DETECTING GENERIC VISUAL EVENTS WITH TEMPORAL CUES

Lexing Xie[†], *Dong Xu*[‡], *Shahram Ebadollahi*[†], *Katya Scheinberg*[†], *Shih-Fu Chang*[‡], *John R. Smith*[†]

[†]IBM T. J. Watson Research Center, NY
[‡]Department of Electrical Engineering, Columbia University, NY*

ABSTRACT

We present novel algorithms for detecting generic visual events from video. Target event models will produce binary decisions on each shot about classes of events involving object actions and their interactions with the scene, such as *airplane taking off*, *exiting car*, *riot*. While event detection has been studied in scenarios with strong scene and imaging assumptions, the detection of generic visual events from an unconstrained domain such as broadcast news has not been explored. This work extends our recent work [3] on event detection by (1) using a novel bag-of-features representation along with the earth movers' distance to account for the temporal variations within a shot, (2) learn the importance among input modalities with a double-convex combination along both different kernels and different support vectors, which is in turn solved via multiple kernel learning. Experiments show that the bag-of-features representation significantly outperforms the static baseline; multiple kernel learning yields promising performance improvement while providing intuitive explanations for the importance of the input kernels.

1. INTRODUCTION

We are concerned with detecting visual event classes from video, i.e., deciding if a video shot contain a class of events involving object actions and their interactions with the scene, such as *airplane taking off*, *exiting car*, *riot*, etc. In the past few years, the detection of generic visual concepts has received much attention through large-scale benchmarking activities [10]. Therein the focus has been mainly on extracting features and building models for static images and video frames, where the dynamic aspect of video has been underemphasized. Modeling time and content evolution in generic video content (e.g., television broadcast) is a hard problem due to large scene diversity and imaging variations. Yet modeling temporal events is crucial to generating rich metadata to video content, without which the afore-mentioned dynamic concept categories will remain undetectable.

Prior approaches to visual event modeling and detection mostly fall in two categories: object-centered and feature-driven. The object-centered approaches are based on the bottom-up perception model in computer vision, they explicitly identifies and tracks objects/agents [5, 6] and infers their actions and interactions via either deterministic grammar [5] or statistical measures of evolution or sequence similarities [6]. The feature-driven approaches [12] typically extract object- and event- independent descriptors from the video stream, and then learn an event model from the statistical distributions and evolutions of the descriptor streams. It is worth noting, however, that the object-centered approaches usually require strong assumptions regarding the camera setup and the scene for object segmentation and tracking, in addition to having strong assumptions on how many objects can be involved in the event. The feature-driven approaches, on the other hand, rely on low-level features that does not directly reflect the semantics in the scene, and they are typically better suited to describe events that span multiple shots.

Our recent work [3] detects events from visual concept streams, extracts features from variable-length multidimensional streams with hidden Markov models (HMM), which are then used to learn a classifier for the event class. This work extends our prior work [3] in two aspects: (1) modeling a shot as a collection of frames and the use a new distance metric, the earth-mover's distance to compute the (dis-)similarity between shots; (2) learning the importance among different input streams by formulating the multi-stream decision function into a double-convex combination of both the different similarity measures and the support vectors, which is in turn solved via multiple kernel learning (MKL). Experiments show that the distance metric induced by the bag-of-frames approach significantly outperform both the static baselines and the HMM-based approaches, and MKL increase average precision by 5% from an SVM baseline, while learning kernel weights with rich ontological explanations.

In the rest of this paper, Section 2 defines the problem scope of visual event detection and gives an overview of the event detection workflow. Section 3 describes the modeling of video shots as a bag of frames and the computation of EMD for deriving a inner product between two shots. Section 4 presents the algorithm for fusing different cues using MKL. Section 5 reports our experimental results followed by a conclusion in Section 6.

*This material is based upon work funded in part by the U. S. Government. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Government.

2. SEMANTIC CONCEPTS AND EVENTS

The term *video event* has been loosely used in the computer vision and multimedia research community. For instance, any of the following three scenarios qualifies as an event of *person/people running*: (1) from a fixed surveillance camera, detect a particular person runs in a particular direction (e.g., into building A). (2) one or more person(s) runs in any direction seen from any continuous camera take within a few seconds. (3) athlete(s) prepare and set off at the blast of a start revolver, several cameras take turns to follow them through the tracks and the scene ends in a closeup upon completion of the race. Each of these scenarios have very different assumptions about the scene, the imaging conditions and the time span of the event, therefore they typically call for different detection strategies. In this work we focus on the second scenario, i.e., detecting the presence and absence of an event class within a shot from any camera. Solutions to this sub-category can generate an importance class of annotations, help reduce the need for camera calibration and help build better search and navigation systems, most of which are useful for open-source surveillance, consumer data and other data domains.

Denote a video shot as v in a video collection $\mathbf{V}$ with $t = 1, \ldots, t_v$ frames in each shot. Each shot has a label $y_v = \pm 1$ denoting whether or not the event of interest exist in any part of the shot. The event recognition workflow include three steps: (1) Descriptor computation that extracts a feature descriptor sequence $x_{v,t}$ from the frame at time t from each shot v. (2) Temporal feature extraction that computes either a fixed-length vector for each shot, or a distance/similarity metric for each pair of shots directly. (3) Decision learning, where an SVM or similar discrimination and combination strategies are learned on the feature vector or distance metric.

For step (1) we choose a few descriptors base on the nature of the target classes and from prior experiences on the performance of different descriptors [10, 1]. Desirable descriptors should be invariant to real-world scene and imaging variations as well as being capable to model a wide range of concepts. A carefully-chosen set of semantic concepts, for example, can have strong correlation with a wide range of target events as well as provide intuitive explanations to the learned event models, as shown previously [1, 3]. For step (2) in our prior work [3] the temporal feature is extracted with HMM. A pair of HMM is trained for each dimension of the concept stream on examples in and out of the event class, respectively. The HMM are evaluated on each of the shots and a feature vector is obtained from either of the following two strategies (1) state-occupancy histogram fraction of time each shot stays in any hidden state. (2) Fisher information score denoting the sensitivity of the HMM parameters with respect to the data sequence being evaluated. While able to capturing the temporal evolution of multi-dimensional feature streams, HMM is naturally constrained by the choice of model size and the Markov assumptions about the hidden states. Therefore we explore an alternative representation for temporal streams and feature combination strategies in the next two sections.

3. SIMILARITY MEASURE VIA BAGS OF FEATURES

In this section, we present a bags-of-features representation and an corresponding algorithm to compute the similarity between two shots. Here the temporal order of frames in the same shot is ignored and their descriptors are treated as a "bag" of orderless features. Similar representations have been successfully applied to information retrieval [4] where text keywords in the same document are collected into a bag, and general object matching and recognition [13], where the features from local image patches are collected ignoring the location of the patches. The main advantage of this representation over explicitly representing the content and location of each word or image patch are in its invariance to grammar and phrasing in text or geometric and photometric variations in the image. It also alleviates the computationally expensive process of finding the correspondence among the words/patches between two documents or images. Here we use the bag-of-features representation along the temporal dimension for the event detection problem. This representation is invariant to temporal scale change and mis-alignment in different event instances.

With this representation we need a strategy to compare two shots, i.e., computing the similarity between two shots of different durations. We can, for example, collect aggregate statistics for the entire shot, such as a histogram over vector-quantized bins [4]. However using a histogram ignores the inherent similarity among different centroids, it would also be corrupted by noisy outlier frames from inaccurate shot boundary segmentations. One distance measure that simultaneously overcomes these two limitations is the Earth Mover's Distance (EMD) [8]. EMD finds a minimum weighted distance among all pair-wise distances between the two bags of point subject to weight-normalization constraints. Intuitively, EMD allows a partial match between the two collections, it also incorporates a ground-distance measure that takes into account the similarity between the histogram alphabets. EMD has shown promising performance in applications such as content based image retrieval [8] and object recognition [13].

The process for computing the distance between two shots are illustrated in Fig. 1. To compare the frame collection within two shots $u, v \in \mathsf{V}$, we cluster the two collections and form their respective signatures $U = \{(u_i, w_{u_i}),\ i = 1, .., m\}$ and $V = \{(v_j, w_{v_j}),\ j = 1, .., n\}$, where m, n are the total number of clusters, u_i, v_j are the cluster centers and w_{u_i}, w_{q_i} are the respective weights of clusters i and j, set as the size of the cluster, i.e., the fraction of frames that the cluster contains. We also have as input a *ground distance* matrix d_{ij} between cluster centers u_i and v_j, for $i = 1, .., m$, $j = 1, .., n$. The Earth Mover's Distance between the shots u and v is a linear combination of the ground distance d_{ij}

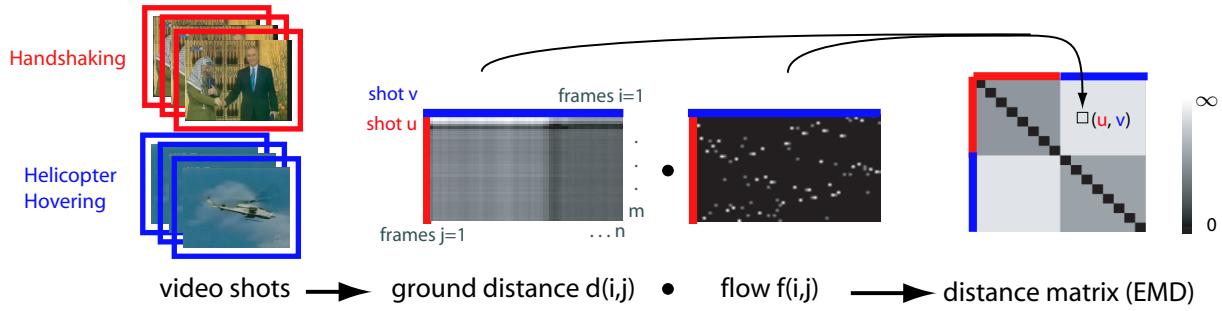

Fig. 1. Computing the Earth Mover's Distance between two shots.

weighted by the *flow* f_{ij} between any two clusters i and j:

$$d(u,v) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \quad (1)$$

Where an optimal flow matrix $f_{i,j}$ is obtained from the following linear program:

$$\begin{array}{ll} \min & \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij} \\ w.r.t. & f_{ij},\ 1 \le i \le m,\ 1 \le j \le n \\ s.t. & f_{ij} > 0, \sum_{j=1}^{n} f_{ij} \le w_{u_i},\ \sum_{i=1}^{m} f_{ij} \le w_{v_j}, \\ & \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\{\sum_{i=1}^{m} w_{u_i}, \sum_{j=1}^{n} w_{v_j}\} \end{array} \quad (2)$$

Intuitively, this linear program solves for the best matching frame pairs in two collections, while the weight-normalization constraints ensure that each frame has enough match in the other collection. The EMD matrix can be transformed into a kernel with the exponential function as shown below, and Support Vector Machine (SVM) is used to learn a separation between shots that belong to the event classes and others. The hyper-parameter A is set to κA_0, where the normalization factor A_0 is the mean of all distances between all training shots. We obtain the optimal scaling factor κ from cross-validation.

$$K(u,v) = exp\{-\frac{1}{A}d(u,v)\}$$

4. COMBINING AND SELECTING AMONG MULTIPLE CUES

In visual recognition applications we often have more than one type of cues from the data. They can come in the form of different types of descriptors, such as color-correlogram or semantic concepts, or in the form of different types of feature design from common features, such as the choices for modeling time and computing similarity in Sections 3 and prior work [3]. Two questions naturally arise: (1) Can we collectively use these multiple cues to make better prediction of the concept? (2) Can we simultaneously learn the importance of each of the input cues?

We consider multiple cue fusion in the context of SVM-like kernel classifiers, i.e., linear fusion for learning a linear discriminant in a high-dimensional feature space as shown in Fig. 2(a). Denote the pool of training shots as v_i, $i = 1, ..$, the collection of k different kernels as $K_j(\cdot,\cdot)$, $j = 1, ..k$. There are several popular practices for this task [11].

Fig. 2(b) depicts "early-fusion", i.e., concatenating input vectors or averaging the different kernel values to arrive at a single kernel $\bar{K}(v_i,\cdot)$, and then learn a single SVM for class separation. Denote the support vector weights as α_i, the decision function for a test example $\hat{v}$ is then written as

$$\hat{y} = \sum_i \alpha_i \bar{K}(v_i, \hat{v}). \quad (3)$$

Fig. 2(c), nick-named "late-fusion", corresponds to learning k SVMs independently and average the decision values, with $\alpha_{i,j}$ the kernel-specific support vector weights, in this case the decision value is computed as in Equation (4).

$$\hat{y} = 1/k \sum_j \sum_i \alpha_{ij} K_j(\hat{x}, x_i). \quad (4)$$

These fusion schemes has two notable drawbacks: (1) neither take into account the relative importance among different kernels, (2) the "late fusion" requires k rounds of training for different SVMs, leading not only to increased computational requirements in training time, but also a larger trace of the model that increases the classification time and memory requirements. It is also possible to learn another layer of SVM for kernel weights on the decision values from the individual SVMs, however this not only increases the computational complexity, but also needs to stratify the training data and is more prone to over-fitting.

To complement the existing fusion schemes in these two aspects, we explore the Multiple Kernel Learning (MKL) decision function in the form of Equation (5) and Fig. 2(d) for multi-cue fusion in visual recognition, i.e., learning linear weights μ_j among the kernels $j = 1, ..k$ with shared support vector weights α_i.

$$\hat{y} = \sum_j \sum_i \mu_j \alpha_i K_j(\hat{x}, x_i) \quad (5)$$

Proposed recently by Bach and Jordan [2], this decision function can also be viewed as one SVM with support vector weights α_i over a "hyper-kernel" $\sum_j \mu_j K_j(\cdot, v_i)$. Compared to the

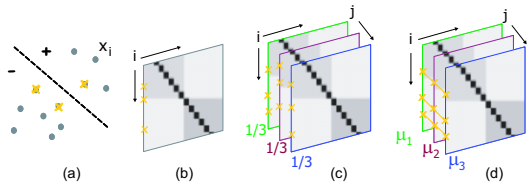

Fig. 2. Learning class discrimination with multiple kernels. Yellow crosses ($\times$) denotes support vectors; red, green and blue denotes different kernels and their weights. (a) Linear classifier in the feature space. (b) A single SVM, or averaging kernels. (c) Averaging multiple SVMs. (d) Multiple Kernel Learning with shared support vectors and learned kernel weights.

early and late-fusion schemes, the number of parameters of MKL is close to those of the early fusion, and the set of kernel weights naturally lends to interpretations of the result.

It is shown [2] that this problem can be formulated in its dual form as Problem (6), i.e., solving for optimal nonnegative linear coefficients $\mu_j \geq 0$ so that the trace of $\sum_{j=1}^{k} \mu_j K_j$ remains constant (chosen to be equal to $d = \mathrm{tr}(\sum_{j=1}^{k} K_j)$) and so that the soft margin SVM is optimized with respect to this linear combination of the kernel matrices.

$$\begin{aligned} \min \quad & \frac{\gamma^2}{2} - e^\top \lambda \\ \text{s. t.} \quad & \lambda^\top D_y K_j D_y \lambda \leq \frac{\mathrm{tr}(K_j)}{d}\gamma^2 \quad j = 1, \ldots, k \end{aligned} \tag{6}$$

where D_y is the diagonal matrix with the labels y on the diagonal and C is the soft margin penalty parameter determined with cross-validation. This problem can in turn be converted into a standard form of second-order-cone programming, and we obtain its solutions with the convex solver Sedumi [9].

5. EXPERIMENTS AND RESULTS

We test the proposed algorithm on a set of events on a large collection of ∼80 hours of broadcast news videos in English, Chinese, and Arabic from October to November 2004, known as the *development* corpus of the TRECVID-2005 [10] benchmark. For each video shot we extract frame sequences and generate multi-modal input streams. The frame sequence contains ten frames per second from the original video. On each frame we extract the confidence value of 33 visual concepts obtained from an SVM classifier trained on color features. These 33 concepts include program categories, scenes, settings, people, objects and activities, which were designed to span the common semantics in news videos [7]. The concept models were produced as part of the TRECVID-2005 [10] benchmark. There are over 69,000 shots in the TRECVID-2005 collection, from which we hold out one-third as the test set, and use the rest for training.

This corpus was manually annotated with a Large-Scale Concept Ontology for Multimedia (LSCOM [7]) of over 400 visual concepts, including scene, people, events, actions and objects. We chose a subset of 13 visual events and actions (Fig. 3) for recognition. These concepts were chosen as they have non-negligible occurrences in the entire corpus, and that they have intuitive ontological relationships with the 33 input concepts on which automatic detection algorithms work reasonably well. We measure Average Precision (AP) [10] at rank 1000 from the prediction values $\hat{y}$.

We compare different features design for the temporal variations within a shot. For the bag-of-features representation we cluster shots with more then 20 frames into 20 clusters with the K-means algorithm; we treat each frame in a shot with less than 20 frames (<2 seconds) as a singleton cluster of equal weights. Euclidean distance is used as the ground distance, and then we solve the EMD program Eq.(2) to obtain the distance between the shot-pairs. SVM is learned on the kernel matrix induced from the EMD matrix using Eq.(3) the soft-margin parameter and the width of the exponential from cross-validation. This approach is compared to the static baseline where SVM is learned on the RBF kernel from a single concept vector computed from the keyframe of the shot, and the HMM-based approach where one HMM is trained on each of the 33 concept dimensions and state-occupancy histogram is used to obtain the SVM input. From the results in Fig. 3 (right) we can see that both HMM and EMD-based representations significantly outperform the static baseline, with EMD more than doubling the average precision from the baseline approaches. The reason for this performance gain is in that the solution to the EMD linear program is sparse, and this puts emphasis on a small set of the best pairwise matching between two frame collections instead of on the aggregate statistics such as the mean of point clouds such as used in HMM or approximated by the keyframe of a shot.

We test MKL as an input selection mechanism. We compute one RBF kernel from the state-occupancy histogram generated by the HMM from each of the 33 input concepts [3], and we use MKL to compute a set of kernel weights as well as support sample weights for classification. Fig. 3 (right) shows that the mean average precision over all events are improved by 5% with MKL compared to an early fusion of HMM-based features. Fig. 3 (left) shows the weights assigned to the 33 input concepts for the 13 target concepts. While the weights are inevitably noisy due to sparse training examples and small training set size, we can nonetheless spot reasonable trends and phenomena in in this weight matrix across both the input kernels and the output events. If we look horizontally and rank the kernels by their weights for predicting a particular target concept, we can see that *street-battle* is mostly predicted by *court, military, people-marching, truck, explosion-fire* with over 50% of the total kernel weights, concurring with intuitive ontological correlations between the input and the target concepts. Looking vertically at each input concept, we can see that *airplane* gets the highest weight for predicting *airplane flying* while its weights for other events are negligible; concept *desert* did not get a high weight for any of the

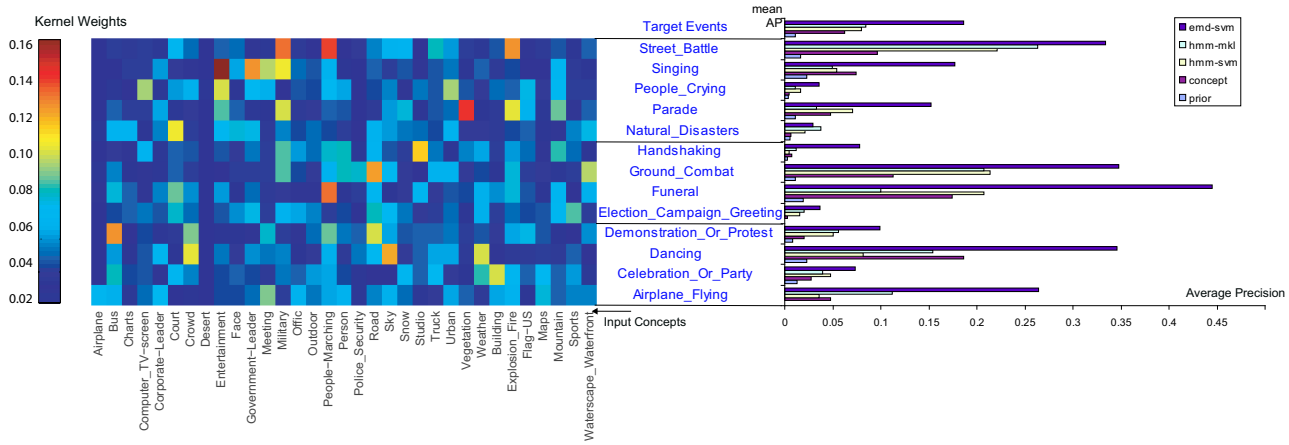

Fig. 3. (right) Average Precision of the different approaches. (left) Input concept weights given by MKL. X-axis: 33 input concepts; Y-axis: 13 target concepts.

target events since its detection performance is low [1].

It is worth noting that the memory and computation load of EMD and MKL has increased compared to SVM. Both requiring storing kernels of size $O(n^2)$ or $O(kn^2)$, with k the number of kernels and n the number of training examples. While MKL can increase average precision by $10 \sim 200\%$ for 8 out of the 13 target events, yet it sometimes degrades the performance compared to early- and late- fusion. This is because some target events exhibit very different patterns in different input dimensions, where trading off the flexibility in k sets of support vector weights with the kernel weights hurts the prediction.

6. CONCLUSION

We study the problem of generic visual event detection and propose two new algorithms from our recent work [3]. In representation we use the earth-movers' distance over bags-of-features, in decision learning we use multiple kernel learning to select and combine the information from multiple kernels. Both techniques show very promising performance while providing intuitive explanations on which input concepts are important for predicting which output concepts. Future work include the automatic prediction of when a particular presentation or fusion scheme works for which target concepts, as well as investigating unifies fusion schemes that solves separate kernel weights and support vectors in one single program.

7. REFERENCES

[1] A. Amir, et., al, "IBM Research TRECVID-2005 video retrieval system," in *NIST TRECVID Workshop*, Gaithersburg, MD, November 2005.

[2] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *ICML*, Banff, Alberta, Canada, July 4–8 2004.

[3] S. Ebadollahi, L. Xie, S.-F. Chang, and J. R. Smith, "Visual event detection using multi-dimensional concept dynamics," in *Intl. Conf. on Multimedia and Expo (ICME)*, Toronto, Canada, July 2006.

[4] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, 1999, pp. 50–57.

[5] S. Hongeng and R. Nevatia, "Multi-agent event recognition," in *In the Proceedings of IEEE Intl. Conf. on Computer Vision (ICCV'01)*, vol. 2, 2001, pp. 84–91.

[6] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000.

[7] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis, "Large-scale concept ontology for multimedia," *IEEE Multimedia Magazine*, vol. 13, no. 3, 2006.

[8] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *Intl. Journal of Computer Vision*, 2000.

[9] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, pp. 625–653, 1999.

[10] The National Institute of Standards and Technology (NIST), "TREC video retrieval evaluation," 2001–2006, http://www-nlpir.nist.gov/projects/trecvid/.

[11] B. L. Tseng, C.-Y. Lin, M. R. Naphade, A. Natsev, and J. R. Smith, "Normalized classifier fusion for semantic visual concept detection." in *ICIP (2)*, 2003, pp. 535–538.

[12] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden markov models," *Pattern Recogn. Lett.*, vol. 25, no. 7, pp. 767–775, 2004.

[13] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Features and kernels for classification of texture and object categories: An in-depth study," *Intl. Journal of Computer Vision*, To appear.