AUDIO PHRASES FOR AUDIO EVENT RECOGNITION

Huy Phan[⋆†], *Lars Hertel*[⋆], *Marco Maass*[⋆], *Radoslaw Mazur*[⋆], and *Alfred Mertins*[⋆]

[⋆]Institute for Signal Processing, University of Lübeck, Germany
[†]Graduate School for Computing in Medicine and Life Sciences, University of Lübeck, Germany
Email: {phan, hertel, maass, mazur, mertins}@isip.uni-luebeck.de

ABSTRACT

The bag-of-audio-words approach has been widely used for audio event recognition. In these models, a local feature of an audio signal is matched to a code word according to a learned codebook. The signal is then represented by frequencies of the matched code words on the whole signal. We present in this paper an improved model based on the idea of *audio phrases* which are sequences of multiple audio words. By using audio phrases, we are able to capture the relationship between the isolated audio words and produce more semantic descriptors. Furthermore, we also propose an efficient approach to learn a compact codebook in a discriminative manner to deal with high-dimensionality of bag-of-audio-phrases representations. Experiments on the Freiburg-106 dataset show that the recognition performance with our proposed bag-of-audio-phrases descriptor outperforms not only the baselines but also the state-of-the-art results on the dataset.

Index Terms— audio phrase, bag-of-words, audio event, recognition, human activity

1. INTRODUCTION

Machine hearing has recently received great attention [1]. In particular, recognition of audio events is important for many applications such as automatic surveillance, multimedia retrieval, and ambient assisted living. Apart from speech and music, audio events can be indicative of natural sounds (e.g. wind sounds, water sounds, and animal sounds) and artificial sounds (e.g. laugh, applause, and foot steps) [2]. In this work, we focus on the recognition of artificial sounds related to daily human activities which are useful for ambient assisted living, the new emerging application to tackle the fast aging population problem [3, 4].

Many descriptors have been proposed to represent audio events for recognition. In general, any features that are used to describe an audio signal are also suited for audio events. Different hand-crafted representations have been proposed. Most of them are borrowed from the field of speech recognition, such as mel-scale filter banks [5], log frequency filter banks [6], and time-frequency features [7, 8]. With the rapid advance of machine learning, automatic feature learning is becoming more common [9–11]. Among these techniques, bag-of-words (BoW) models have been widely adapted to the field and good performance has been reported [10–12].

Many audio events expose temporal structure, i.e. it is possible to decompose them into atomic units of sound [13]. For example, the sound of a "using water tap" event may be further composed of the sounds of the water running in the tap, then pushing into the air, and finally splashing into the sink. Therefore, aggregating temporal configurations of audio events is a promising approach. The problem with the BoW descriptors is that they are produced by unordered isolated words, hence do not take the structural information into account. To model the temporal context for audio events, pyramid BoW models [11] and n-gram extensions [14] have been proposed.

In this work, we propose to use *audio phrases* which are composites of multiple words. By grouping audio words into phrases, we are able to encode the arrangement between the words and capture the temporal information at a certain degree. The idea is similar to the n-gram language models [14, 15] and the visual phrase concept in computer vision field [16, 17]. However, this class of representations confronts one with the large induced dimensionality [14, 16, 17]. Our proposed audio phrase focuses on coping with this problem. The dimensionality of the bag-of-phrases (BoP) feature space grows exponentially with the size of the codebook, which hinders the conventional clustering-based codebook learning approaches in which the number of audio words needs to be reasonably large to obtain a good performance. To alleviate this issue, we alternatively employ a classification model to discriminatively learn a compact codebook in which the number of code words is equal to the number of target event categories. The experiments on the Freiburg-106 dataset show that: (1) the BoW descriptors with the compact codebook show superior performance compared to the clustering-based counterparts, and (2) the recognition with BoP descriptors outperforms not only the BoW and pyramid BoW baselines

This work was supported by the Graduate School for Computing in Medicine and Life Sciences funded by Germany's Excellence Initiative [DFG GSC 235/1]. We would also like to thank Johannes A. Stork for providing the Freiburg-106 dataset.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

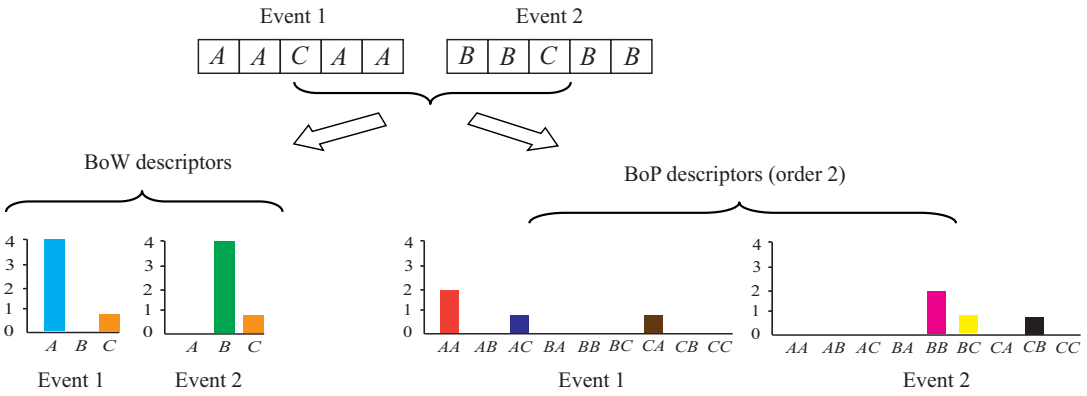

Fig. 1. Illustration of BoW and order-2 BoP descriptors produced for two different events. The events are simulated as two sequences of matched code words of the codebook $\mathcal{K} = \{A, B, C\}$.

but also the state-of-the-art results on the dataset in terms of the f-score measure.

Our main contributions are two-fold. First, we propose the concept of audio phrases which are combinations of multiple words and BoP descriptors for efficient audio event representation. Second, we propose to learn a compact codebook to deal with the large dimensionality of BoP feature space.

2. THE APPROACH

2.1. A typical BoW model

The BoW approach is a technique used to model an audio signal using its local features. Typically, the signal is decomposed into multiple segments each of which is described by a vector of low-level features. The goal is to quantize these local features using a codebook. The codebook can be built from the local features obtained by audio events in training data using a clustering method such as k-means [12] or Gaussian Mixture Model (GMM) [11]. In k-means based methods, a code word is usually represented by the cluster centroid. Within a probabilistic clustering framework, code words can be represented by the GMM . A local feature vector is then matched to a code word in the learned codebook with a certain weight. The weight assignment can be "hard" (e.g. with k-means) or "soft" (e.g. with GMM). The descriptor for the signal is finally produced by simply accumulating the weights of the code words.

2.2. Audio phrases and BoP descriptor

While the audio words in a BoW model are unordered, it is reasonable to group words into phrases which offer a higher semantic information level to enrich the BoW representation. Suppose that we have learned a codebook $\mathcal{K} = \{c_1, \ldots, c_K\}$ of size K from training data. Without loss of generality, we denote an audio phrase $\mathcal{P}_{(c_{k_1},\ldots,c_{k_N})}$ of order $N \geq 1$ as an ordered sequence of N code words $(c_{k_1}, \ldots, c_{k_N})$ where $c_{k_1}, \ldots, c_{k_N} \in \mathcal{K}$. As a result, there are totally K^N possible order-N audio phrases. It reduces to the standard BoW model when $N = 1$.

Given an audio signal, we decompose it into a sequence of S segments $(\mathbf{x}_1, \ldots, \mathbf{x}_S)$ where $\mathbf{x}_i$ is the descriptor of the segment at the time index i. Each subsequence of N local segments $(\mathbf{x}_i, \ldots, \mathbf{x}_{i+N-1})$ is then matched to the order-N audio phrase $\mathcal{P}_{(c_{k_1},\ldots,c_{k_N})}$ with the assigned weight given by

$$\mathcal{W}\big(\mathcal{P}_{(c_{k_1},\ldots,c_{k_N})}|(\mathbf{x}_i, \ldots, \mathbf{x}_{i+N-1})\big) = \prod_{m=1}^{N} \mathcal{W}(c_{k_m}|\mathbf{x}_{i+m-1}). \quad (1)$$

Here, $\mathcal{W}(c|\mathbf{x})$ is the assigned weight by matching the segment $\mathbf{x}$ to the code word c. $\mathcal{W}$ can be a probability function (e.g. using GMM-based clustering) or an indicator function (e.g. using k-means clustering). The accumulated weight by matching all possible order-N subsequences of the signal to the audio phrase $\mathcal{P}_{(c_{k_1},\ldots,c_{k_N})}$ reads

$$\mathcal{W}\big(\mathcal{P}_{(c_{k_1},\ldots,c_{k_N})}|(\mathbf{x}_1, \ldots, \mathbf{x}_S)\big) = \sum_{i=1}^{S-N} \mathcal{W}\big(\mathcal{P}_{(c_{k_1},\ldots,c_{k_N})}|(\mathbf{x}_i, \ldots, \mathbf{x}_{i+N-1})\big). \quad (2)$$

Eventually, the audio signal is represented by the weights obtained by matching it to all possible order-N audio phrases. In Fig. 1, we illustrate the BoW and BoP representations for two simple simulated events.

It has been shown that audio events embed temporal structure [13]. Descriptors that encode these temporal configurations would offer better discrimination. Recently, the approach using temporal pyramids of BoW representations [11] has demonstrated state-of-the-art results on several benchmark datasets. This model encodes the temporal layouts by splitting the audio signal into hierarchical cells, then computes BoW representations for each cell, and concatenates all

the representations at the end. Towards this goal, the rational behind using phrases is to model the co-occurrences of the words in local neighborhoods, and therefore encode the temporal configuration of the events.

Furthermore, the BoP representations also exhibit a denoising property. Usually, if there exist sharing features between audio events [18], in which two events may have similar subsequences, they likely occur in patterns of multiple consecutive segments. The intermittent occurrence of a code word, which is different from its neighbors, should be considered as noise, and therefore, filtered out. Let us revisit the example in Fig. 1. Two different events have the code word "*C*" in common which should be considered as noise. Comparison of the BoW descriptors, e.g. histogram intersection, will result in a positive similarity value due to the positive weights assigned to "*C*". Whereas, the similarity value is zero when using the BoP descriptors. In other words, using the BoP descriptors has canceled out the noisy "*C*" and increased the distinction between two events.

2.3. Discriminative learning of compact codebook

For the BoW models that use clustering methods for codebook learning, the performance heavily depends on the codebook size. More often than not, the codebook size is multiple-order larger than the number of target event categories. To support our argument, we show in Fig. 2 the performance of the baseline system using a BoW model (more details in Section 3) on the Freiburg-106 dataset [19] as a function of codebook size. The codebook was constructed using k-means. It can be seen that a codebook size of 200 is a good choice in this case. Given the fact that the number of event categories is 22, the codebook size is about ten times larger. On the other hand, using this codebook, the feature space induced by the order-N BoP has the dimensionality of 200^N. It is 4×10^4 with $N=2$ and 8×10^6 with $N=3$. This exponential growth of dimensionality makes clustering-based codebook learning inappropriate for the BoP models.

We propose to learn a compact codebook in a supervised manner to alleviate the high-dimensionality problem. While the conventional clustering methods ignore the labeling information, integrating them into the codebook construction offers more discrimination power [20]. Inspired by this, rather than using clustering, we employ classification models for codebook matching. As a result, the codebook size is equal to the number of target event categories, and the dimensionality of the BoP descriptors will be magnificently reduced. Although multiple one-vs-rest binary classifiers would suite this goal, we use random-forest classification [21] to learn a multi-class classifier at once. Moreover, random forest naturally supports probability outputs. Therefore, both hard and soft codebook matching can be explored simultaneously.

Suppose that we have C event categories of interest, and hence, the number of code words is $K \equiv C$. Furthermore,

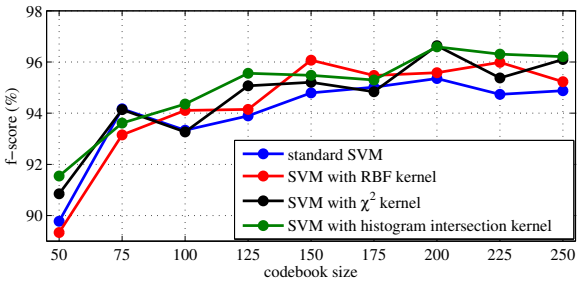

Fig. 2. Performance variation of the BoW model on the Freiburg-106 dataset as a function of codebook size.

suppose that we have learned the random forest classifier $\mathcal{M}$ for codebook matching from training audio segments. The soft assigned weight by matching an unseen audio segment $\mathbf{x}$ to a code word $c \in \{1,\ldots,C\}$ reads

$$\mathcal{W}(c|\mathbf{x}) = P(c|\mathbf{x}). \tag{3}$$

Here, $P(c|\mathbf{x})$ is the probability that $\mathbf{x}$ is classified as class c. On the other extreme, the hard assignment yields the weight

$$\mathcal{W}(c|\mathbf{x}) = \mathbb{I}(c=\hat{c}|\mathbf{x}), \tag{4}$$

where

$$\hat{c} = \underset{c\in\{1,\ldots,C\}}{\operatorname{argmax}} \; P(c|\mathbf{x}), \tag{5}$$

and

$$\mathbb{I}(c=\hat{c}) = \begin{cases} 1, & \text{if } c=\hat{c} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

It will be shown in the experiments that the hard assignment scheme produces much sparser descriptors compared to those obtained with the soft assignment scheme at the cost of lower recognition accuracies.

3. EXPERIMENTS

3.1. Experimental setup

Test datasets. We tested our approach on the Freiburg-106 dataset [19]. This dataset was collected using a consumer-level dynamic cardioid microphone. It contains 1,479 audio-based human activities of 22 categories. Several sources of stationary ambient noise were also present. As in [19], we divided the dataset so that the test set contains every second recordings of a category, and the training set contains all the remaining recordings[1].

Parameters. Each audio signal was decomposed into a sequence of 50 ms segments with a step size of 10 ms. We trained a classifier $\mathcal{M}$ using random-forest classification [21]

[1]This is based on unofficial communication with the authors of [19].

with 200 trees for codebook matching. For the purpose of classification, an audio segment was labeled with the label of the event from which it originated.

Audio event classification models. Our event recognition systems were trained on the BoP descriptors using one-vs-one support vector machine classification (SVM) with histogram intersection kernel. To extract the descriptors for the training events, we conducted 10-fold cross validation on the training data. The hyperparameters of the SVMs were tuned via leave-one-out cross-validation.

Baseline systems. We compare the performance of our systems with two baseline systems:

1. Bag-of-words system (BoW): this system used a BoW model which has been widely used for audio event recognition [10, 12]. Using this model, an audio event is represented by a histogram of codebook entries.
2. Pyramid bag-of-words system (pBoW): We extracted BoW descriptors on different pyramid levels [22] to encode temporal structure of audio events. This approach has recently achieved state-of-the-art results on different benchmark datasets [11].

For all baselines, we used k-means for unsupervised codebook learning. The entries were obtained as the cluster centroids, and codebook matching was based on Euclidean distance. We used different codebook sizes $\{50, 75, \ldots, 250\}$. In particular, we tried 2, 3, and 4 pyramid levels for the pBoW systems. In addition to standard SVM, nonlinear SVMs with radial basis function (RBF), χ^2, and histogram intersection kernels were also implemented. All the hyperparameters were tuned by cross-validation. Finally, the systems which obtained the best performance were compared with our systems.

Evaluation metrics. For evaluation, we used the *f-score* metric, which considers both precision and recall values, to compare recognition accuracies:

$$\text{f-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (7)$$

3.2. Experimental results

Efficiency of the discriminative codebook. Let us denote an order-N BoP system as BoP-N. To show the advantage of the discriminative compact codebook, we compare the performance achieved by our BoP-1 systems (both hard and soft assignment schemes) with those of the baselines as in Table 1. It is worth emphasizing that no structural information was introduced in the model with the order-1 BoP descriptors, thus, they are essentially bag-of-words descriptors with the discriminative codebook. For the baselines, the best performances were obtained with the χ^2 kernel and a codebook size of 200. On the other hand, a pyramid level of two is found optimal for the pBoW baseline. It can be seen that our systems consistently outperform all baselines. Individually, our BoP-1 systems achieve equivalent or higher f-score on 17 out of 22 and 20 out of 22 event categories with the hard and soft assigment schemes, respectively. They also outperform the state-of-the-art results on the dataset reported in [19] with 5.4% and 5.9% relative improvements on average f-score, respectively.

Table 1. Recognition performance comparison in terms of f-score (%) of the BoP-1 systems and the baselines. We marked in bold where the BoP-1 systems give equal or better performance than both BoW and pBoW baselines.

Event Type	ID	BoW	pBoW	hard BoP-1	soft BoP-1
background	1	76.9	80.0	**82.9**	**88.9**
food bag opening	2	96.3	96.3	**98.7**	**98.7**
blender	3	100	100	**100**	**100**
cornflakes bowl	4	92.3	92.3	**97.3**	**97.3**
cornflakes eating	5	97.7	100	**100**	**100**
pouring cup	6	100	100	95.7	**100**
dish washer	7	96.7	97.8	**97.8**	**98.9**
electric razor	8	98.8	98.8	**100**	**100**
flatware sorting	9	91.9	91.9	**92.3**	**92.3**
food processor	10	91.9	91.9	**100**	**97.1**
hair dryer	11	100	100	**100**	**100**
microwave	12	97.9	96.8	**100**	**100**
microwave bell	13	100	100	**100**	**100**
microwave door	14	96.5	96.5	96.4	93.8
plates sorting	15	100	99.3	97.8	98.5
stirring cup	16	100	100	**100**	**100**
toilet flush	17	95.9	95.0	95.3	**96.8**
tooth brushing	18	96.3	96.3	**100**	**100**
vacuum cleaner	19	98.7	98.7	**100**	**100**
washing machine	20	100	100	**100**	**100**
water boiler	21	100	100	**100**	**100**
water tap	22	98.2	98.2	97.3	**99.1**
Average		96.6	96.8	**97.8**	**98.3**

Increasing the order of the BoP descriptors. In this experiment, we studied how the recognition performance and the sparseness of the BoP descriptors change with increasing orders. With a higher order, we are able to encode higher-level dependency between the isolated words in the BoP descriptors. We show in Table 2 the recognition performance of the BoP descriptors with different orders $N = \{1, 2, 3, 4\}$ for both hard and soft assignment schemes. One can clearly see the upward trend in f-score of the soft-assignment BoP systems when the order increases. The BoP-4 system achieves an improvement of 0.6% on f-score compared to the BoP-1 system. Given the high-level accuracy of the BoP-1 system, this improvement is very meaningful. When comparing the BoP-4 system to the pBoW baseline which takes into account the temporal structure of the events, an improvement of 2.1% on f-score is seen. Nevertheless, the upward trend is not clear on the system with the hard assignment scheme, most likely

Table 2. Recognition performance and sparseness of the BoP descriptors with different orders.

		BoP-1	**BoP-2**	**BoP-3**	**BoP-4**
hard	f-score	97.8	97.8	98.0	97.8
soft	(%)	98.3	98.7	98.7	98.9
hard	sparseness	81.53	97.74	99.81	99.99
soft	(%)	8.57	23.23	38.74	52.29

due to higher quantization errors. It is also expected that the performance will level off at a certain order.

It is also worth analyzing the sparseness of the BoP descriptors. We measure the sparseness by the percentage of zeros in all descriptors. It can be seen in Table 2 that when the order increases, the descriptors become sparser. In addition, the hard-assignment descriptors are much sparser than the soft-assignment counterparts, especially at high orders. Therefore, although the dimensionality of the BoP feature space grows fast with increasing orders, computation and storage can be very efficient due to the sparseness.

4. CONCLUSIONS

We introduced in this paper the idea of bag-of-audio-phrases descriptor to represent audio events. An audio phrase is defined as a sequence of multiple words. By using phrases instead of isolated words, we are able to capture temporal structure information of the events. We also proposed to employ classification models to discriminatively learn a compact codebook to cope with the high dimensionality induced by high-order audio phrases. The empirical results on the Freiburg-106 show that recognition with the discriminative codebook achieves much better performance compared to conventional clustering-based codebook. Furthermore, using bag-of-audio-phrases descriptors, our recognition systems outperform all baselines and the state-of-the-art results in terms of the f-score measure.

REFERENCES

[1] R. F. Lyon, "Machine hearing: An emerging field," *Signal Processing Magazine*, vol. 27, no. 5, pp. 131–139, 2010.

[2] D. Gerhard, "Audio signal classification: History and current techniques," Tech. Rep. TR-CS 2003-07, University of Regina, 2003.

[3] Jens Schröder, Stefan Wabnik, Peter W. J. van Hengel, and Stefan Götze, *Ambient Assisted Living*, chapter Detection and Classification of Acoustic Events for In-Home Care, pp. 181–195, Springer, 2011.

[4] T. Croonenborghs, S. Luca, P. Karsmakers, and B. Vanrumste, "Healthcare decision support systems at home," in *Proc. AAAI-14 Workshop on Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 2014.

[5] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[6] C. Nadeu, D. Macho, and J. Hernando, "Frequency and time filtering of filter-bank energies for robust HMM speech recognition," *Speech Communication*, vol. 34, pp. 93–114, 2001.

[7] J. Dennis, H. D. Tran, and E. S. Chng, "Image feature representation of the subband power distribution for robust sound event classification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 367–377, 2013.

[8] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[9] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, 2015.

[10] S. Pancoast and M. Akbacak, "Softening quantization in bag-of-audio-words," in *Proc. ICASSP*, 2014, pp. 1370–1374.

[11] A. Plinge, R. Grzeszick, and G. Fink, "A bag-of-features approach to acoustic event detection," in *Proc. ICASSP*, 2014, pp. 3704–3708.

[12] V. Carletti, P. Foggia, G. Percannella, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance using a bag of aural words classifier," in *Proc. AVSS*, 2013, pp. 81–86.

[13] A. Kumar, P. Dighe, R. Singh, S. Chaudhuri, and B. Raj, "Audio event detection from acoustic unit occurrence patterns," in *Proc. ICASSP*, 2012, pp. 489–492.

[14] S. Pancoast and M. Akbacak, "N-gram extension for bag-of-audio-words," in *Proc. ICASSP*, 2013, pp. 778–782.

[15] C. Y. Suen, "n-gram statistics for natural language understanding and text processing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 164–172, 1979.

[16] L. Torresani, M. Szummer, and A. Fitzgibbon, "Learning query-dependent prefilters for scalable image retrieval," in *Proc. CVPR*, 2009, pp. 2615–2622.

[17] M. A. Sadeghi and A. Farhadi, "Recognition using visual phrases," in *Proc. CVPR*, 2011, pp. 1745–1752.

[18] H. Phan and A. Mertins, "Exploring superframe co-occurrence for acoustic event recognition," in *Proc. EUSIPCO*, 2014, pp. 631–635.

[19] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras, "Audio-based human activity recognition using non-Markovian ensemble voting," in *Proc. RO-MAN'12*, 2012, pp. 509–514.

[20] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Proc. NIPS*, 2006, pp. 985–992.

[21] L. Breiman, "Random forest," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[22] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, 2006, pp. 2169–2178.