach is not consistent with the assumptions with the cascading works [6–8], but more resembles the combining classifiers ideology [9] where "strong classifiers" are trained and their combination weights optimised. Our combination, however, is a cascade structure and weights are detection thresholds.

A strong detector cascade is constructed by combining N detectors $D_i, i = 1, \ldots, N$ that map an input feature space X to a decision $t \in T$ in the decision space T. For simplicity, we may assume that t is binary, i.e. $D : X \rightarrow \{0, 1\}$. Our scene change detection is also a binary task. A cascade can be represented as a logical function, such as

$$D = (D_1 \cap D_2 \cap \ldots) \tag{1}$$

or

$$D = (D_1 \cup D_2 \cup \ldots) \tag{2}$$

or any disjunction of conjunctions. It is clear, that significant computational improvement can be achieved if the detector D_i returns 0 in (1) or 1 in (2), since then execution of D_j for $j > i$ is then unnecessary. In particular, if the detectors are indexed such that the computational complexities increase, $\Omega(D_i) \ll \Omega(D_j)$ for $i < j$, then the cascade can provide remarkable computational speedup.

The problem is that there are a large number of ways to combine outputs of the detectors, especially when the number of detectors increase, and only a few are optimal for a certain task. Moreover, the optimal configuration does not only mean an optimal form of the logical function D, but also optimal values for each detector's internal parameters $\boldsymbol{\Theta}_i$ and cascade parameters $\boldsymbol{\Phi}$ (detection thresholds). The only approach guaranteeing the global optimum is the exhaustive search which easily becomes unfeasible. We propose a doable but still effective optimisation procedure by the following assumptions:

- The cascade structure is built such that the complexity of detectors increase gradually: the computationally lightest detector first and heaviest last.
- The detectors are pre-trained: the parameters $\boldsymbol{\Theta}_i$ are optimised independently for the given task.
- The task is to optimise the cascade parameters $\boldsymbol{\Phi}$ for the fixed structure and pre-trained detectors.

The first assumption is justified by the fact that it can provide the lowest computational complexity for similar performance. If the detectors mutually correctly detect (in case of D as in Eq.(2)) and leave undetected (in case of D as in Eq.(1)) the same part of the input space, the detection performance can be even improved in addition to saving in the computational load. The second and third assumptions are justified by the fact that since the exhaustive search is not feasible, a separate optimisation of each detector still provides the best average performance and their mutual relationship is compensated on the cascade level parameter optimisation. A greedy algorithm for the optimisation is given in Algorithm 1. The algorithm is in the sense greedy that it moves thresholds one by one always selecting the threshold that provides the smallest amount of negative examples while including one more positive example. This iteration is repeated until all positive examples are covered.

4 Audio and Visual Cues

For our cascade construction in Section 3 we only need that a selected classifier outputs classification scores for tested example (y_n^i). For scene detection we

Data: Target class classification scores y_n^i, for $i = 1 \dots M$ data points and N classifiers; logical cascade expression in the disjunctive normal form;
Result: Precision(P) – recall(R) curve and cascade parameters $\mathbf{\Theta}$ for every point on it.
Init: $\mathbf{\Theta} = [\theta_1, \theta_2, \dots, \theta_N] = [\infty, \infty, \dots, \infty]$; `// P=R=0`
while $R < 1$ **do**
  for *each conjunctive ($\wedge$) part of the cascade* **do**
    Find the new thresholds $\hat{\theta}_j$ for participating sub-classifiers $\mathcal{D}_j$ that select one new positive example and count the number of negative examples introduced.
  end
  Set $\theta_j \leftarrow \hat{\theta}_j$ based on the component providing the smallest amount of negative examples;
  Store $\mathbf{\Theta}$;
  Compute and store P and R ;
end

selected the audio and visual cues most successful in earlier works. These cues are shortly reviewed next.

4.1 MFCC Detector

As audio features we use Mel-frequency cepstral coefficients (MFCC) [18] which have proved to be useful in many audio information retrieval tasks like speech recognition [21], audio event detection [22, 23] and music information retrieval [24].

The audio track is analysed in successive, non-overlapping frames (not to be conflicted with video frames). From an audio frame at time t, one MFCC-vector $\mathbf{x}(t)$ of length $D_{\mathbf{x}}$ is extracted. The context change with MFCC cut detector is measured according to changes in distributions of vectors $\mathbf{x}$. A mean $\mu_d(t)$ and variance $\sigma_d(t)$ of each MFCC, indexed by d, is calculated within a sliding audio frame sequence of length T_s preceding time t. A distance between consecutive audio frames, $L_{\mathrm{MFCC}}(t)$, for scene and shot change detection at time t is then given by

$$L_{\mathrm{MFCC}}(t) = \sum_{d=1}^{D_{\mathbf{x}}} \left| \frac{\mu_d(t) - \mu_d(t+T_s)}{\sigma_d(t) + \sigma_d(t+T_s)} \right|^2 \tag{3}$$

that is slightly different to Fisher's linear discriminant, but found better in our experiments.

4.2 Colour (RGB) Detector

Despite of its simplicity, variants of colour (RGB) histogram distance have been used in the most state-of-the-art shot detection methods [20] and since it is also one of the computationally cheapest visual features it was selected for our experiments. An RGB histogram is computed from each video frame. The histogram

vector $\mathbf{h}(t)$ of the frame t is of length 192, containing the incidence frequencies of pixel values 1-64 on red, green and blue channel.

A distance $L_{\mathrm{RGB}}(t)$ of two consecutive RGB frames is calculated as

$$\begin{aligned} L_{\mathrm{RGB}}(t) = &\left| \Delta_{\mathbf{h}}(t) - \overline{\Delta_{\mathbf{h}}(t-1)} \right| \\ &+ \left| \Delta_{\mathbf{h}}(t) - \overline{\Delta_{\mathbf{h}}(t+T_{\mathbf{h}})} \right|. \end{aligned} \tag{4}$$

The idea is, that gradual change is a natural way of RGB histogram evolving. Thus we compare the L_1 change $\Delta_{\mathbf{h}}(t) = \|\mathbf{h}(t) - \mathbf{h}(t-1)\|_1$ between RGB-histograms of consecutive frames to running average change $\overline{\Delta_{\mathbf{h}}(t-1)}$ over $T_{\mathbf{h}}$ preceding frames to see whether the view has changed entirely instead of natural evolution. The second term in (4) accounts for comparing the current change to the forthcoming video frames respectively (not available for on-line processing).

4.3 SIFT Bag-of-Words Detector

This approach is computationally much slower than MFCC and RGB based detectors, but it has been the mainstream approach in detection of visual object classes [25, 26]. At the core of this method are histograms of codes of local patch descriptors (SIFT) extracted from each video frame. For patch encoding, a visual codebook must be constructed from extracted descriptors. It has been reported that specific codebooks constructed from the input video perform much better than general codebooks and therefore this approach was adopted by us. The codebook is constructed from k-means clustering with a fixed k (codebook size). For each video frame, SIFT descriptors are extracted on a dense grid, assigned to the best matching codes, and the histogram of codes computed and used as a feature. To compute a shot or scene change score at time t from SIFT-histograms $\mathbf{b}$, a plain L_1-distance $L_{\mathrm{BOW}}(t) = \|\mathbf{b}(t) - \mathbf{b}(t-1)\|_1$ is used. Overall, the L_1 distance instead of the Euclidean distance for evaluating the difference between consecutive histograms, both RGB and BoW detectors, worked clearly best. The settings were selected based on the best found in unsupervised image classification using SIFT bag-of-features [27].

5 Experiments

Data, experiments, performance measures, and results for the selflog video scene detection are reported in this section. Since the same method also applies for shot detection (camera switched) we also report our shot detection results.

5.1 Captured selflog data set

We have collected over 7 hours of video data for our evaluations (Fig. 1). The videos were shot with a small spy camera with the frame rate 15 frames/second and frame size of 176x144 pixels. The frames are YUV420p encoded with h263

compression and stored in an mp4 container. The stereo sound tracks are recorded by a pair of in-ear microphones with 44.1kHz sampling rate and stored without compression.

The database contains video from 23 different types of environments (scenes), 6 - 16 shootings from each: amusement park, basketball game, beach, bus, cafeteria, inside car, family yard, football game, hallway, home, inside train, nature, office, outdoor festival, outdoor market, party, pub/club, railway station, restaurant, shop, sports event, at street and track'n'field.

The video was annotated for shot and scene detection. Shot detection corresponds to the situation that the scene remains the same, but the camera was turned off and turned back on in a different location in the same scene. The scene change corresponds to the situation that the user moves to another environment.

In our evaluation, the automatically found change points were compared to the known true scene and shot change times (groundtruth). If the found change point was within 0.25 seconds from a true change point, the detection was assigned correct.

5.2 Performance Measures

To compare the performances of the used shot and scene detection methods, we use precision, $P = tp/(tp + fp)$, recall $R = tp/N$ where tp stands for the number of correct shot or scene changes depending on the task, fp stands for the number of incorrectly identified change points and N is the total number of true change points in the video. A combination of R and P, an F-measure $F = 2 \cdot R \cdot P/(R + P)$, is also used as it simplifies comparison by describing the detection performance with a single value. We are also taking the computation time needed by different systems into account. The computation time CT is given as a number relative to the length of a video, i.e. for $CT = 1$ the system works tightly in real time.

5.3 Detector parameters

To avoid overfitting to our test data, we trained the detector parameters with separate material of home videos collected before the selflog data. The data is similar to lifelog data, but does not contain the same scenes and was recorded with a standard-quality hand-held camcorder.

In the colour histogram based RGB detector the only method parameter is the length of the time interval to calculate the average change of consecutive RGB-histograms T_{RGB}. The value $T_{\mathrm{RGB}} = 10$ video frames was found best.

In the BoW detector the main method parameter is the SIFT codebook size. We also experimented different detectors and descriptors, but the dense SIFT in the VLFeat toolbox (http://vlfeat.org) was found the best. The codebook is computed from the input data and the optimal codebook size was $D_{\mathrm{SIFT}} = 100$.

Based on experiments with the homevideo data, the following MFCC parameters were selected:

– audio window length = 80 ms
– number of Mel-frequency bands = 80
– number of MFCCs, $D_{\mathbf{x}} = 20$
– audio frame sequence length, $T_s = 10$ s

The number of Mel-frequency bands and the number of used MFCCs did not make a big difference in performance. The audio frame length and the sequence length for distribution estimation were more sensitive. Another finding was that the longer the audio window and the longer the sequence length, the better is the performance. However, to be able to detect also short scenes, these parameters were restricted.

5.4 Results

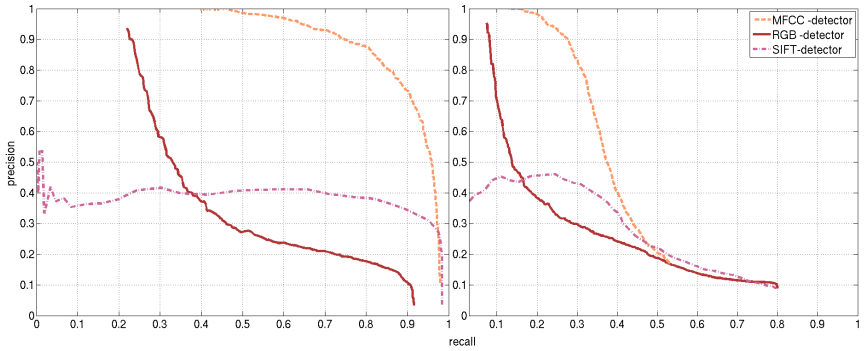

Fig. 2. Precision-recall curves for the single MFCC, RGB and SIFT detectors in scene detection (left) and shot detection (right).

Single detectors - The results of the single detectors in scene and shot detection are shown in Figure 2. It is noteworthy that all detectors have very different behaviour with respect to precision and recall. The striking result, however, is that for selflog data the audio cue outperforms the both visual cues with clear margins and being more prominent in scene detection where it is almost twice better. The result is quite opposite to state-of-the-art results with pre-edited material such as movies and TV programs [13, 14, 3, 15].

Detector Cascades - The results for various cascades are shown in Table 1 including the single detectors. The single audio MFCC detector performs surprisingly well (F-score: 0.84), but as indicated by the different behaviour of the single precision-recall curves in Figure 2 the other detectors also provide strong complementary information about scene changes. This is evident as the optimal relationship is AND ($\cap$) and for the two cascades MFCC and RGB and MFCC and SIFT the results are 0.90 and 0.95: when two detectors make a wrong decision the third corrects it. Note that for the both cases the computation time

Table 1. Selflog scene and shot detection cascade performances. Performances are reported as the best F-scores in the precision-recall curve with the corresponding cascade computing time (CT) (processing time in seconds per second of video).

Cascade	Scene detection		Shot detection	
	F-score	CT (s/s)	F-score	CT (s/s)
MFCC only	0.84	0.01	0.46	0.01
RGB only	0.40	0.43	0.31	0.43
SIFT only	0.52	184.00	0.37	184.00
MFCC ∪ RGB	0.85	0.44	0.46	0.44
MFCC ∩ RGB	0.90	0.02	0.49	0.03
MFCC ∪ SIFT	0.84	184.00	0.45	184.00
MFCC ∩ SIFT	0.95	0.30	0.51	0.81
RGB ∩ SIFT	0.68	1.30	0.42	1.30
MFCC ∩ RGB ∩ SIFT	0.91	0.30	0.48	0.38
(MFCC∩RGB) ∪ (MFCC∩SIFT)	0.96	0.30	0.52	0.31
(MFCC∩RGB) ∪ (MFCC∩SIFT) ∪ (RGB∩SIFT)	0.96	0.30	0.53	0.32

is 2× faster than real-time (super frame rate). The best scene detection accuracy is F-score 0.96 which is achieved with classifiers trained with completely separate data and only optimising the cascade parameters. The resulting classifier is a disjunction of the two available strong conjunctions and achieves the performance with computation time 0.30 seconds needed to process 1.0 seconds of video input ($> 3\times$ frame rate). It is noteworthy that the SIFT detector is essential for the performance while it is active only in very few cases as apparent by comparing its single detector and cascade detector computing times.

The same findings hold also for shot detection (best single 0.46, best cascade 0.53) which is much more difficult task in the case of lifelog data.

It should be noted that the selection of cascade parameters is not critical for good performance, since they mutually compensate each other providing smooth and intuitive performance change.

6 Conclusions

The ultimate goal of our work is fast streaming, storing, indexing, retrieval and sharing of selflog video produced by millions of users using their wearable video capturing devices. Past research on video analysis has provided effective but often too slow methods for the above tasks. In this work, we sought to improve the existing techniques with the help of two hypotheses: *multiple video modalities* provide complementary information and *cascade type processing* improves efficiency. The both assumptions were found valid in our experiments where scene and shot detection from real lifelog recordings of more than seven hours were investigated. The strikingly important role of audio, complementary of audio and video, and finally the optimised cascade structure provided us superior detection

accuracy in super frame rate. These results indicate that cascades are the tools of future, fusing even more modalities (GPS, accelerometer, gyroscope, compass, barometer, proximity etc.) can be beneficial, and computationally light methods can be constructed from the existing methods. In our future work, we will follow these findings and investigate a light-weight cascade for on-line video skimming and scene indexing.

References

1. Gygli, M., Grabner, H., Riemenschneider, H., Gool, L.V.: Creating summaries from user videos. (2014)
2. Zhao, B., Xing, E.: Quasi real-time summarization for consumer videos. In: Proc. of the CVPR. (2014)
3. Kyperountas, M., Kotropoulos, C., Pitas, I.: Enhanced eigen-audioframes for audiovisual scene change detection. **9** (2007)
4. Song, Y., Zhao, M., Yagnik, J., Wu, X.: Taxonomic classification for web-based videos. In: Proc. of the CVPR. (2010)
5. Viola, P., Jones, M.: Robust real-time face detection. Int J Comput Vis **57** (2001)
6. Chen, M., Xu, Z., Weinberger, K., Chapelle, O., Kedem, D.: Classifier cascade for minimizing feature evaluation cost. In: AISTATS. (2012)
7. Wu, T., Zhu, S.C.: Learning near-optimal cost-sensitive decision policy for object detection. In: ICCV. (2013)
8. Shen, C., Wang, P., Paisitkriangkrai, S., van den Hengel, A.: Training effective node classifiers for cascade classification. Int J Comput Vis **103** (2013) 326–347
9. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classfiers. IEEE PAMI **20** (1998)
10. Wang, M.: Movie2comics: Towards a lively video content presentation. IEEE Transactions on Multimedia **14** (2012) 858–870
11. Yip, S.: The automatic video editor. In: ACM Multimedia. (2003) 596–597
12. Chen, S.C., Shyu, M.L., Liao, W., Zhang, C.: Scene change detection by audio and video clues. In: ICME (2). (2002) 365–368
13. Pfeiffer, S., Lienhart, R., Effelsberg, W.: Scene determination based on video and audio features. In: Multimedia Tools and Applications. (1999) 685–690
14. Jiang, H., Lin, T., Zhang, H.: Video segmentation with the assistance of audio content analysis. In: IEEE International Conference on Multimedia and Expo (III). (2000) 1507–1510
15. Smeaton, A.F., Over, P., Kraaij, W.: Trecvid: Evaluating the effectiveness of information retrieval tasks on digital video. In: Proceedings of ACM Multimedia, New York, USA (2004)
16. Gargi, U., Kasturi, R., Strayer, S.H.: Performance characterization of video-shot-change detection methods. **10** (2000)
17. Lowe, D.G.: Distinctive features from scale-invariant keypoints. Int J Comp Vis **60** (2004) 91–110
18. Steven B. Davis, P.M.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing **ASSP-28** (1980) 357–366
19. Fabro, M., Boszormenyi, L.: State-of-the-art and future challenges in video scene detection: a survey. Multimedia Systems **19** (2013) 427–454

20. Smeaton, A., Over, P., Doherty, A.: Video shot boundary detection: Seven years of TRECVid activity. Computer Vision and Image Understanding **114** (2010) 411–418
21. L. R. Rabiner, B.J.H.: Fundamentals of Speech Recognition. Prentice Hall (1993)
22. Heittola, T., Measaros, A., Virtanen, T., Eronen, A.: Sound event detection in multisource environments using source separation. In: Workshop on Machine Listening in Multisource Environments, Florence, Italy (2011) 36–40
23. J.-J. Aucouturier, B. Defreville, F.P.: The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscape but not for polyphonic music. Journal of Acoustical Society of America **122** (2007) 881–891
24. Downie, J.: Music information retrieval. Annual Review of Information Science and Technology **37** (2003) 295–340
25. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. of the ICCV. (2003)
26. Csurka, G., Dance, C., Willamowski, J., Fan, L., Bray, C.: Visual categorization with bags of keypoints. In: ECCV Workshop on Statistical Learning in Computer Vision. (2004)
27. Tuytelaars, T., Lampert, C., Blaschko, M., Buntine, W.: Unsupervised object discovery: A comparison. Int J Comput Vis **88** (2010)