A Probabilistic Representation for Efficient Large Scale Visual Recognition Tasks

Subhabrata Bhattacharya[1], Rahul Sukthankar[2,1], Rong Jin[3], and Mubarak Shah[1]

[1]Computer Vision Lab, University of Central Florida, Orlando, FL

[2]Intel Labs and Carnegie Mellon University, Pittsburgh, PA

[3]Michigan State University, East Lansing, MI

subh@cs.ucf.edu, rahuls@cs.cmu.edu, rongjin@cse.msu.edu, shah@eecs.ucf.edu

Abstract

In this paper, we present an efficient alternative to the traditional vocabulary based on bag-of-visual words (BoW) used for visual classification tasks. Our representation is both conceptually and computationally superior to the bag-of-visual words: (1) We iteratively generate a Maximum Likelihood estimate of an image given a set of characteristic features in contrast to the BoW methods where an image is represented as a histogram of visual words, (2) We randomly sample a set of characteristic features instead of employing computation-intensive clustering algorithms used during the vocabulary generation step of BoW methods. Our performance compares favorably to the state-of-the-art on experiments over three challenging human action and a scene categorization dataset, demonstrating the universal applicability of our method.

1. Introduction

Automatic visual classification for content-based semantic interpretation of images and video remains an active area of research in computer vision. Canonical examples of such tasks include distinguishing an image of an urban scene containing buildings and street lights from that of a natural scene containing mountains, or detecting a particular type of human action (like running) observed in a video. Earlier approaches [4, 11, 22] have demonstrated the utility of constructing representations based on local features in images [16] and video [5, 9], analogous to words in text documents, enabling researchers to apply algorithms from text retrieval and classification to computer vision. These methods, popularly termed as "bag-of-words" (BoW) algorithms, advocate the creation of a vocabulary based on a clustering of visual words extracted from a corpus of images. A new image can then be expressed as a histogram (bag) of words using the designated vocabulary, thereby rendering it suitable for categorization using a classifier such as an SVM.

There have been several innovations in the traditional bag-of-words model that have been used for several visual classification tasks. These advances could be categorized broadly into two levels: representation and classification. At the representation level, Jurie and Triggs [8] show that the clustering process (usually k-means) required during vocabulary generation, is only capable of encoding regions rich in descriptor space. They introduce a radius-based clustering that is capable of generating better codebooks for general scenes. The authors of [26] propose an algorithm for learning a compact visual vocabulary through an iterative pair-wise merging approach, resulting in visual words described by Gaussian Mixture Models (GMMs). GMMs are also employed in the construction of adaptive class-specific vocabularies by Perronnin *et al.* [20] and Farquahar *et al.* [6]. Using hidden topic learning models, Bosch *et al.* [3] introduce a novel vocabulary construction technique that represents each image with a topic distribution vector. Inspired by the success of generative techniques like pLSA in [3], Perronnin *et al.* apply Fisher kernels [19] to image categorization. Furthermore in [13], the authors introduce a method based on maximization of mutual information to group semantically similar visual words resulting in an efficient vocabulary. Tuytelaars and Schmid [23] discretize the high-dimensional space of image features using an optimal lattice structure to create a compact bag of visual words representation for images.

At the classification level, Grauman and Darrell [7] present a pyramid match kernel function that maps unordered feature sets into a higher-dimensional space of multi-resolution histograms, projecting the classification problem into a weighted histogram intersection in that space. This approach is further adapted by Lazebnik *et al.* [10] to spatial pyramid features that preserve a rough spatial information within the codeword, which is beneficial for classification using a histogram intersection kernel.

Methods such as [12, 17, 27] are also popular, where the classification stage is not independent on the representation stage. [17] uses an ensemble of randomized trees for codebook generation as opposed to expensive clustering, followed by employing a tree-based classifier for the recognition task.

One of the problems with the codebook approach, analyzed by [2, 25], is the hard assignment of cluster centers to the visual words in an image which is performed while generating the vocabulary. To this extent Van Gemert *et al.* [25] propose a method to model ambiguity in assigning codewords to images, thereby improving classification accuracy for natural images that have large variation in appearance. In our approach, we circumvent this problem by creating a representation that maximizes the likelihood of generating the visual words corresponding to an image using a kernel density estimator. In fact, Van Gemert *et al.*'s soft-assignment representation becomes a special case of our proposed method, where our representative visual words are set to the cluster centers from a pre-defined codebook and the algorithm terminated after a single iteration.

Our approach also bears some philosophical resemblance to [6], wherein the authors first associate GMMs with each visual word, whose parameters are iteratively tuned using an expectation maximization algorithm. However, their approach suffers from overfitting, for which they need to apply additional regularization techniques. Our approach (as we show in the following sections) has fewer parameters, is guaranteed to converge to a global optimum, is not prone to overfitting and does not require explicit regularization.

Besides providing a better representation, in contrast to [8, 10, 13, 25], the proposed approach does not require expensive data clustering and therefore is computationally efficient, particularly when the number of datapoints is large. While our proposed method can certainly utilize any existing codebook, we show that the anchors in our maximum likelihood representation can also be simply initialized on a randomly-sampled unique set of visual words from a given dataset.

Our primary aim in this paper is to propose a universal representation for images and videos that is based on sound statistical principles (maximum likelihood estimate of observed visual words in the given image). It inherits the benefits of soft-assignment [25] and is made computationally efficient through the use of bounded-support kernels and sampling-based (rather than clustering-based) anchor generation. Importantly, our representation is completely compatible with the existing classifier machinery used in bag-of-visual words approaches, enabling it to be easily integrated into existing real-world image and video recognition systems. Our experiments show the broad applicability of our representation to both image and video domains; wherever possible, we follow existing experimental methodology and avoid the temptation of tuning parameters to maximize performance on the dataset. Thus, our contribution is that of a novel representation rather than the development of a complete system for either scene or action recognition.

The rest of this paper is organized as follows: An overview of our approach is provided in Section 2. In Section 3, we discuss our experiments on a widely accepted scene dataset [10] coupled with in depth analysis of our representation framework followed by more experiments on two challenging human-action datasets. We conclude our paper in Section 4 with a summary of our observations and some pointers towards future work.

2. Proposed Framework

Let D_i be the set of visual features extracted from the i-th image I_i in a large collection of M labeled images $I \equiv \{I_1, I_2, \ldots, I_i, \ldots, I_M\}$. Thus, each D_i could be interepreted as a set of m-dimensional feature vectors whose cardinality may vary from image to image depending on the number of features extracted per image. Let us also denote by D the collection of all features extracted from all labeled training samples (I).

Consider a universal vocabulary of N representative visual features ($\{\mathbf{C}_j\}_{j=1}^{N}$), termed *anchors*. These anchors could be generated using traditional clustering or (as we suggest) sampled directly from D. Our proposed model assumes that visual features are generated i.i.d. from some unknown distribution specified by a set of image-level parameters. Thus, we can express the probability of observing a particular feature $\mathbf{d}$ given an image I_i as:

$$p(\mathbf{d}|I_i) = \sum_{j=1}^{N} w_j K(\mathbf{d}, \mathbf{C}_j), \tag{1}$$

where $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_N)$ are the image-level parameters (weights) that control a kernel density function with kernel $K(.,.)$. In the proposed formulation, these weights $\mathbf{w}$ serve as the image representation and estimating them from the observed features is the primary task.

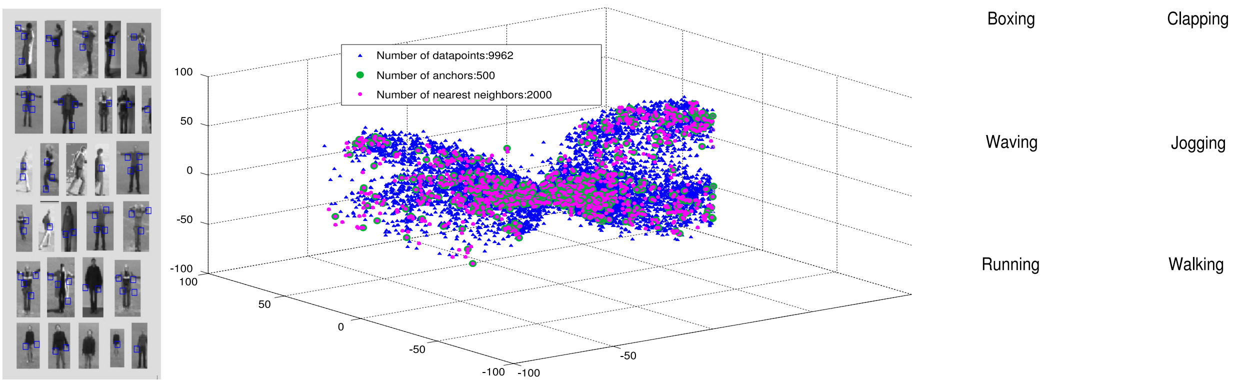
We propose determining $\mathbf{w}$ using a maximum likelihood estimator:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w} \in \Delta} L(I_i, \mathbf{w}), \tag{2}$$

where $\Delta = \{\mathbf{w} \in R_+^N : \mathbf{w}^\top \mathbf{1} = 1\}$ denotes all possible probability distributions for $\mathbf{w}$ and,

$$L(I_i, \mathbf{w}) = \sum_{p=1}^{k} \log \sum_{j=1}^{N} w_j K(\mathbf{d}_p, \mathbf{C}_j).$$

k is the number of features extracted from image I_i. Eqn. (2) being a convex optimization problem, has solutions that are globally optimal.



(a) (b) (c)

Figure 1: An illustration of the proposed representation using the KTH human action dataset as an example. 1(a) Low-level feature extraction: Spatio-temporal features are extracted from input video sequences. 1(b) N anchors (shown as red triangular markers) are selected from the set of all video words (shown as blue triangular markers). Each feature contributes to nearby anchors (shown as green spheres), but in a manner that maximizes likelihood over the entire video. 1(c) A horizontally truncated sparse matrix 20 × 100 (originally 20 × 5000) corresponding to each of 20 training instances from each of the 6 classes is shown.

We propose the following computationally efficient iterative approach based on bound optimization that converges to the maximum likelihood representation. Let $\mathbf{w}'$ be the solution to Eqn. (2) at the current step and $\mathbf{w}$ be the solution at the next step, then $L(I_i, \mathbf{w}) - L(I_i, \mathbf{w}')$ is bounded as:

$$L(I_i, \mathbf{w}) - L(I_i, \mathbf{w}') \;=\; \sum_{p=1}^{k} \log \left[\frac{\sum_{j=1}^{N} w_p K(\mathbf{d}_p, \mathbf{C}_j)}{\sum_{j=1}^{N} w'_p K(\mathbf{d}_p, \mathbf{C}_j)} \right]$$

$$\geq \sum_{p=1}^{k} \sum_{j=1}^{N} \frac{w'_j K(\mathbf{d}_p, \mathbf{C}_j)}{\sum_{l=1}^{N} w'_l K(\mathbf{d}_p, \mathbf{C}_l)} \log \frac{w_j}{w'_j}. \tag{3}$$

The above bound can be easily verified by using Jensen's inequality for convex functions. By optimizing the lower bound in Eqn. (3), we have the following equation for updating w_j at each iteration:

$$w_j = \frac{1}{Z} \sum_{p=1}^{k} \frac{w'_j K(\mathbf{d}_p, \mathbf{C}_j)}{\sum_{l=1}^{N} w'_l K(\mathbf{d}_p, \mathbf{C}_l)}, \tag{4}$$

where Z is a normalization term that guarantees $\sum_{j=1}^{N} w_j = 1$. Note that an approximation to Eqn. (4) can be obtained by initializing each of the elements of $\mathbf{w}$ to $1/N$, leading to a good solution even after just a single iteration, as:

$$w_j = \frac{1}{k} \sum_{p=1}^{k} \frac{K(\mathbf{d}_p, \mathbf{C}_j)}{\sum_{l=1}^{N} K(\mathbf{d}_p, \mathbf{C}_l)}. \tag{5}$$

Given a codebook, Eqn. (5) is thus equivalent to the familiar soft-assignment representation proposed by [25].

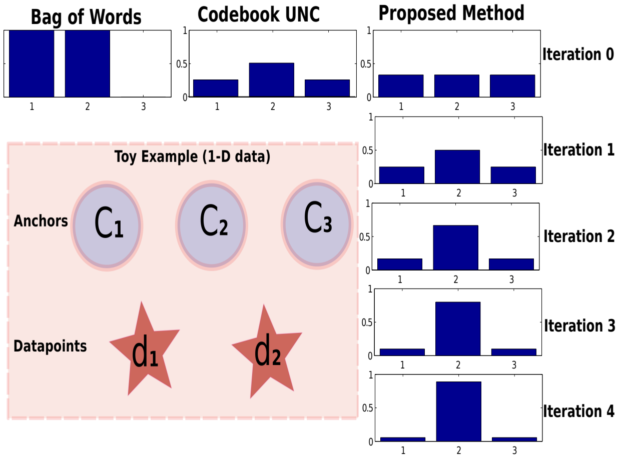


Figure 2: A toy example contrasting the proposed representation against traditional BoW and soft-assignment BoW (Codebook Uncertainty [25]). Note that the proposed representation is initially identical to soft BoW but diverges since it maximizes an image-level likelihood score.

Toy Example: To contrast the proposed representation against traditional BoW and soft-assignment variants of BoW (such as Codebook Uncertainty [25]), we present a toy example with an "image" containing two features in a 1-D space, a vocabulary with three anchors and a uniform ball kernel (see Fig. 2). Traditional BoW simply increments the two bins corresponding to the closest anchors, failing to express the fact that the image contains two similar features. Soft BoW captures this since bin 2 accumulates weight from both features. The proposed maximum likelihood representation seeks anchor weights that optimize the likelihood at an image level (rather than simply accumulating weights).

As a result, bin 2 continues to accumulate a greater fraction of weight, resulting in a stronger peak for the shared feature. Algo. 1 summarizes the entire procedure. In practice, even on real data, the algorithm converges in just 3–5 iterations.

Kernel function: For brevity, let us drop the indices from the data-point $\mathbf{d}_p$ and the anchor $\mathbf{C}_j$, to understand the kernel function (K) in detail. A natural choice for a kernel $K(.,.)$ is the Gaussian:

$$K(\mathbf{d}, \mathbf{C}) = \frac{1}{\sqrt{2\pi}r} \exp(-\|\mathbf{d} - \mathbf{C}\|_2^2 / 2r^2). \tag{6}$$

However, such soft-assignment representations can be unwieldy for large image and video collections because the unbounded support of the Gaussian kernel implies that each visual feature in the image affects the weight corresponding to every anchor. For this computational reason, we advocate the use of bounded support kernels such as a truncated Gaussian or even the simple hyper-ball kernel, which corresponds to a uniform probability of observing a feature in a fixed radius neighborhood of an anchor:

$$K(\mathbf{d}, \mathbf{C}) = \begin{cases} 1 & \text{if } |\mathbf{d} - \mathbf{C}| \leq r, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Such a kernel function can be efficiently computed on a large set of anchors, particularly when paired with an approximate nearest neighbor algorithm [1, 18]. Also, due to the binary nature of the kernel function, some of the keypoints will not be quantized by the centers and are therefore ignored in the computation.

Fig. 3 illustrates the factors that affect the computation of the weights for any given image using Algo. 1. The anchors that are input to the algorithm can either be taken from a standard clustering-based vocabulary or selected from the ensemble of visual words using random sampling, with a uniqueness constraint to ensure better initialization.

3. Experiments

We conducted several independent sets of experiments on a standard scene dataset and two widely popular video datasets, namely Scene-15 dataset [10], KTH Human Actions [21] and UCF [15] action datasets. In addition, we used an aerial video dataset that has recently been released by the DARPA VIRAT program. For all these datasets, anchors are generated by sampling $1.2\%, 2.5\%, 5\%, 10\%, 20\%, 40\%$, and 80% of the total number of features in their individual feature ensembles (S).

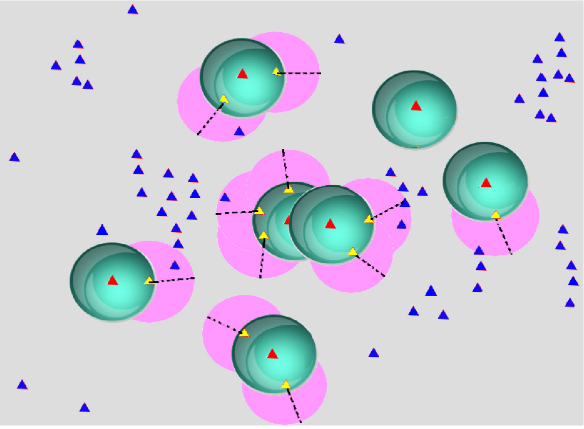


Figure 3: A schematic diagram of the proposed procedure. Blue triangular markers indicate all data points (S). Yellow markers denote the *anchors* (C). Purple circles centered at the anchors signify the m-sphere (β_j) that is constructed using the simple kernel function in Eqn. (7). Datapoints within these spheres, which have anchors in their r-neighborhood (represented by a green sphere), are indicated as red markers. These datapoints can be viewed as contributors to the representation of the datapoint $\mathbf{d}_p$ through the denominator of Eqn. (5).

```
Procedure ComputeWeights ($C, d_p, r$)
Input: Set of $N$ anchors ($C$), Set of $M$ Interest Points ($d_p$) from
       p-th instance $I_p$, Radius of influence ($r$)
Output: Set of weights $\mathbf{w}$
$\mathbf{w}' \leftarrow \mathbf{0}$;
while not converged do
    for $j = 1 \dots N$ do
        $n \leftarrow 0\ i \leftarrow 1\ w[j] \leftarrow 0$;
        for each $d_p[i] \in \{||C[j] - d_p|| \leq r\}$ do
            $S_l \leftarrow 0\ l \leftarrow 0$;
            for each $C[l] \in \{||d_p[l] - C|| \leq r\}$ do
                $S_l = S_l + w'[l]K(d_p[i], C[l])$;
            $n \leftarrow n + 1$;
            $w[j] \leftarrow w[j] + \frac{w'[j]K(d_p[i],C[j])}{S_l}$;
        Normalize ($\mathbf{w}$);
        $\mathbf{w}' \leftarrow \mathbf{w}$
```

Algorithm 1: Algorithm to compute $\mathbf{w}$ for a set of datapoints extracted from a single image or a video.

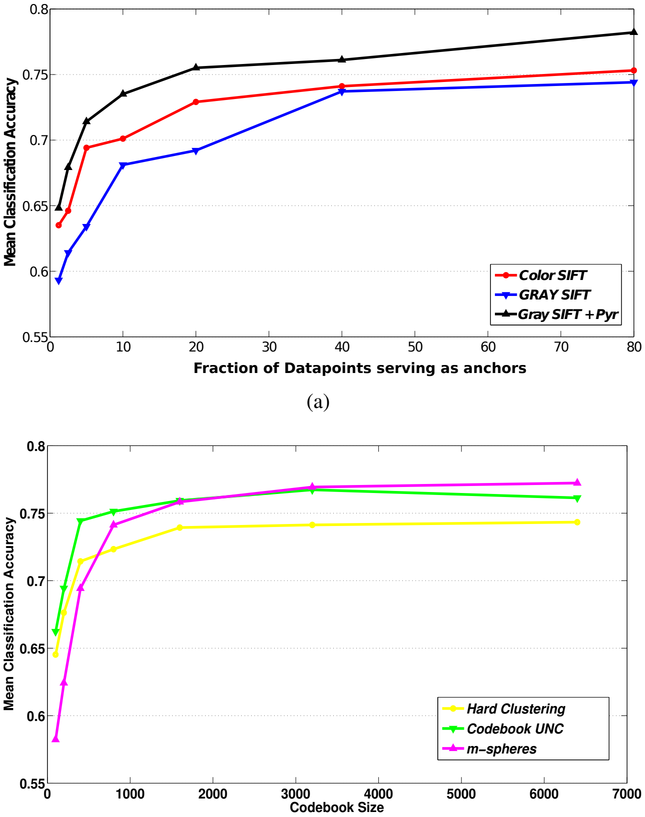
These anchors are input to our maximum likelihood representation, which depends upon the range search technique we employ in Algo. 1, in particular the search radius which corresponds to the radius (r) of m-spheres ($\{\beta_j\}_{j=1}^N$). We use a composite tree indexing scheme (combination of kd-tree and hierarchical k-means) with different search radii to perform the range search required to identify neighbors within the radius of influence of each m-sphere in question. The initial search radius is obtained by computing the average Euclidean distance between a randomly sampled tenth from S. This measure is further refined by increasing it until a situation is reached where all anchors have quantized at least one feature from S.

Classification is performed using a multi-class SVM with a histogram intersection kernel. This kernel has been shown to perform well in conjunction with bag-of-visual words representations on a variety of datasets, including Scene-15 and Youtube in [10, 15, 25].

3.1. Scene-15 Dataset

This dataset consists of a collection of 4,485 images spanning 15 categories, including both natural and man-made scenes. We closely follow Lazebnik *et al.*'s experimental methodology, where we select 100 random images of each category for training and employ the remaining 2,985 images for testing. For all scenes, visual words are extracted using three popular approaches, (a) SIFT [16] on grayscale images, (b) Color SIFT [24], and (c) Gray-SIFT Spatial Pyramid Features [10]. As in Lazebnik *et al.*, we densely sample these descriptors over the image with an 8-pixel stride rather than using an interest-point detector. We use the first two levels of a pyramid with codebook size of 400 while extracting the features, as this was reported to work best.

Fig. 4(a) shows a performance comparison of these three features for different sets of anchors. Consistent with earlier work, visual words based on spatial pyramid features perform better than gray SIFT or color SIFT features alone. Our method achieves $75.5 \pm 0.63\%$ accuracy with only 20% of the total number of visual words. We directly compare the proposed representation with our implementations of: (a) standard codebook model with hard clustering, (b) a soft-assignment model (Codeword Uncertainty [25]) using densely-sampled gray SIFT features. In this setting, the anchors input to Algorithm 1 are replaced by cluster centers returned by k-means clustering algorithm. Our results are shown in Fig. 4(b). For codebook sizes greater than 1600, our method performs better than the hard and soft assigned codebook models. The main computationally intensive step in our method involves determining the memberships of each anchor, which we perform using FLANN [18]. The computation of weights using Algorithm 1 is very efficient. A MATLAB implementation of the alogrithm takes less than 5 secs on a standard laptop.



(a)

(b)

Figure 4: Quantitative analysis of performance of our method in the Scene-15 dataset: 4(a) across different feature modalities, namely gray SIFT, color SIFT and spatial pyramid on top of Gray SIFT on vocabularies created using 1.2%, 2.5%, 5%, 10%, 20%, 40%,and 80% of the total number of datapoints from the dataset. Spatial Pyramid features outperform both gray SIFT and color SIFT features. 4(b) against different vocabulary construction strategies. The sampling of anchors is replaced by k-means clustering. The x-axis indicates the number of clusters chosen starting from 500 to 6500. The yellow curve shows the performance of standard bag-of-visual-words where the representation is a histogram. The green curve corresponds to bag-of-visual-words with soft assignment proposed in [25]. Our method outperforms both methods at codebook sizes greater than 1600.

3.2. KTH Action Dataset

The KTH action dataset [21] is a human action dataset that remains popular in the computer vision community. KTH consists of six sets of actions performed by 25 different human actors under four different illumination scenarios. We handpick a set consisting of 598 action clips from all scenarios for our experiments.

Our low-level features are identical to those employed by recent action recognition methods. Each video clip is represented using a collection of datapoints that are extracted in the following manner: (1) Spatio-temporal cuboids are extracted around regions where the detector proposed by Dollar *et al.* [5] produces maximal responses, only a maximum of 200 cuboids are retained per video, (2) Each cuboid is represented by using normalized gradients descriptors, (3) PCA is applied to reduce the feature vector dimension to 100. Thus each video is represented in terms of about 200 visual words, each described by a 100-dimensional vector.

For classification, we build a training set from 10 randomly-selected actors, actions performed by the remaining 15 actors are used as test set. This is repeated 5 times using a multi-class SVM. Since the feature vectors are extremely sparse (as seen in Fig. 1(c)), the classification is computationally efficient.

The best average classification accuracies for this dataset are achieved with 10,730 anchors, which is 10% of the to-

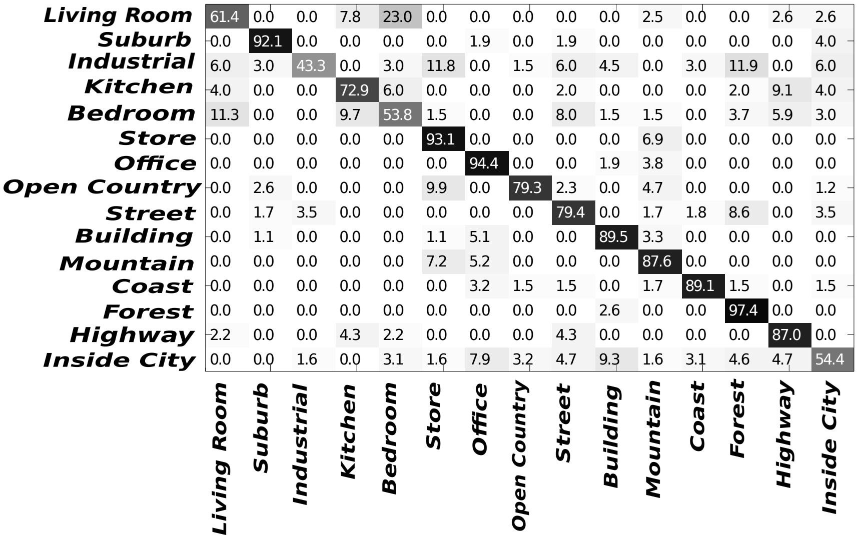


Figure 5: Confusion matrix for the Scene-15 dataset. The results shown here are based on Gray-SIFT Spatial Pyramid features. These results (mean accuracy per class: $78.8 \pm 0.45\%$) correspond to the maximum likelihood representation generated by using 80% of visual words as anchors. With only 20% of anchors we achieve a mean accuracy of $75.5 \pm 0.63\%$ per class.

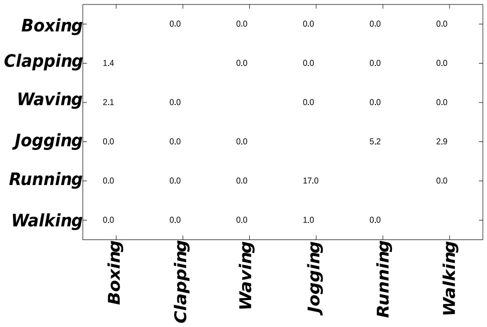


Figure 6: Classification results on KTH action dataset with anchors selected from 10% of the total number of video words (avg. accuracy: $95.06 \pm 0.44\%$). The actions Running and Jogging are most confused because of their visual similarities.

tal number of visual features. Table 1 presents the performance reported by our method and some of the popularly-cited methods in action recognition literature. The accuracy scores are directly imported from the respective authors' papers. A direct comparison is unwise since the experimental methodologies are not identical. However, these results do support our claim that the proposed representation can achieve state-of-the-art performance on standard vision datasets without any explicit tuning. A quantitative confusion matrix is presented in Fig. 6 showing the average classification accuracies of each action category using the same set of 10,730 anchors.

Method	Mean Accuracy (%)
Proposed method	95.06 ± 0.44
Lin *et al.* [12]	95.77
Liu and Shah [14]	94.15
K-means clustering + SVM	88.34

Table 1: Comparison of the proposed method with published results in action recognition on KTH dataset. We also compare our results with a standard hard-clustering Bag-of-video-words technique that uses k-means clustering to construct its vocabulary followed by SVM for classification.

3.3. YouTube Action Dataset

Motivated by the success of our technique on action recognition in KTH, we investigate how our method performs on a newer and more challenging dataset, the YouTube Action Dataset.This dataset is a categorized collection of amateur video clips downloaded from YouTube organized in 11 different categories corresponding to real-world actions such as riding a bicycle/horse/swing, swinging a golf club/tennis racquet, shooting basketball, jumping on a trampoline, juggling a football, diving into a pool, spiking a volleyball and walking with a dog. There are about 100 clips per action. Most of the clips are of poor resolution compared to the KTH data and have noisy and cluttered backgrounds. These clips also exhibit a lot of variation in object scale and viewpoint coupled with significant camera motion. We performed our experiments on the first 10 action instances, distributed over 1051 videos.

Similar to the KTH setup, we extract 400 spatio-temporal volumes from each video, and describe them us-

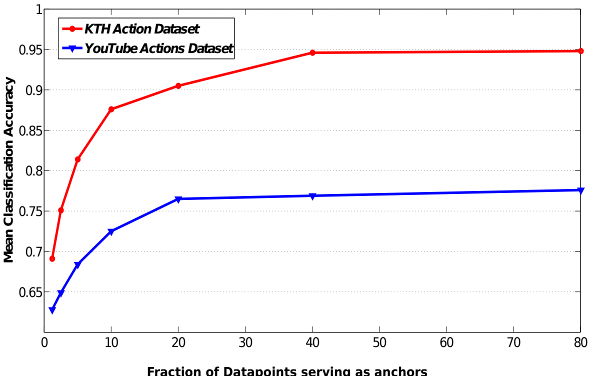
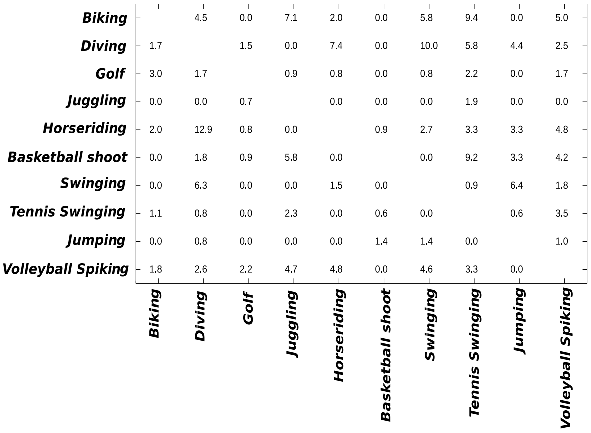


Figure 7: Classification results on KTH and YouTube action datasets as the number of anchors is varied. Our accuracy on YouTube ($76.5 \pm 0.8\%$) with ten classes compares favorably with the state-of-the-art results of [15], who report 76.1% on eight classes.

ing gradient features which are further PCA reduced to 200 dimensions. In this case, our ensemble contains a total of 350,693 visual features. Anchors are selected at different granularities ($1.2\%, \ldots, 80\%$). For each action category, 40 examples are chosen randomly for training, limiting the number of actors to 10. The remaining videos serve as test examples. This process is repeated 10 times. Classification is performed in a similar fashion as covered in the earlier section. As observed in Fig. 7, we achieve 76.5% average classification accuracy when 20% of the visual words serve as anchors, beyond which increasing anchors is not beneficial. This shows that the selected anchors are sufficiently representative to express the important aspects of the videos. The confusion matrix (Fig. 8) confirms that the proposed approach is effective at classifying actions in unscripted real-world video.

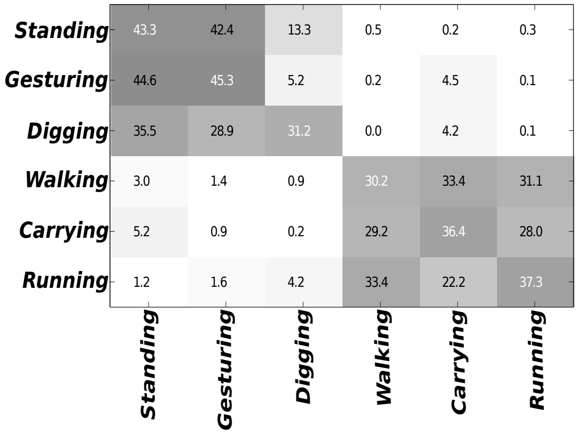


	Biking	Diving	Golf	Juggling	Horseriding	Basketball shoot	Swinging	Tennis Swinging	Jumping	Volleyball Spiking
Biking		4.5	0.0	7.1	2.0	0.0	5.8	9.4	0.0	5.0
Diving	1.7		1.5	0.0	7.4	0.0	10.0	5.8	4.4	2.5
Golf	3.0	1.7		0.9	0.8	0.0	0.8	2.2	0.0	1.7
Juggling	0.0	0.0	0.7		0.0	0.0	0.0	1.9	0.0	0.0
Horseriding	2.0	12.9	0.8	0.0		0.9	2.7	3.3	3.3	4.8
Basketball shoot	0.0	1.8	0.9	5.8	0.0		0.0	9.2	3.3	4.2
Swinging	0.0	6.3	0.0	0.0	1.5	0.0		0.9	6.4	1.8
Tennis Swinging	1.1	0.8	0.0	2.3	0.0	0.6	0.0		0.6	3.5
Jumping	0.0	0.8	0.0	0.0	0.0	1.4	1.4	0.0		1.0
Volleyball Spiking	1.8	2.6	2.2	4.7	4.8	0.0	4.6	3.3	0.0	

Figure 8: Classification results on YouTube action datasets. The mean classification accuracy as determined from the above reaches 76.5%.

3.4. VIRAT Aerial Video Dataset

This is a recently released challenging dataset collected under the DARPA VIRAT program consisting of several human and vehicle activities, captured from a moving aerial platform. In this paper we focus on a subset of the dataset consisting of six human actions. These videos have the following properties that make the action recognition problem in this context more challenging: (1) ego-motion of the camera typically characterized by frequent jitter, (2) extreme low resolution of human actors (50×50 pixels), and (3) large amount of similarity across actions observed from high altitude. For example, the actions standing, gesturing and digging appear similar to each other when viewed from a shaky platform mounted about forty feet above the ground. Similarly, actions such as walking, carrying a box and running can be confused with each other. Each action in the dataset has 200 instances except for gesturing which only has 42 instances.



	Standing	Gesturing	Digging	Walking	Carrying	Running
Standing	43.3	42.4	13.3	0.5	0.2	0.3
Gesturing	44.6	45.3	5.2	0.2	4.5	0.1
Digging	35.5	28.9	31.2	0.0	4.2	0.1
Walking	3.0	1.4	0.9	30.2	33.4	31.1
Carrying	5.2	0.9	0.2	29.2	36.4	28.0
Running	1.2	1.6	4.2	33.4	22.2	37.3

Figure 9: Classification results on VIRAT Aerial Video Dataset. Ambulatory actions can be distinguished from stationary actions, but there is still significant confusion within these broad categories.

We extracted two different types of features using two widely popular spatio-temporal feature extraction implementations. In the first setting we used the methodology similar to the previous two experiments on action datasets. In the second, we used Laptev's STIP [9] implementation, which uses a 3-D Harris corner as a space-time interest point detector, with a 144-dimensional concatenation of Histogram of Gradient and Histogram of Optical Flow descriptors. We represent both sets of datapoints using two techniques —the standard bag of video words and the proposed representation. The classification is performed by an SVM with a histogram intersectionkernel using a 10-fold cross validation, similar to the previous experimentalframework. The best performance in this dataset was observed with 388,322 anchors, which is 40% of the datapoints. In this setting, we achieved the maximum mean accuracy of

37.7% per class. Fig. 9 shows the confusion matrix. On this challenging dataset, we see from the confusion matrix that the ambulatory actions (walking, running and carrying) can be distinguished from the stationary ones (gesturing, digging and standing). However, there is significant misclassification within these broad categories. We also compare our results with two different types of feature extraction schemes and their respective bag of words representations in Tab. 2. The maximum performance for both the representations are empirically recorded to be at the point where the number of anchors (for the proposed method) or codewords (for BoW) versus mean accuracy becomes asymptotic.

	BoW		Proposed	
Action	HOGHOF	PCA-G	HOGHOF	PCA-G
Standing	41.1	39.2	**43.3**	42.2
Gesturing	40.5	41.5	**45.3**	44.9
Digging	**34.9**	34.6	31.2	34.2
Walking	**32.9**	32.6	30.2	31.7
Carrying	35.5	33.7	**36.4**	36.1
Running	34.5	**39.3**	37.4	38.2

Table 2: Comparative results with two different types of spatio temporal feature extraction/description techniques, namely HoG+HoF [9] and PCA-G [5] on two representation schemes: standard bag of words and our proposed method.

4. Conclusion

We present a novel, principled representation for both images and videos that is based on maximizing the likelihood of generating the observed visual words using a vocabulary. We present a computationally-efficient iterative algorithm that identifies the globally optimal parameters. Recent approaches that employ soft assignments are shown to be special cases of our approach, and our method is completely compatible with recognition systems that operate with standard bags-of-visual words representations. Furthermore, we show how the expensive step of clustering visual words to generate a vocabulary can be replaced (for our representation) with a sampling-based approach over visual words without significantly impacting classification accuracy. In future work, we plan to explore how we can better leverage sparsity in our representation and combine the proposed approach with manifold learning techniques.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-08-C-0135 and Army Research Office under Contract No. ARO W911NF-08-1-0403.

References

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1), 2008. 2596

[2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008. 2594

[3] A. Bosch, A. Zisserman, and X. Munoz. Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. PAMI*, 30, 2007. 2593

[4] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *ECCV*, 2004. 2593

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features, 2005. IEEE International Workshop on VS-PETS. 2593, 2597, 2600

[6] J. Farquhar, S. Szedmak, H. Meng, and J. Shawe-Taylor. Improving "bag-of-keypoints" image categorisation. Technical report, University of Southampton, 2005. 2593, 2594

[7] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*, 2005. 2593

[8] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, 2005. 2593, 2594

[9] I. Laptev. On space time interest points. *IJCV*, 64, 2005. 2593, 2599, 2600

[10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2593, 2594, 2596, 2597

[11] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001. 2593

[12] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009. 2594, 2598

[13] J. Liu and M. Shah. Scene modeling using co-clustering. In *ICCV*, 2007. 2593, 2594

[14] J. Liu and M. Shah. Learning human action via information maximization. In *CVPR*, 2008. 2598

[15] J. Liu, Y.Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. In *CVPR*, 2009. 2596, 2597, 2599

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004. 2593, 2597

[17] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, pages 985–992, 2006. 2594

[18] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009. 2596, 2597

[19] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 2593

[20] F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *ECCV*, 2006. 2593

[21] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004. 2596, 2597

[22] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *ICCV*, 2003. 2593

[23] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *ICCV*, 2007. 2593

[24] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Trans. PAMI*, 2010. 2597

[25] J. Van Gemert, J. Geusebroek, J. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008. 2594, 2595, 2597

[26] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *ICCV*, 2005. 2593

[27] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *CVPR*, 2008. 2594