# Ensembles of Novel Visual Keywords Descriptors for Image Categorization

Azizi Abdullah
School of Computer Science
Universiti Kebangsaan Malaysia
Malaysia
azizi@ftsm.ukm.my

Remco C. Veltkamp
Dept. of Information and Computer Sciences
Utrecht University
The Netherlands
Remco.Veltkamp@cs.uu.nl

Marco A. Wiering (*IEEE Member*)
Dept. of Artificial Intelligence
University of Groningen
The Netherlands
mwiering@ai.rug.nl

***Abstract*—Object recognition systems need effective image descriptors to obtain good performance levels. Currently, the most widely used image descriptor is the SIFT descriptor that computes histograms of orientation gradients around points in an image. A possible problem of this approach is that the number of features becomes very large when a dense grid is used where the histograms are computed and combined for many different points. The current dominating solution to this problem is to use a clustering method to create a visual codebook that is exploited by an appearance based descriptor to create a histogram of visual keywords present in an image. In this paper we introduce several novel bag of visual keywords methods and compare them with the currently dominating hard bag-of-features (HBOF) approach that uses a hard assignment scheme to compute cluster frequencies. Furthermore, we combine all descriptors with a spatial pyramid and two ensemble classifiers. Experimental results on 10 and 101 classes of the Caltech-101 object database show that our novel methods significantly outperform the traditional HBOF approach and that our ensemble methods obtain state-of-the-art performance levels.**

## I. Introduction

Object recognition algorithms aim to classify images based on their visual content. During the last decade machine vision systems have become more effective for dealing with the complex problem of handling high dimensional pixel representations. For this most machine vision systems use an image descriptor to extract feature vectors from images which are given to a machine learning algorithm to map the image features to desired class labels. The most widely used image descriptor is the SIFT descriptor [1] that describes an image using a histogram of pixel gradient orientations. Although the original SIFT algorithm [1] consists of a method to extract salient keypoints next to the descriptor, many recent machine vision systems [2], [3], [4] replace the keypoint extractor with a grid consisting of gridpoints at regular intervals so that the whole image content is represented.

Using the SIFT descriptor on many points of a dense grid in an image leads to very large feature representations that are more complex to handle with a machine learning algorithm. Therefore, the bag of visual keywords representation has been proposed [5]. This method *can* work with dense grids without increasing the dimensionality of the resulting feature vectors. This method consists of the following steps: (1) Extract patches (small parts of an image) and compute their feature vectors using a visual descriptor, (2) Cluster the feature vectors to create a visual codebook, (3) Represent an image using a histogram of visual keywords by using the codebook together with the feature vectors extracted from the patches. The main idea of this approach is to describe the content of images by a histogram of an orderless collection of visual words, similar to the bag-of-words (BOW) representation that shows very good performance for classifying text documents [6].

The hard bag-of-features (HBOF) [5] approach can be considered as the most often used method for creating the visual keywords histogram. In the HBOF approach the keyword histogram is computed by following a winner take all scheme, also referred to as "hard assignment". In this scheme, each image patch is used for incrementing a value of a *single* cluster in feature space, or keyword in the visual codebook. The resulting HBOF histogram therefore only contains the frequencies of winning cluster centroids to represent an image. In the literature, experimental results have shown that labeling each region by its nearest cluster center only, is not an optimal choice [7], [8], [9]. In HBOF other cluster centroids are ignored to describe the frequency distribution of visual keywords that occur in images, whereas other cluster centers also contain specific features that can enhance the complete description of images. Thus, a number of novel bag of visual keywords methods have been proposed [7], [8], [9] that use a "soft assignment" as an improved way for describing images. A rather different way of using the visual codebook was developed in the HMAX system [10]. In the last stage of the HMAX approach a visual keyword receives a value based on its maximal similarity to one of the patches in an image.

**Contributions of this paper.** We present a novel object recognition systems that contributes in several ways to the state-of-the-art in machine vision. (1) We present and evaluate a novel soft assignment method using the codebook model. (2) We describe a novel approach related to the use of image patches by the HMAX architecture, and compare this and the original HMAX method to hard and soft assignment methods, and the use of SIFT without a codebook. (3) We combine all these methods with spatial pyramids [11] and evaluate how much they can profit from the use of multiple levels to describe images. (4) We combine all the used descriptors using two ensemble algorithms consisting of support vector machines (SVMs) [12]. As ensemble methods we use the product and

978-1-4244-7815-6/10/$26.00  

mean rules [13] to efficiently combine the different classifiers. (5) All methods are compared on 10 and 101 images from the Caltech-101 dataset, and the results show that our ensemble methods obtain state-of-the-art performance levels.

The paper is organized as follows. Section II describes previous image descriptors related to our work. After that we describe our novel bag of visual keywords approaches in Section III. Section IV describes the ensemble methods and how we used the support vector machine as classifiers. Experimental results on 10 and 101 image classes of Caltech-101 are presented in Section V, and Section VI concludes the paper.

## II. Related Work

In this section we will first describe the SIFT descriptor [1], since all methods presented in this paper use it to describe image patches or complete images. After that we will describe previous methods that use a codebook to create a visual keywords histogram. We will end with a description of the method used in the HMAX architecture.

### A. SIFT descriptor

There are many types of image descriptors, which rely on features such as color, texture and shapes. Nowadays, the most successful image descriptors extract information about edges and shapes. The best known ones are SIFT [1] and histograms of orientation gradients (HOGs) [14]. The original SIFT algorithm first computes salient points, and then describes the regions around these extracted keypoints using an orientation histogram. In contrast to the use of salient points, we use a fixed partitioning scheme, which is a simpler method with similar performance [15]. Furthermore, using this approach the spatial relationships between the SIFT features can be represented more efficiently. The fixed partitioning method keeps the order of the keypoints always the same, whereas when the SIFT keypoint extraction method is used, the order of image parts is lost. Therefore, the SIFT keypoint extraction method is either combined with a keypoint matching method as in SIFT [1] or with a clustering method [5], [2]. In our case, we are not obliged to use clustering, but can also use the features computed by SIFT at the gridpoints of the fixed partitioning grid.

The orientation histogram we use is computed from a smoothed region in the image. For each pixel intensity in a cell, $I(x,y)$, the gradient magnitude $m(x,y)$ and orientation $\theta(x,y)$ are computed using the differences in pixel intensities:

$$\begin{aligned} I_x(x,y) &= I(x+1,y) - I(x-1,y) \\ I_y(x,y) &= I(x,y+1) - I(x,y-1) \\ m(x,y) &= \sqrt{I_x(x,y)^2 + I_y(x,y)^2} \\ \theta(x,y) &= tan^{-1}\left(\frac{I_x(x,y)}{I_y(x,y)}\right) \end{aligned}$$

where $I_x$ and $I_y$ are image derivatives of $I(x,y)$ for $x$ and $y$ directions respectively.

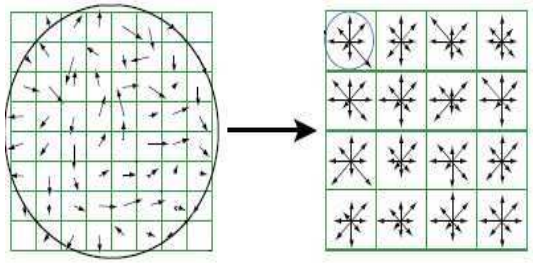

Fig. 1. *Orientation histogram is constructed from a specific region.*

To compute the image descriptor, an input image is first convolved with a Gaussian filter. Then a fixed number of regions to construct the descriptor is generated. After that, the center point of each region is determined by its location and dividing its $width$ and $height$ with 2. The descriptor is then constructed by a circular region around the center point of the region. The circular region radius $r$ is determined by: $r = \sqrt{(\frac{width}{2})^2 + (\frac{height}{2})^2}$. After that, the descriptor breaks apart a window around the center point into $4 \times 4$ sub-blocks and calculates a gradient orientation histogram, whereby each gradient is weighted by its magnitude to better reflect strong orientations. Each histogram has 8 bins and in total there are 128 bins per histogram for each region. Fig. 1 shows the orientation histogram constructed from a given region.

### B. Bags of visual keywords

The bag of visual keywords approach has been widely used and demonstrated impressive levels of performance in image categorization applications [16], [17]. This approach works by clustering local feature vectors such as computed by the SIFT descriptor, extracted from separate regions or patches, into similar group patterns or clusters. The *k*-means clustering algorithm is for example widely used to cluster image features. The *k*-means method is quite fast, simple and has been applied and shown to be useful in many applications. It works by subdividing samples consisting of feature values into a set of clusters, based on the distances between the samples [18]. When applied to image features, this results in a visual codebook. The codebook contains a compact representation of the local image features and is used to build the histogram of visual keywords. There are a number of methods that create the histogram values in different ways.

### C. Hard bag-of-features (HBOF)

In HBOF [5], a winner take all scheme is used, where the cluster centroid which corresponds to the minimum distance to the feature vector of the patch is used to label the specific patch or region. Therefore, HBOF is also termed as a "hard assignment" approach. Using this approach, it is quite common that two similar patches are assigned to different visual words, especially when the size of the visual codebook and the dimensionality of features are increased [7], [8]. Therefore, similar images can be mapped to very different histograms.

The traditional HBOF works with a given vocabulary of visual keywords that are extracted using a clustering method.

After that, the minimum (Euclidean) distance is computed between the codebook cluster centroids and the feature vectors of some image patch to compute a histogram that contains the frequencies of winning visual words. For each visual word *w* in the visual vocabulary $V$, the histogram of visual words is computed as follows:

$$HBOF(w) = \sum_{i=1}^{n} \begin{cases} 1 & \text{if } w = arg\min_c(dist(c, r_i)) \\ 0 & \text{otherwise} \end{cases}$$

where *n* is the number of local regions in an image, $r_i$ is the feature vector computed at local image region *i*, $dist(c, r_i)$ is the (Euclidean) distance between a cluster centroid $c$ and the feature vector $r_i$, and $c \in V$.

### D. Soft assignment methods

Recently, bags of visual keywords with the soft assignment scheme have attracted more attention. This approach is believed to be more efficient than HBOF, because it uses multiple combinations of visual keywords to describe each image patch that allows the complete description of an input image. The main idea of this approach is to give a certain weight to multiple nearby clusters, instead of only to the winning cluster. In [8], the authors proposed a soft-weighting scheme where for each image patch a cluster centroid receives a weight of $\frac{1}{2^{i-1}}$, where *i* is the $i^{th}$ nearest neighbor in the codebook.

Besides this approach, Philbin *et al* [9] uses weights to each cluster centroid according to $\exp(-\frac{d^2}{2\sigma^2})$, where *d* is the distance between the cluster centroid and the feature vector of the image patch. The authors found that the parameter $\sigma$ and the number of nearest neighbors most influenced the image categorization performance. Both approaches have demonstrated significant improvements compared to the hard assignment approach. Following this, a new state-of-the-art soft assignment method called *Codeword uncertainty* (UNC) was proposed [7] that indicated a significant improvement when combining the kernel distances to multiple nearby neighbors. This approach can be defined as follows:

$$UNC(c) = \frac{1}{n} \sum_{i=1}^{n} \frac{K_\sigma(dist(c, r_i))}{\sum_{j=1}^{|V|} K_\sigma(dist(v_j, r_i))}$$

where $K_\sigma$ is the one-dimensional Gaussian kernel. In contrast to [9], given a codeword *c*, UNC normalizes the amount of probability mass and distributes the weight over all codewords.

### E. HMAX visual keywords approach (MAX)

In the hard assignment model, the keyword frequency measures how often the cluster centroid has the minimum distance to one of the patches in the image. A rather different way is proposed in the HMAX architecture [10], which we will compare in our study to other bag of visual keywords approaches. Although the HMAX architecture consists of multiple layers, somewhat mimicking the workings of the visual cortex, here we only consider the workings of layer C2 in the HMAX architecture. Furthermore, in the original HMAX architecture no clustering was applied to compute a visual codebook, but distances to random patches were computed. We will call the method that uses a visual codebook the Max similarity map or simply MAX descriptor.

Given a set of feature vectors computed in the patches of an image, MAX computes the maximum similarity of all patches to a keyword from the codebook and use this similarity in the resulting histogram. Therefore, instead of a competition between cluster centroids, here there is a competition between patches. The resulting feature vector describes how much each keyword is present in the image. The MAX descriptor is described more formally with the following equation:

$$MAX(c) = \max_r(\exp(-\lambda \cdot dist(c, r)))$$

Here, an exponential function is used together with the parameter $\lambda$ to calculate similarity scores between 0 and 1. The $\lambda$ parameter is optimized empirically.

## III. Novel Visual Keyword Descriptors

In this section three novel descriptors based on codebooks will be described. The first method is a novel soft assignment method, the second one is a variant of the MAX descriptor explained before that does not need an additional parameter, and the last descriptor computes histograms for whole images instead of using small patches.

### A. Weighted centroid maps (WCM)

WCM is a soft assignment approach and thus increments multiple keyword counters when examining each patch. WCM uses a ranking scheme where the closest centroid receives the highest increment and centroids not within a predefined number of nearest neighbors do not receive anything. Let **Rank**(*p*, $c_i$) $\in [1, k]$, where $k$ is the number of cluster centroids, be the rank of nearest cluster $c_i$ from the set of cluster centroids, where *p* is an image patch. The clusters having a rank below some number *N* contain the most relevant information. Thus, the weight associated with the centroid $c_i$ for patch $p$ is:

$$W(c_i) = \begin{cases} \frac{(N - \textbf{Rank}(p, c_i)) + 1}{N} & \text{if } \textbf{Rank}(p, c_i) \leq N \\ 0 & \text{otherwise} \end{cases}$$

For each keyword in the codebook all these weights are added up when examining all patches in an image.

### B. Min distance map (MIN)

Our MIN approach is inspired by the HMAX architecture [10] and is quite similar to the MAX descriptor. The problem of the MAX descriptor is that it requires fine-tuning the parameter $\lambda$ to get the best results. The MIN approach computes a minimum distance map without the use of any parameter. The minimum distance map MIN for each visual word $c$ in the visual vocabulary $V$ is computed as follows:

$$MIN(c) = \min_r(dist(c, r))$$

In our experiments the Euclidean distance is used to compute the distances. The size of the descriptor is equal to the number of cluster centroids in the codebook.

### C. Spatial correspondence distance map (SCDM)

In the previous visual keywords descriptors, the image was split up in regions using overlapping or non-overlapping patches. After that, these regions are clustered to produce a codebook. The SCDM does not use patches, but computes a feature vector based on the whole image. It is combined with spatial pyramids [11] to compute spatial correspondences.

One of the simplest and most efficient ways to capture the spatial correspondence is to use the spatial pyramid approach [11]. This approach consists of one global (single level, $L = 0$) and several local regions to describe multiple levels of resolution. The local region numbers are increased with increasing the number of levels by $2^L$, where $L = 0, 1, 2, \ldots, N$. The idea is simply to split up an image in 1, $2\times2$, $4\times4$, etc. local regions and combine them all. Although we use the spatial pyramids with all previously described descriptors in the experiments, for the SCDM it computes spatial correspondence codebooks for all levels independently.

The spatial correspondence distance map is constructed using distances between a (local) region feature vector and the cluster centroids from the spatial correspondence codebooks at multiple resolutions. If $I_L$ is the image feature vector at level $L$, and $C_i(L)$ is a cluster centroid at level $L$ then SCDM computes the following histogram for each level $L$:

$$S_{scdm}(C_i(L)) = dist(I_L, C_i(L)) \tag{1}$$

The method therefore computes a distance map from an image to cluster centroids representing other images, and does this using different pyramid levels.

## IV. Classification Methods

### A. SVM classifier

We employ an SVM [12] to learn to classify the images. The one-vs-one approach is used to train and classify images in the Caltech-101 dataset. For the SVMs, we use Radial-Basis-Function (RBF) kernels in all experiments. Initially, all attributes in the training and testing sets were normalized to the interval [-1,+1]. We did not use the fixed weighting scheme for the spatial pyramid classifier [11]. Our previous experiments [4] indicated that this did not improve the results.

We also need to find the SVM parameters C and $\gamma$ that perform best for the descriptors. To optimize the classification performance, the parameters were determined by using the libsvm grid-search algorithm [19]. We tried the following values $\{2^{-5}$,$2^{-3}$,...,$2^{15}\}$ and $\{2^{-15}$,$2^{-13}$,...,$2^{3}\}$ for C and $\gamma$, respectively. The values which gave the best accuracy performance with 5-fold cross-validation are picked and used to train on the training set.

### B. Ensemble methods for combining classifiers

Our previous research [15], [3], [4] showed that combining multiple features and classifiers with ensemble methods significantly increases classification performance. Ensemble methods have received considerable attention in the machine learning community to increase the effectiveness of classifiers. In order to construct a good ensemble classifier, the ensemble needs to construct accurate and diverse classifiers and to combine outputs from the classifiers effectively [20]. There exist several methods to obtain and combine the diverse classifiers. Here we employ two ensemble algorithms namely (1) product rule and (2) mean rule [13].

The product rule is one of the simplest and most efficient ways for combining outputs of classifiers [13]. When the classifiers have small errors and operate in independent feature spaces, it is efficient to combine their (probabilistic) outputs by multiplying them. Thus, we use this product rule to determine the final decision of the ensemble. First the posterior probability outputs $P_j^k(x^k)$ for class $j$ of $n$ different classifiers are combined by the product rule:

$$P_j^p(x^1, ..., x^n) = \prod_{k=1}^{n} P_j^k(x^k) \tag{2}$$

where $x^k$ is the pattern representation of the $k^{th}$ descriptor. Then the class with the largest probability product is considered as the final class label belonging to the input pattern.

When estimators of the different classifiers contain large errors, it can be more efficient to combine their estimated probabilities by the mean rule [13] as follows:

$$P_j^m(x^1, ..., x^n) = \frac{1}{n} \sum_{k=1}^{n} P_j^k(x^k) \tag{3}$$

Similar to the product rule, the class with the largest probability mean is considered as the final class label.

In the experiments we will compare these ensemble methods to the naive approach that combines the feature vectors computed at all spatial resolution levels in one large feature vector.

## V. Experiments and Results

### A. Caltech dataset

Our experiments contain two stages. In the first stage, 10 categories were selected and a total of $10 \times 30 = 300$ images for evaluation. The first ten categories were as follows: airplane, cameras, cars, cell phones, cups, helicopters, motorbikes, scissors, umbrellas, and watches. All images are in JPEG format with medium resolution (about $300 \times 300$ pixels). Based on results of the first stage, we extended the experiment to all categories of the dataset (Caltech-101). Fig. 2 shows some images of the Caltech-101 dataset with large intra-class variations.

In order to evaluate the described approaches, we used the region of interest (ROI) taken from [2] for our images. For evaluating the combination methods and the other single descriptors, we used 15 training and 15 testing images for each image class. We chose 10 times different training and test images randomly from a set of candidate images from the 10 and 101 classes of the Caltech-101 dataset. Finally, we report the performances using mean and standard deviation to verify significances of the obtained classification results.

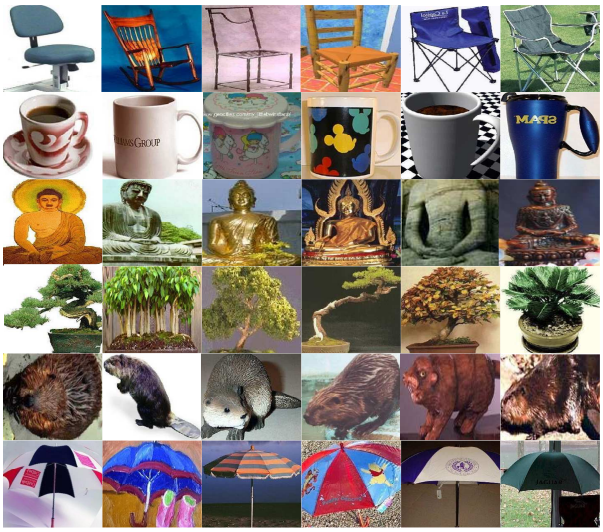

Fig. 2. *Some examples of images from the Caltech-101 dataset with intra-class variations namely chair, cup, Buddha, bonsai, beaver, and umbrella, respectively.*

### B. Experimental setup

For SIFT, we use the maximum angle 180°. We applied Gaussian blur with $\sigma = 1.0$ to smooth the images. Feature vectors are quantized into visual words using $k$-means clustering where we tried $k$=300, 650, 700 and 750. The best value for each descriptor is used to compute the final results. For extracting the patches, we used a rectangular grid of 32 × 32 pixels with spacing of 8 pixels in each image. We used several levels of the spatial pyramid, $L = 0, 1$ and 2.

### C. Results on Caltech-10

Table I shows the average classification accuracy (%) and the standard deviation of the different descriptors to classify images in 10 classes. In our experiments, increasing the number of levels in HBOF and WCM from 1 to 2 made classification performance much worse, thus we do not report their results. In this case, we believe that levels 0 and 1 have sufficiently rich information to describe objects at these levels, and that using too many clusters (like at level 2) leads to less discriminative descriptors. The table clearly shows that the proposed methods (MIN, WCM, and SCDM) outperform the commonly used HBOF approach. This demonstrates that each cluster centroid alone is not the best method to describe the appearance of local regions.

Combining all levels of a single descriptor often improves the performance of the best single level as shown in the last three columns of Table I. The best combination method is the mean rule with the MIN descriptor that achieves an accuracy of 96.2%. The same training and testing images are applied to the state-of-the-art method, UNC, with normalization of feature vectors [7]. The results show that the MIN descriptor works very well and significantly outperforms the other approaches, including the MAX descriptor.

We extended our experiments to combine all classifiers of the different descriptors (except for UNC) on 10 classes. We compare three combination methods with the two ensemble methods (product and mean rules). (1) Combining the classifier output probabilities when the features are combined from all levels. (2) Combining the outputs from classifiers based on features from separate levels (note that this leads to more probabilities that are combined). (3) Combining the outputs from the classifiers using the best single level only. The results are reported in Table II. In this experiment, combining the naive classifiers from Table I with the mean rule gives the best performance of 97.0%. This is probably caused by the fewer and more accurate values that are combined compared to combining all classifiers from separate levels. Furthermore, this method does not throw away information which only combining the classifiers from the best level does.

TABLE I
THE AVERAGE CLASSIFICATION ACCURACY (MEAN AND SD) OF THE DIFFERENT DESCRIPTORS FOR EACH LEVEL AND COMBINATION METHOD ON 10 CLASSES. NAIVE=NAIVE FEATURE COMBINATION METHOD, PR=PRODUCT RULE, MR=MEAN RULE.

| | L0 | L1 | L2 | Naive | PR | MR |
|---|---|---|---|---|---|---|
| SIFT | 79.7 | 89.7 | 89.4 | 91.5 | 91.3 | 91.7 |
| | ±2.5 | ±2.3 | ±3.8 | ±2.1 | ±2.5 | ±2.5 |
| HBOF | 77.7 | 72.1 | - | 78.8 | 75.9 | 76.3 |
| | ±2.4 | ±10.0 | - | ±3.5 | ±6.2 | ±6.9 |
| MIN | 79.1 | 86.9 | 90.7 | 86.7 | 95.5 | **96.2** |
| | ±2.6 | ±5.3 | ±3.6 | ±2.5 | ±3.8 | ±3.7 |
| MAX | 80.1 | 85.0 | 88.2 | 89.0 | 89.6 | 89.5 |
| | ±1.8 | ±2.5 | ±2.4 | ±2.3 | ±1.7 | ±1.6 |
| WCM | 79.2 | 85.9 | - | 84.4 | 84.1 | 83.9 |
| | ±3.0 | ±1.6 | - | ±3.3 | ±1.7 | ±1.7 |
| SCDM | 75.3 | 87.9 | 91.5 | 89.9 | 90.9 | 91.1 |
| | ±1.8 | ±2.0 | ±2.0 | ±1.9 | ±1.4 | ±1.8 |
| UNC | 64.4 | 80.2 | 79.1 | 81.7 | 83.1 | 83.5 |
| | ±3.7 | ±3.4 | ±2.2 | ±2.9 | ±2.5 | ±1.8 |

TABLE II
THE AVERAGE CLASSIFICATION ACCURACY (MEAN AND SD) OF DIFFERENT COMBINATION CLASSIFIERS AND ENSEMBLE METHODS ON 10 CLASSES. M1=CLASSIFIERS BASED ON ALL LEVELS COMBINED, M2=CLASSIFIERS BASED ON SEPARATE LEVELS, M3=CLASSIFIERS BASED ON THE BEST SINGLE LEVEL

| | Product Rule | Mean Rule |
|---|---|---|
| M1 | 96.5±1.3 | **97.0**±1.3 |
| M2 | 93.5±1.9 | 94.1±2.1 |
| M3 | 94.7±1.9 | 95.3±2.1 |

### D. Results on Caltech-101

Based on the Caltech-10 dataset findings, we extend our experiments to the whole dataset. We used the same optimal parameters as in 10 classes for generating feature descriptors and for the $k$-means clustering algorithm. However, the learning parameters for each SVM classifier are adjusted to the need of many categories using libsvm grid-search. Table III shows the average categorization performance of the single combined descriptors on 101 classes, where the naive combination method is used. These results also clearly show that using WCM and methods inspired by the HMAX architecture (MIN and MAX) significantly outperform the standard hard bag-of-features approach, although the immediate use of

SIFT features without using visual codebooks obtains the best performance. We also performed experiments with the UNC approach without normalization of feature vectors (our results with normalization of feature vectors are worse). This confirms that HMAX based visual keywords descriptors and also our weighted centroid maps improve classification performance compared to previous bag of visual keywords descriptors.

TABLE III
THE AVERAGE CLASSIFICATION ACCURACY (MEAN AND SD) OF THE SINGLE DESCRIPTORS ON 101 CLASSES.

| | Naive |
|---|---|
| SIFT | **62.7**±1.3 |
| HBOF | 51.8±1.5 |
| MIN | 57.9±1.0 |
| MAX | 59.0±0.9 |
| WCM | 57.6±1.2 |
| SCDM | 55.1±1.6 |
| UNC | 51.6±0.9 |

Table IV shows that a combination of the descriptors (without UNC) performed very well with an ensemble of support vector machines. It gives $66.8 \pm 1.6$ with the mean rule on 101 classes of the Caltech 101 dataset.

TABLE IV
THE AVERAGE CLASSIFICATION ACCURACY (MEAN AND SD) OF USING THE SINGLE CLASSIFIERS AND ENSEMBLE METHODS ON 101 CLASSES

| | Product Rule | Mean Rule |
|---|---|---|
| M1 | 66.6±1.2 | **66.8**±1.6 |

## VI. CONCLUSIONS

In this paper, we have introduced several novel approaches for exploiting visual codebooks. We have reported a significant comparison between these approaches and current state of the art bag of visual keywords descriptors, and shown that our novel approaches significantly outperform the previous methods. Still, the best single descriptor on the 101 classes is the SIFT descriptor that computes and combines feature vectors at various gridpoints. This may be caused by its ability to keep structural relationships between parts of the image. The visual keywords descriptors all compute an orderless collection of features that leads to losing information about structures. Although this problem is slightly overcome by using the spatial pyramid, when using too many levels of the pyramid these approaches lead to a very large number of features.

Another problem of the combination of the descriptors with an SVM is that particular very relevant keywords (such as a wheel for recognizing a car) receive a small value in the resulting complete image representation when these relevant parts only occupy a small part of the image.

In future work we want to research novel methods that can deal with dense grids and keep the structural relationships between parts of the image in the resulting image representation. This is not a simple problem, since there can be many relationships between image parts. Therefore the system should be able to represent relevant parts that co-occur with other relevant parts in discriminative spatial structures.

## ACKNOWLEDGMENT

The first author wishes to thank UKM-AP-ICT-17-2009, UKM-TT-03-FRGS0129-2010 and UKM-GGPM-ICT-119-2010 for funding this project.

## REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, vol. 60, 2004, pp. 91–110.
[2] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Conference On Image And Video Retrieval (CIVR 2007)*, 2007, pp. 401–408.
[3] A. Abdullah, R. C. Veltkamp, and M. A. Wiering., "An ensemble of deep support vector machines for image categorization," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPar09)*, 2009, pp. 301–306.
[4] A. Abdullah, R. C. Veltkamp, and M. A. Wiering, "Spatial pyramids and two-layer stacking SVM classifiers for image categorization: A comparative study," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN-09)*, 2009, pp. 5–12.
[5] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003, p. 1470.
[6] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *10th European Conference on Machine Learning (ECML)*, 1998, pp. 4–15.
[7] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek, "Visual word ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. in press, 2010.
[8] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007, pp. 494–501.
[9] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[10] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 411–426, 2007.
[11] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006, pp. 2169–2178.
[12] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
[13] D. M. Tax, R. P. Duin, and M. V. Breukelen, "Comparison between product and mean classifier combination rules," *In Proceedings of the Workshop on Statistical Pattern Recognition*, pp. 165–170, 1997.
[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 1*, June 2005, pp. 886–893.
[15] A. Abdullah, R. C. Veltkamp, and M. A. Wiering., "Fixed partitioning and salient points with MPEG-7 cluster correlograms for image categorization." *Pattern Recognition*, vol. 43(3), pp. 650–662, March 2010.
[16] G. Csurka, C. Dance, L. Fan, and C. Bray., "Visual categorization with bag of keypoints," in *The 8th European Conference on Computer Vision*, 2004, pp. III:513–516.
[17] F. Perronnin, C. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," *European Conference on Computer Vision (ECCV 2006)*, pp. 464–475, 2006.
[18] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
[19] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," in *Department of Computer Science, National Taiwan University, Taipei Taiwan*, 2008.
[20] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Springer-Verlag, 2000, pp. 1–15.