Spatial Pyramids and Two-layer Stacking SVM Classifiers for Image Categorization: A Comparative Study

Azizi Abdullah, Remco C. Veltkamp and Marco A. Wiering

***Abstract*— Recent research in image recognition has shown that combining multiple descriptors is a very useful way to improve classification performance. Furthermore, the use of spatial pyramids that compute descriptors at multiple spatial resolution levels generally increases the discriminative power of the descriptors. In this paper we focus on combination methods that combine multiple descriptors at multiple spatial resolution levels. A possible problem of the naive solution to create one large input vector for a machine learning classifier such as a support vector machine, is that the input vector becomes of very large dimensionality, which can increase problems of overfitting and hinder generalization performance. Therefore we propose the use of stacking support vector machines where at the first layer each support vector machine receives the input constructed by each single descriptor and is trained to compute the right output class. A second layer support vector machine is then used to combine the class probabilities of all trained first layer support vector models to learn the right output class given these reduced input vectors. We have performed experiments on 20 classes from the Caltech object database with 10 different single descriptors at 3 different resolutions. The results show that our 2-layer stacking approach outperforms the naive approach that combines all descriptors directly in a very large single input vector.**

I. Introduction

MACHINE VISION is a subfield of artificial intelligence that focuses on extracting useful information from images. During the last decade a large number of novel algorithms have been described for image recognition and this has led to good recognition performance on many different benchmarks. These algorithms use descriptors describing an image and then a machine learning algorithm to classify the images. Although traditional approaches focus on color- and texture-based descriptors, their lack of discriminative power led researchers to use more advanced shape-based and/or appearance-based descriptors. Shape-based descriptors often use a histogram of orientation gradients (HoG) [16], [7] and recent research combines this with a spatial pyramid [15], [3] approach where the HoGs are computed at multiple spatial resolution levels and positions inside a viewing window. These shape-based descriptors are quite invariant to image distortions and have a good discriminative power. Appearance-based descriptors [21], [6] use a descriptor such as the HoG or another descriptor and create a bag of visual keywords from multiple patches in an image. This is most often done using clustering techniques to create a particular visual code-book. By looking at multiple positions in the image, a histogram is constructed that reflects the distribution of visual keywords in an image. Combining many of such descriptors and giving them as input to a learning classifier such as a support vector machine (SVM) [24] has been shown to lead to very good results.

In [11], [15], the computed descriptions at different levels of the spatial pyramid are combined into a single vector. Besides that, each level is manually weighted using a certain scheme because it provides different kinds of information. As a result, a large feature input is constructed for indexing an image. However, when this method is used to combine many descriptors in a single large input vector, this may lead to overfitting the data and worse generalization performance. Therefore, a method by Zhang et al. [8] was proposed to provide a more efficient way to combine multiple descriptors. Although their method is not published in a separate paper, it worked very well in the PASCAL 2006 challenge. It basically uses a stacking method [25] where at the first layer support vector machines are trained using different descriptors and at a different level of the spatial pyramid to learn to compute the right classes. The output probabilities are computed by the individual support vector machines and then these probabilities are all combined to serve as input vector for another support vector machine that learns to make the final classification.

Contributions. In this paper we use 20 classes from the Caltech dataset to compare ten different single descriptors computed at different levels of the spatial pyramid, and the combination of all levels. Furthermore, we show the results of three methods that combine all descriptors and spatial levels: (1) The naive approach that combines all features computed by all descriptors in a single input vector for a support vector machine. (2) The 2-layer stacking SVM of Zhang [8] that uses as first layer models the support vector machines that receive as input a single descriptor computed at each different level. (3) Our novel 2-layer stacking SVM that uses first layer models that receive the inputs of a single descriptor computed at all different spatial levels.

The originality of our work is: (1) We compare the effectiveness of two different 2-layer stacking SVMs to the naive approach. (2) We compare many different single edge descriptors based on intensity and color information. (3) We compare the usefulness of different spatial levels and the combination of all spatial levels for different descriptors.

The rest of the paper is organized as follows. In section II we briefly describe related work in using image partitioning schemes. In section III we describe the different methods

Azizi Abdullah and Remco C. Veltkamp are with the Department of Information and Computer Sciences, Utrecht University, The Netherlands (email: {azizi, Remco.Veltkamp}@cs.uu.nl).

Marco A. Wiering is with the Department of Artificial Intelligence, University of Groningen, The Netherlands (email: mwiering@ai.rug.nl).

for combining multiple descriptors computed at different spatial levels. In section IV, we describe the descriptors that we used. These descriptors compute feature vectors that are used to construct support vector machine classifiers. In section V, we describe our new two-layer stacking spatial dense pyramids for image recognition. In section VI, the categorization effectiveness of three different combination methods are evaluated and compared to single descriptors on 20 classes of the Caltech-101 dataset. Section VII concludes this paper.

II. Related Work in Image Partitioning Schemes

One of the major difficulties in managing visual information is to encode the image in a discriminative feature space. Usually, an image is represented by a feature vector and a machine learning method is used to learn to discriminate image classes based on these feature vectors. The feature vector can be extracted either globally from the whole image or locally as in region-based image schemes. Once the image representation is selected, the next steps are to select a visual descriptor and a machine learning algorithm for learning to compute the right output class given the feature vector. In this section, we briefly describe some image partitioning methods. These partitioning methods compute particular histograms (e.g. orientation histograms) to compute a feature vector in a part of the image.

A. Global approach

In literature, global histogramming is the most commonly used scheme to capture the visual information in an image. The scheme provides compact representations of images, where each image corresponds to a point in some feature space. However, the scheme suffers from occlusion, clutter or spatial variation of objects in the image. For example, in [23] this scheme is used with an edge direction and various color histograms and in [10] this scheme is used with the simple color histogram. Retrieval results using this global approach were not very promising, which led to many variations of partitioning schemes. One of the widely used variations of global histogramming is local histogramming as used in region-based approaches.

B. Local region-based approach

Region-based approaches are quite popular to represent the local image content. The region-based approach tries to apply an image segmentation technique to extract regions from images. Then, the similarity between images is measured by calculating the correspondences between their regions. Typical examples of region-based retrieval systems include Blobworld [5] and VisualSEEK [22]. However, it is quite difficult to achieve accurate segmentation in an image with less distinctive objects [22].

Besides image segmentation, another way to overcome the limitation of the global feature approach is to use a fixed partitioning scheme. This approach has become more popular and has been shown to be a powerful image representation technique [1], [19]. In fixed partitioning, an image is equally divided into several partitions and for each partition a different local histogram is computed. One of the main advantages of using this approach is that it gives additional information to the histogram to capture the spatial distribution of the image content. Besides the fixed partitioning scheme, a multi-level histogramming scheme [17] based on the quad-tree structure is also used to incorporate spatial components in an image. However, the dimensionality of the feature space can become very large, because many different local histograms need to be computed and stored. Thus, to reduce the number of its inputs, the random patches scheme is proposed, where several patches are randomly generated and combined to obtain the image signature. A drawback of this method is that it needs a clustering method to compute an invariant histogram, and sometimes the use of clustering leads to less discriminative descriptors. Instead of using multi-level histogramming, the sliding windows scheme is also possible to represent the local image content, but this approach is computationally inefficient, i.e., one has to visit every portion of the image, resulting in thousands of evaluations that have to be performed.

An alternative approach in representing images with local regions has been developed that is called the saliency-based approach, which is said to be capable in handling images with complex structures. These methods are claimed to be robust and invariant to scale, rotation, viewpoints and illumination. The most popular and widely used salient points method is SIFT (Scale-Invariant Feature Transform) [16] and SURF (Speeded Up Robust Features) [2] is another (computationally more efficient) method. One of the main problems of the salient points scheme is that its local patches are orderless. To ease recognition of the image content, the local patches should be in a certain order in the spatial layout. Fortunately, it is easy to capture the spatial relationship between local patches to enrich the semantic description of the visual information. There exists a simple, but quite discriminative approach to represent the spatial order of the local patches. This method will be explained next.

C. Spatial pyramid approach

The multi-resolution approach in [13] uses a pyramid representation to capture the spatial correspondence between histograms. A multi-resolution image was constructed using four levels of the Burt-Adelson pyramid [4]. In this method, each level is obtained by filtering with a Gaussian kernel and subsampling. After that, the authors computed the histogram of each of the four levels. The distance between two multi-resolution histograms is the sum of the four individual L_1 distances between pairs of histograms corresponding to the same pyramid levels. In contrast with this approach, the spatial pyramid approach [15], [3] uses the fixed partitioning scheme to combine several levels of histograms as illustrated in Fig. 1. Combining multiple levels using this approach has been shown to improve recognition performance compared to using a single level [12], [15], [3], [13]. In this paper this spatial pyramid scheme is used.

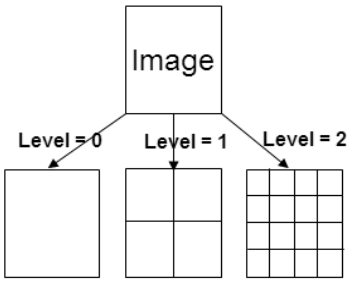

Fig. 1. *A spatial pyramid representation with correspondence to level 0, 1 and 2 respectively.*

III. Combining multiple features

Many content based information retrieval or machine vision systems combine multiple image features to improve their performance. Multiple image features normally produce different evidences of visual information for feature matching between reference and observed images. The main idea of combining multiple evidences is that repeated evidences of the same object would increase the probability of relevant features in the object. As a result, by using this approach, its retrieval results are improved as reported in [1], [3], [18]

We use the spatial pyramid representation approach and combine multiple features in our experiments. We used this for several reasons: (1) The features can be computed easily and efficiently. (2) The system preserves the spatial information of images by simply combining local histograms at multiple levels. (3) The histogram itself has many merits, such as invariance to image rotations and robustness to image translations around the viewing axis, and it varies slowly with the angle of view [20], [13]. (4) Each level in the spatial pyramid presents different information for recognizing the image.

We believe this approach enriches the semantic description of the visual information. With these advantages, the spatial pyramid approach provides more discriminative power for recognizing images than other approaches. However, we still have to combine all local histograms in a classifier. In this paper, we report two different proposed methods which are relevant to our study.

A. Spatial Pyramid Classifier

We construct a representation using three levels of the spatial pyramid [15], see Fig. 1. In general, the method uses one global and multiple local feature histograms to describe images. The global feature histogram is suitable to describe simple images and has the ability to represent an entire object with a single small vector. In contrast, the local histograms are computed in multiple regions and are more robust to complex disturbances such as occlusion and clutter. After the histograms are computed at multiple spatial resolution levels, they are combined to form a set of histograms. In our implementation, three different levels of resolutions were chosen, i.e, levels. 0, 1, and 2, to represent the finest, middle, and coarsest resolution, respectively.

The spatial pyramid approach uses the fixed partitioning scheme to construct multiple spatial resolution levels in the image. Each histogram in each partition is used to capture spatial information in the image. In this case, the input image is equally divided into several partitions or regions. The number of partitions depends on the number of spatial cells for each level. In [15], for each level i, the number of cells is determined by 4^i. After that, any descriptor can be applied to each partition. Finally, histograms (vectors) of the image at all levels are concatenated to form a single vector that incorporates the global and local histograms to describe the image. After that, a support vector machine (SVM) [24] is used to classify images. The idea is illustrated in Fig. 2.

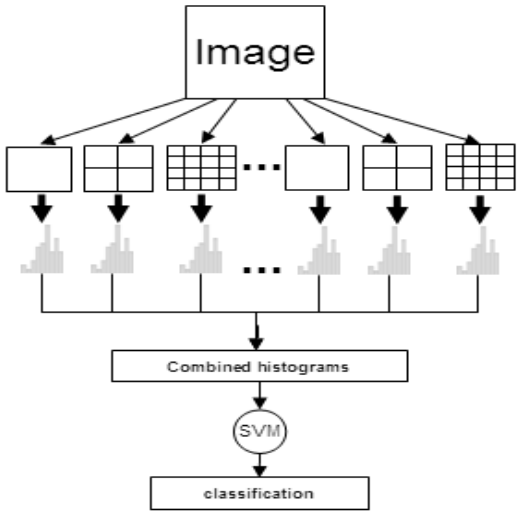

Fig. 2. *Combining a spatial pyramid using multiple descriptors with correspondence to level 0, 1 and 2 respectively. The histograms are combined at all levels and a support vector machine is used for classification.*

B. Two-Layer Stacking Spatial Pyramid Classifier

A problem of the above technique is that the spatial pyramid approach will increase the size of the concatenated description of the image. Furthermore, when many descriptors are used, the feature vectors become very large, and the computational time becomes large for training the SVM and for querying images. Finally, this naive combination method can also cause overfitting and decrease generalization performance.

In [15], it is shown that the performance at level 0 is worse than using level 2. Therefore, the authors used a fixed weighting scheme for features computed at different levels. This fixed weighting scheme might be not optimal for classification performance. We argue that the weighting scheme should be dynamic or more specifically adapted to yield optimal classification performance.

For these reasons, we explore a two-layer stacking method that reduces the size of input vectors and at the same time replaces the fixed weighting scheme. The stacking algorithm or more specifically a two-layer spatial stacking algorithm was proposed by Zhang et al. and described as an algorithm

that competed in the PASCAL-2006 visual object challenge [8]. This method can reduce the size of the large feature vectors and improve the generalization performance of the spatial pyramid classifier. The two-layer spatial stacking method combines outputs from different classifiers of the spatial pyramid approach. It uses the fact that the probability estimates or outputs from each classifier can be combined and used for recognizing images with many different descriptors. The system first trains a set of SVM classifiers on the histograms of each level with a single different descriptor in the pyramid. In this case, each classifier estimates the posterior class probability values or class predictions of a given image. The posterior probabilities contain important information about the predicted classes and can be used instead of the feature vectors of the descriptor to train the final classifier. After that, the outputs of these SVM classifiers are concatenated into a feature vector for each image. Then, this feature vector is used to learn another SVM classifier. In our implementation, an SVM classifier with the RBF kernel using the one-vs-all approach is used to provide probability outputs on the decision values. Fig. 3 shows the 2-layer stacking spatial pyramid approach.

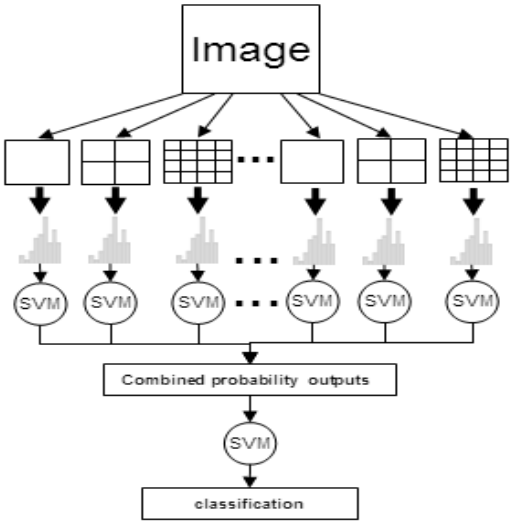

Fig. 3. *The 2-layer stacking spatial pyramid classifier.*

IV. Feature Extraction and Representation

It is often difficult to determine which image features are most useful to describe the information in an image. Good image features are crucial because they can give a compact representation and help to discover meaningful patterns in the image. Until now, there is no single solution to produce an optimal query result for all images. Recently, most studies are focusing on multiple image features for satisfactory recognition results. Using multiple image features may help to recognize different structures of images efficiently and enrich the semantic description of the visual information. Although many general feature detectors can be used, selected detectors should simulate a certain ability of the human vision system to get the most discriminative information. One of the most important features of our visual system is the construction of edge features, and using such edge orientation information it is possible to describe shapes. For this reason, edge-based descriptors such as SIFT [16] and histograms of oriented gradients [7] have become popular and are nowadays widely used in image recognition systems. Therefore, like many other researchers, we have chosen to concentrate on various edge descriptors to represent the image content.

These descriptors are applicable to real-world images and provide significant relationships between lines or contours and have enough power for shape discrimination. Moreover, our own experiments with other features than edges (not described in this paper) were performing worse. Therefore, we used three main different descriptors that are tested individually and combined in our system. Both color and intensity information are used in these descriptors.

A. The Detection of Color and Intensity Changes

The importance of color and intensity changes and edges in visual processing had led to extensive research and use in computer vision systems. Like other researches, both color and intensity features are used in the selected descriptors to describe images in our image recognition system. We believe that these features convey different information about edges in the image. Furthermore, the different descriptors can provide richer and more reliable descriptions of physical edges which can help to recognize the images.

The process of extracting information from edges can be divided into two main tasks. The first task is to detect the color and intensity changes in the image, and the second task is to describe the properties of edges by using a certain descriptor. Before the color or intensity changes are detected, pixels in the RGB color space are converted into a more robust color space. In our case, HSV and YIQ color models are used to describe color and intensity information respectively. In HSV space, each pixel is converted into hue, saturation and value components. After that all components are used to describe edges in the image. In YIQ space, only the Y component is used since this variable or dimension represents the luma information. It is demonstrated that most color images can be very well displayed using only 256 or 512 colors. Thus, all components are quantized in the interval 0 to 255 and this range also takes up less space. The overall feature extraction process for computing edges is shown in Fig. 4.

Once the image pixels are converted into H, S, V and Y components, the next step is to smooth or directly convolve each component with a convolution kernel. Finally, orientations and magnitudes at local regions are detected and used to describe edges. In our experiments, three different descriptors are used to describe edge features. The details of these descriptors are discussed below.

B. MPEG-7's Edge Histogram

Texture is important to check homogeneity and non-homogeneity between images. We used the MPEG-7 edge

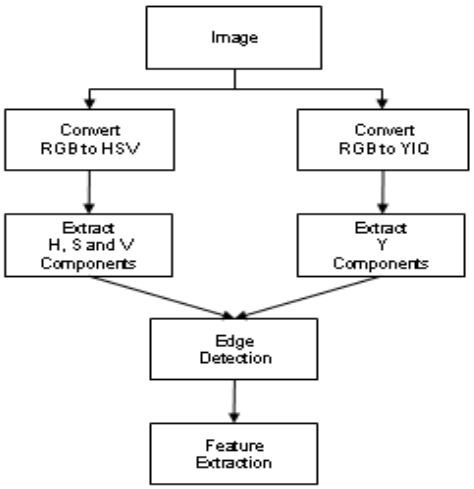

Fig. 4. *The overall feature extraction process for computing edges based on color and intensity perception.*

histogram [18] to compute texture information. The edge histogram describes a non-homogeneous texture and captures a local spatial distribution of edges. Given an input image or a region, the image or region is divided into 4x4 overlapping blocks.

The four mean values of the relevant color channel from the sub-blocks are convolved (left multiplied) with the following matrix with filter coefficients that represent different edge detectors:

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & 0 & 0 & \sqrt{-2} \\ 0 & \sqrt{2} & \sqrt{-2} & 0 \\ 2 & -2 & -2 & 2 \end{bmatrix}$$

The maximum of the most dominant edges is determined by comparing it with other edges' strength. Then the maximum of these results is compared with a threshold. The edge strength is composed of six different edge types, i.e. horizontal, vertical, 45°, 135°, non-directional, and no-edge. Finally, the descriptor with 80-bin and 240-bin histograms for intensity and color, respectively, are constructed for the input image by excluding the no-edge information. We named them as EH_G and EH_C to represent the edge histogram with intensity and color, respectively.

C. Histograms of Threshold-oriented Gradients (HTOG)

Shape is important to discriminate between objects. Local shape histograms are represented by edge orientations within an image subregion quantized into *N* bins. We model the shape by first applying a Gaussian smoothing function on color and intensity signals, and then we compute orientations by detecting the signal changes that are visible and significant in a certain angular range.

The histogram of oriented gradients descriptor [7] describes an image by a set of local histograms. These histograms count occurrences of thresholded gradients in a local part of the image. Before the HTOG is computed, the image colors and intensities are smoothed by the Gaussian smoothing kernel. The smoothing kernel is used here to reduce the effect of noise on the detection of color or intensity changes. Besides that, it is also used to set the resolution or scale at which color and intensity changes are detected. In our experiments, a 3x3 Gaussian kernel with $\sigma = 1.0$ is used to convolve all images. After that, the image is divided into 4 x 4 sub regions to capture the spatial relationship between edge attributes. Then the gradients dx and dy are computed on each point in each region by using the following filters in *x* and *y* directions, respectively.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

To compute the magnitude and orientation of the gradient the following formulas are used:

$$m(x,y) = \sqrt{dy^2 + dx^2}$$
$$\Theta(x,y) = arctan(dy/dx)$$

where *m* is the magnitude, Θ is the orientation of the gradient, and *dy* and *dx* are gradients in vertical and horizontal directions, respectively.

In order to compute the histogram of occurrences of different orientations, a certain threshold value is used to select the strongest edges. In case *m(x,y)* is below the threshold (in our experiments set to 10), the edge is considered as a weak response or noise rather than a strong edge and not counted. All Θ's which have a magnitude above the threshold are selected and then quantized into N bins. In our experiments, N = 8 gave the best results. Finally, the descriptor with 72 or 128 bins is constructed for the whole region (consisting of 3x3 or 4x4 blocks). Each bin in the histogram represents the number of occurrences of edges that have a certain orientation. We chose several angular ranges to recognize different structures of images and to enrich the semantic description of the edge information. We found two angular ranges i.e., 180° and 360° to be optimal in our dataset. An angular range of 180° maps angles between 180° and 360° to the range between 0 and 180 degrees. We named the four resulting descriptors HG_180_G, HG_180_C, HG_360_G and HG_360_C to represent the HTOG with intensity and color, respectively.

D. SIFT (Scale Invariant Feature Transform)

We also applied the SIFT descriptor proposed by Lowe [16] which constructs the histograms of gradient orientations computed around the points as the descriptor. The original SIFT version uses an interest points detector to detect salient locations which have certain repeatable properties. In contrast with this approach, we believe that using fixed partitioning blocks gives a simpler method with the same or better performance on our dataset. Furthermore, using this approach the spatial relationships between the SIFT features can be

represented more efficiently, i.e. we do not need clustering. Therefore, fixed regions without orientation alignment are constructed over the image and instead of 'salient points' we compute the center of each region.

To compute the descriptor, an input image (whole image) is smoothed with the same smoothing function and differentiated using the same dx and dy filters as in the HTOG descriptor. Then the number of regions to construct the descriptor is generated corresponding to each level in the pyramid. After that, the center point of the region is determined by 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 is determined by taking the *min*($\frac{width}{2}, \frac{height}{2}$), where width and height are the sizes of the region. After that, the descriptor breaks apart a window around the center point into 4x4 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. Our use of SIFT differs from the HTOG in the following ways: it uses a circular region instead of a rectangular block and it does not use a threshold on the magnitude. In this way we compute complementary features with SIFT and HTOG.

We also used SIFT descriptors with 180° and 360° angular ranges to enrich its visual information. We named them S_180$_G$, S_180$_C$, S_360$_G$, and S_360$_C$ to represent the SIFT descriptors with intensity and color information, respectively.

V. Two-Layer Stacking Spatial Dense Pyramid Classifier

The two-layer stacking algorithm, which we have discussed in Section III is based on each spatial level to generate the probability outputs. Here we provide an alternative method that combines features at all levels from the same descriptor. We modified the approach of Zhang et al. [8] for the following reasons: (1) Our method can combine the best performing classifiers by combining global and local features at all levels. (2) Using the approach of Zhang et al., a single classifier might be less efficient to discriminate different image classes, because it uses a smaller feature size. (3) Combining features at all levels from the same descriptor can be more discriminative, since it uses the whole spatial pyramid that can cope with varying degrees of spatial correspondences in the image. Fig. 5 shows our new architecture.

Similar to the 2-layer stacking spatial pyramid method, our method uses RBF kernels and the one-vs-all approach to generate probability outputs from each descriptor. Suppose that we have N image classes, then a support vector machine with a single descriptor gives N decision values and thus a N-dimensional space is created. When using M descriptors, there are in total MxN probability values for the second-layer SVM classifier. These values may give better distinctions between images classes since the separate prediction values of a first layer support vector classifier will give more accurate class probability values or outputs.

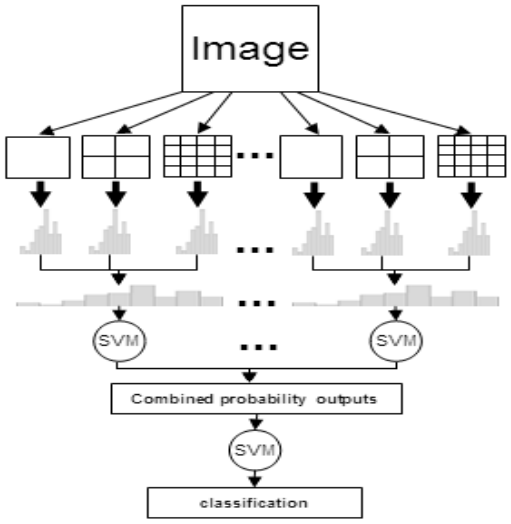

Fig. 5. *The 2-layer stacking spatial dense pyramid classifier.*

VI. Experiments and Results

For our comparison between the different descriptors and combination algorithms, a variety of image classes were chosen. The images should be common and familiar to machine vision researchers, and therefore we used a well known dataset, i.e. Caltech-101. The dataset contains various image sizes and were categorized into 101 different classes. In our experiment, only the first 20 classes were chosen for evaluation due to computational restrictions. Each image in the dataset consists of different sizes and contains different viewpoints, which makes the recognition process more challenging.

A. SVM Classifier

We employ an SVM [24] to learn to classify the images. The one-vs-all approach is used to train and classify images in the Caltech-101 dataset. For the SVMs, we use both Radial-Basis-Function (RBF) and linear kernels in the experiments and after that we compare them to get the best classification performance.

Initially, all attributes in the training and testing were normalized to the interval [-1,+1] by using this equation:

$$x' = \frac{2(x-min)}{(max-min)} - 1.$$

The normalization is used to avoid numerical difficulties during the calculation and to make sure the largest values do not dominate the smaller ones. Besides that, by doing this the matching of spatial information in the spatial pyramid is based on this range rather than simply on differences in intensity histograms. We did not use the fixed weighting scheme for the spatial pyramid classifier. Preliminary experiments indicated that it did not improve the results.

We also need to find the SVM parameters C and γ that perform best for the descriptors. To optimize the classification performance, the kernel parameters were determined by using the libsvm grid-search algorithm [14]. The C and

γ values can be tried out exponentially to get the best accuracy performance. Therefore, we tried the following values $\{2^{-5},2^{-3},...,2^{15}\}$ and $\{2^{-15},2^{-13},...,2^{3}\}$ for C and γ respectively. The values which gave the best accuracy performance are picked and used to train on the training set.

We found in our experiments that it is quite difficult to get the best C and γ parameters to train the dataset. The main reason is that the dataset is unbalanced. Thus, we have to find the best ratio between positive and negative samples to get the optimum C and γ values. In this case, we have tried two possibilities. The first experiment is to use an unbalanced dataset of 5% positive samples and 95% negative samples, and the second experiment is to use 50% positive samples and 50% negative samples of similar shape appearance. Besides the SVM parameters, the scaling factor to normalize the features is another issue. The scaling factor influences the classification performance [14]. We have tried two different scaling factors to determine the best min and max values for scaling the training and testing datasets. The first experiment is to use 600 feature vectors and the second experiment to use 300 feature vectors. After that, we scale all feature vectors using these values. Similar to the above mentioned problems, we also found that the spatial arrangement of HTOG and the radius of SIFT descriptors influence the image indexing performance. For the HTOG we have tried two spatial arrangements which return 4x4 histograms and 3x3 histograms of 8 orientations. For the Sift descriptor we have used two types of radius for each overlapping block i.e *min*($\frac{width}{2}, \frac{height}{2}$) and $\sqrt{(\frac{width}{2})^2 + (\frac{height}{2})^2}$. We report only the results obtained with the best parameters below. The indexing process takes some time and it depends on the number of images, number of features used, and system configuration. The time taken for optimization and training was much longer for the spatial pyramid classifier than for the 2-layer stacking methods.

B. Caltech-101 dataset

The Caltech-101 is one of the most popular and widely used datasets to demonstrate the performance of object recognition systems [9]. It consists of 101 categories depicting real world object images such as camera, airplanes, bonsai, anchor, etc. In general, Caltech-101 contains a collection of more than 1000 photos and about 31 to 800 images per category. In our experiments, we used the first 20 categories (in alphabetical category order) and a total of 20x30= 600 images for evaluation. These images are all in JPEG format with medium resolution about 300 x 300 pixels and both in color and gray level representation. Fig. 6 shows the ground truth for the 20 different classes we used of the Caltech-101 dataset.

We used the region of interest (ROI) taken from [3] 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. To compute the performances of the different methods, we choose 5 times different training and test images randomly from a set of candidate images in the 20 classes of the Caltech-101 dataset. Finally, we report the performance using mean and standard deviation to verify significances of the obtained classification results.

TABLE I

THE AVERAGE CLASSIFICATION ACCURACY (MEAN AND SD) OF THE DIFFERENT DESCRIPTORS.

	level 0	level 1	level 2	pyramid
EH_G	59.02±2.06	59.80±0.99	-	62.20±1.43
EH_C	61.73±1.70	62.07±1.82	-	64.07±2.14
S_180_G	63.07±1.19	68.60±3.10	74.53±1.52	72.67±1.43
S_180_C	60.73±66.47	66.47±2.17	68.93±1.36	71.07±1.04
S_360_G	61.07±2.03	66.07±1.28	71.53±1.74	65.40±2.61
S_360_C	60.93±1.80	62.80±0.50	64.00±0.97	66.40±2.29
HG_180_G	57.33±1.70	65.07±1.21	67.47±2.45	70.13±2.53
HG_180_C	56.40±2.76	67.27±1.52	64.80±2.07	69.13±1.98
HG_360_G	53.93±2.35	60.47±2.37	60.80±1.39	63.54±2.29
HG_360_C	50.53±62.13	62.33±1.43	62.33±1.85	65.53±2.84

Fig. 6. *Image examples with ground truth for different groups namely accordion, airplane, anchor, ant, background, barrel, bass, beaver, binocular, bonsai, brain, brontosaurus, Buddha, butterfly, camera, cannon, car side, ceiling fan, cell phone and chair respectively.*

C. Classification Results and Discussion

Table I shows the average classification accuracy and the standard deviation of the different descriptors to classify images using the RBF kernel. The results show that the average classification accuracy for each descriptor is best for level 1 from the 3 levels. Increasing the number of levels in EH_C, HG_180_C, and HG_360_C from 1 to 2 made classification performance much worse, so we do not report their results or use them in the pyramid. In this case, levels 0 and 1 have sufficiently rich information to describe objects and perform better than the intensity based descriptors at these levels. Finally, the table shows that combining all used levels in the pyramid often improves the performance of the best single level.

TABLE II

THE AVERAGE CLASSIFICATION ACCURACY (MEAN AND SD) OF THE DIFFERENT COMBINATION CLASSIFIERS. M1=SPATIAL PYRAMID, M2=TWO-LAYER STACKING SPATIAL PYRAMID, AND M3=TWO-LAYER STACKING SPATIAL DENSE PYRAMID

	M1	M2	M3
RBF	77.35±0.88	79.00±1.55	83.40±3.03
Linear	75.33±2.27	76.87±1.57	83.60±3.13

To compare the three combination methods i.e. spatial pyramid, two-layer stacking spatial pyramid, and two-layer stacking spatial dense pyramid, the same average classification accuracy is computed using the same training and test sets. Table II shows the overall image classification performance of these methods using the SVM classifier. In this experiment, our novel two-layer stacking dense spatial pyramid algorithms gave the best performance using both RBF and linear kernels and outperforms all other methods. This is probably caused by the fewer values that need to be combined, preventing overfitting, and the more accurate probability values resulting from directly using the pyramids. Zhang's approach did not significantly outperform the naive approach.

VII. CONCLUSION

In this paper, we have introduced a novel stacking SVM approach that combines many different features and different spatial resolutions. We reported a significant comparison between this approach and existing spatial pyramid and two-layer stacking SVMs, and our novel method significantly outperforms the previous methods. Different texture and shape descriptors, notably MPEG-7 edge histograms, SIFT features, and histograms of oriented gradients are used to construct the SVM models. SIFT turned out to give the best results, and the MPEG-7 edge histogram gave the worst results. It is a bit remarkable that Zhang's stacking approach does not perform significantly better than the naive approach. Probably this is because particular features computed at specific spatial resolution levels do not give very good results, so that they disturb the final performance. This problem is circumvented by using the probability outputs from the spatial pyramids like in our approach, since these values are much more reliable.

There are several ways to extend this research. We are currently working on creating stacking SVM classifiers with more than 2 layers. For this, we will research how to build the hierarchical SVM stacking layers to optimize the feature integration process. We also want to research other ensemble methods like majority voting and deep SVM architectures which we are currently developing. Finally, we want to test the methods using more features and classes from the Caltech-101 dataset.

REFERENCES

[1] A. Abdullah and Marco A. Wiering. Cirec: Cluster correlogram image retrieval and categorization using mpeg-7 descriptors. In *IEEE Symposium on Computational Intelligence in Image and Signal Processing, CIISP 2007*, pages 431 – 437, 2007.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006.

[3] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Conference On Image And Video Retrieval(CIVR 2007)*, pages 401–408, 2007.

[4] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. In *IEEE Transactions on Communications*, volume COM-31,4, pages 532–540. IEEE, 1983.

[5] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: a system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*, pages 509–516. Springer, 1999.

[6] G. Csurka, C. Dance, C. Bray, and L. Fan. Visual categorization with bags of keypoints. In *Proceedings Workshop on Statistical Learning in Computer Vision*, 2004.

[7] 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*, pages 886–893, June 2005.

[8] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The pascal visual object classes challenge 2006 (voc2006) results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[9] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, page 178, 2004.

[10] Monika M. Gorkani and Rosalind W. Picard. Texture orientation for sorting photos at a glance. In *TR-292, M.I.T., Media Laboratory, Perceptual Computing Section*, pages 459–464, 1994.

[11] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1458–1465, 2007.

[12] Kristen Grauman and Trevor Darrell. Efficient image matching with distributions of local invariant features. In *CVPR (2)*, pages 627–634, 2005.

[13] E. Hadjidemetriou, M. Grossberg, and S. K. Nayar. Spatial information in multiresolution histograms. In *Proceedings of IEEE 2001 Conference on Computer Vision and Pattern Recgnition*, 2001.

[14] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. In *Department of Computer Science, National Taiwan University, Taipei Taiwan*, 2008.

[15] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[16] David G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 60, pages 91–110, 2004.

[17] H. Lu, B. Ooi, and K. Tan. Efficient image retrieval by color contents. In *Lecture Notes in Computer Science*, pages 95–108. Springer Berlin/Heidelberg, 1994.

[18] M. Lux, J. Becker, and H. Krottmaier. Caliph and emir: Semantic annotation and retrieval in personal digital photo libraries. In *Proc. of 15th CAiSE 2003*, pages 85–89, 2003.

[19] I. K. Sethi, I. Coman, B. Day, F. Jiang, F., D. Li, J. Segovia-Juarez, G. Wei, and B. You. Color-wise: a system for image similarity retrieval using color. In *Proc. SPIE, Storage and Retrieval for Image and Video Databases VI, Vol. 3312*, pages 140–149, 1997.

[20] Linda G. Shapiro and George C. Stockman. Computer vision. Prentice Hall, ISBN 0130307963, 2003.

[21] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 2, 2003.

[22] John R. Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. In *Fourth ACM international conference on Multimedia*, pages 87–98, 1996.

[23] Aditya Vailaya, Mrio A. T. Figueiredo, Anil K. Jain, and Hong-Jiang Zhang. Image classification for content-based indexing. In *IEEE Transactions on Image Processing*, volume 10, pages 117–130, 2001.

[24] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[25] D. H. Wolpert. Stacked generalization. In *Neural Networks*, volume 5, pages 241–259, 1992.