
WWJMRD 2021; 7(1): 38-49
www.wwjmrd.com
*International Journal*
*Peer Reviewed Journal*
*Refereed Journal*
*Indexed Journal*
*Impact Factor MJIF: 4.25*
E-ISSN: 2454-6615

**Noha Elfiky**
Saint Mary's College of California, Moraga CA, USA

**Correspondence:**
**Noha Elfiky**
Saint Mary's College of California, Moraga CA, USA

# Image Classification using Spatial Layouts derived from 3D Scene Geometry

## Noha Elfiky

**Abstract**
The Bag-of-Words (BoW) approach has been successfully applied in the context of category-level image classification. To incorporate spatial image information in the BoW model, Spatial Pyramids (SPs) are used. However, spa-tial pyramids are rigid in nature and are based on pre-defined grid configurations. As a consequence, they often fail to coincide with the underlying spatial structure of images from different categories which may negatively affect the classification accuracy.
The aim of the paper is to use the 3D scene geometry to steer the layout of spatial pyramids for category-level image classification (object recognition). The proposed approach provides an image representation by inferring the constituent geometrical parts of a scene. As a result, the image representation retains the de-scriptive spatial information to yield a structural description of the image.
From large scale experiments on the Pascal VOC2007 and Caltech101, it can be derived that SPs which are obtained by selective search outperforms the stand-ard SPs. The use of 3D scene geometry, to select the proper SP configuration, provides an even higher improvement.

**Keywords:** Big Data Analytics, Machine Vision, Image Classification and Ob-ject Recognition Tasks, Bag of Words, Spatial Pyramids

## Introduction
For category-level image classification and object recognition, the Bag-of-Words (BoW) approach has been successfully applied [1], [2], [3], [4]. The BoW is based on the occurrences of image features. Hence, it treats the image as an order-less collection of local features completely ignoring the spatial image layout. Hence, it treats the image as an order-less collection of local features completely ignoring the spatial image lay-out.
Extending the *BoW* with spatial information has therefore received considerable at-tention. Recently, several approaches consider the success of the Spatial Pyramid *(SP)* approach proposed by Lazebnik et al. [5]. It is shown that the use of *SP* outperforms the 1×1 image representation on challenging image classification tasks [5], due to the inclusion of image-to-image geometric correspondences. However, in general, SPs are based on rigid image subdivisions (e.g., grids). These rigid spatial configurations are not well suited for freely shaped objects and scenes. In Figure 1(a), some examples are shown taken from different image categories together with their standard SP sub-divi-sion. Sub-regions divide objects into two separate parts increasing the probability of dissimilar image features within cells and similar image features across cells. Hence, a rigid division may a negative equivalence-class configuration of image features.
Our aim is to use 3D scene geometry to steer the layout of spatial pyramids for category-level image classification. Images within a category usually share similar scene geometries. We exploit correspondences between categories by a scene geometry matching scheme. For example, Figure 1(b) shows the geometrical (depth) layers of some example images. The *cow example* corresponds to scene geometry style consist-ing of 3 segments: (1) ground, (2) background and (3) sky. The "ground" part depicts different objects than the background and sky. Each segment contains similar features and features across segments are more dissimilar. Therefore, the *BoW* should be applied separately to each geometrical scene sub-region.

In this paper, we propose a method to obtain a holistic image representation by in-ferring the constituent geometrical parts of a scene. The method steers the image layout on the basis of 3D scene geometry (i.e., "Stages") computed from a single image. We propose three alternative approaches to obtain structural image representations from 3D scene geometries:

1. ***Generic SPs:*** by exploiting the 3*D* scene geometry of images. The 13 stages of [7] are used as 3D priors. After the image scene geometry is estimated, the most appro-priate stage per object category is selected as the spatial pyramid.
2. ***Adaptive spatial pyramids:*** by learning the spatial image subdivision per object cat-egory using class specific binary masks. Hence, the corresponding geometrical rep-resentation is used as spatial pyramids.
3. ***Selective SPs:*** by obtaining the spatial subdivision representation based on selective search guided by the *Agglomerative Information Bottleneck (AIB)* theory [8], [9]. The models are used to select the spatial geometry that best suits each category at hand.

The three methods to generate SPs will be compared to existing rigid SP for object recognition tasks. To this end, two benchmark data sets are used in the experiments: Pascal VOC 2007 [6] and Caltech-101 [10]. Furthermore, a large data set is provided *(denoted as "stage data set")* to learn each stage.

This paper is organized as follows. First, in section 2 and 3, we give the motivation of our approach and discuss related work on rigid spatial pyramids. In section 4, the method is proposed to generate dynamic spatial pyramids. In section 5, the experi-mental setup is discussed and the results are given. Finally, we give the conclusion in Sec. 6.

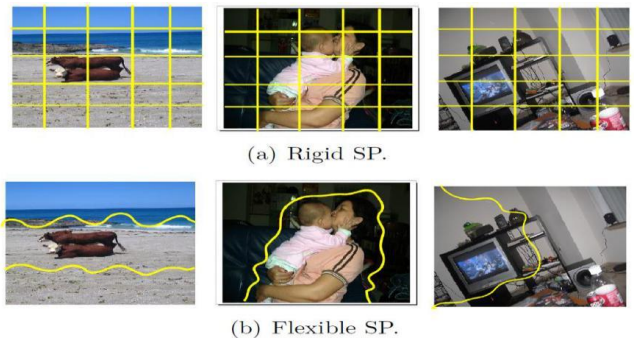


**Fig. 1:** Show example images from different categories. (a) shows the standard SP proposed by Lazebnik et al. [5]. (b) shows the proposed flexible spatial partitionings which best suit each category. Images are from the Pascal dataset [6].

## Motivation

The subdivision scheme should consider the trade-off between two important design properties *invariance* and *descriptiveness*. *Larger* subregions are preferred to gain in-variance to viewpoint changes (translation, orientation and scale) and object occlusion. Sub-regions should cover the range of possible positions of occurring objects. For ex-ample, the entire image is invariant to all possible object positions. *Smaller* sub-regions are required to obtain more descriptive regions and spatial layout. Sub-regions should depict similar object/background augmenting the descriptive ability of the SP. Finally, regions should not be constrained in shape allowing for a natural division of the image into its constituent parts.

In this paper, we propose a strategy to divide the image into its constituent *scene geometry* parts to obtain an *invariant* and *descriptive* image representation. The aim is to split the image into sub-regions corresponding to generic scene (depth) layers. These layers provide a middle ground between low-level features and high-level object cate-gories. A number of methods have been proposed to estimate the rough scene geometry from single images [11], [12], [13]. We use the scheme which derives scene information for a wider range of generic scene categories by using *stages* [7]. Stages are defined as a set of prototypes of often recurring scene configurations. They can be seen as discrete classes of scene geometries. Typical classes of discrete 3D *scene geometries* include single-side backgrounds (e.g. walls and buildings) or three sides (e.g. corridor and nar-row streets). A number of stage models are shown in Figure 2. These models are de-pendent on the inherent geometrical structure of images. In this paper, 13 different stages are used excluding *noDepth or tab+pers+bkg*, as these stages are specific char-acteristics of the data set used in [7].

As shown in Figure 2, the scene structures of the stage models are shown in different colors. The stage models are used to determine how the image is divided in subregions. For instance, images of stage *sky+backgnd+gnd* are divided into three layers: sky (in blue), background (in yellow) and ground (in brown). In Fig. 3(a), it is shown that the example images from Fig. 1 are instantiations of the *"sky+backgnd+gnd"*, *"person"* and *"gnd+diagonal"* stages, respectively. Each scene (depth) layer is equivalent to an image segment. Hence, SPs are constructed based on 3D scene geometries in which each geometry layer (e.g., ground, background and sky) is represented by a different sub-region.

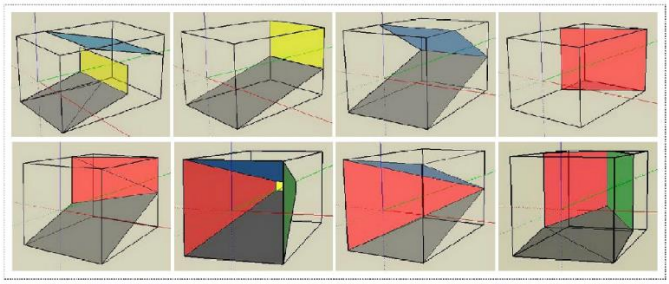

**Fig. 2:** Stage models and their corresponding instantiations. Top row, from left to right: "sky+backgnd+gnd", "backgnd+gnd", "sky + gnd", "gnd + diagalBackgndLR". Bottom row: "diagalBackgndLR", "box", "1side-wallLR", "corner". This figure is taken from [14].

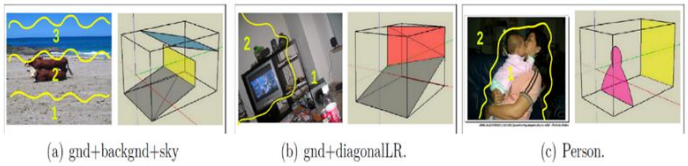


**Fig. 3:** Example depth images with their corresponding 3D geometries.

## Preliminaries

In this section, we briefly discuss the method for obtaining standard spatial pyramids [5] and for determining scene geometries (stages) for obtaining Generic pyramids [31].

***Rigid Spatial Pyramids***

The spatial pyramid scheme proposed by [5] is a simple and computationally efficient extension of an order-less bag-of-words image representation. This approach represents an image by using weighted multi-resolution histograms which are obtained by repeatedly sub-dividing an image into increasingly finer sub-regions. Histograms are computed over the resulting sub-regions. For each resolution level, the image is subdivided into the cells of a grid. At resolution $l$, the grid has $2^{2l}$ cells. The number of points in each grid cell is then recorded.

Marszalek et al.[15] evaluate both regular and irregular grids. Further, they consider a broader set of coarse subdivisions for each dimension, such as a 1×1 grid corresponding to the standard representation of the bagof-words, a 2×2 grid (i.e. four blocks), a horizontal 3×1 grid as well as a vertical 1×3 one. They show that dividing the image plane in three horizontal (i.e. 3 × 1 grid) regions, provides the highest recognition performance. Further, this approach reduces the dimensionality of the conventional 4 × 4 (i.e. sixteen blocks) structure; from *vocabularysize* × 21 to *vocabularysize* × 8.

The use of various image layouts shows the influence of the image configuration and spatial image representation. However, unconstrained spatial image representations have not been studied [16]. Moreover, pyramids commonly applied to BoW are not designed for the specific task of categorization, due to the assumption of having fixed rigid grid representation that suits all the dataset categories. The proposed approach resolves the use of rigid spatial pyramids. The aim is to generate more natural spatial pyramids based on the underlying image geometry.

***Image Segmentation to Obtain Depth Layers***

For each scene geometry, the different image segments correspond to a scene part at a certain depth (layer). Each segment represents geometrical entities like walls, ground, and sky. The image divisions provided by the scene geometry models will be used to learn the best geometry that suits each category of concern. Segmentation is based on the occurrence probability in the training set. Ground truth is obtained by manual annotation, thereby dividing the training set according to the scene geometry patterns, and fitting the parameters of each geometry model (horizon, vanishing points) such as to visually best fit the underlying data. For this purpose, the stages data set described in section 5.1 will be used for obtaining the segmentation masks used to represent each scene geometry.

More precisely, suppose that an image belongs to stage S, which is composed of N layers, correspondingly there will be N mask maps. The mask map for the $i^{th}$ partition $T_i$ is obtained by taking the average of the mask maps for each image:

$$T_i(x) = \frac{\sum_{j=1}^{n} M_{j,i}(x)}{n}, \qquad (1)$$

where $n$ is the total number of images in the training data set, and $M_{j,i}(x)$ is the mask map of the $j^{th}$ image for $i^{th}$ partition. Note that $M_{j,i}(x)$ is an indicator function: $M_{j,i}(x)$ =1, if $\boldsymbol{x}$ belongs to the $i^{th}$ partition and 0 otherwise.

**Segmentation:** mask maps are used to automatically divide the images. Assuming that the images of a stage can be partitioned into $N$ layers, there exist $N$ mask maps corresponding to the partitions in the training data set. Then, the binary mask map is defined as follows:

$$\mathrm{T'}_i(\mathbf{x}) = \begin{cases} 1, & \mathrm{T}_i(\mathbf{x}) = \max\limits_{\mathbf{j=1}}^{\mathbf{N}} \mathrm{T_j}(\mathbf{x}), \\ 0, & otherwise. \end{cases} \qquad (2)$$

As a consequence, the values in the mask map are either 0 or 1, as shown in Figure 4. In the next section, scene geometry (i.e. scene depth) maps will be used in order to achieve a proper selection of the spatial partitionings that suit each object category of concern.

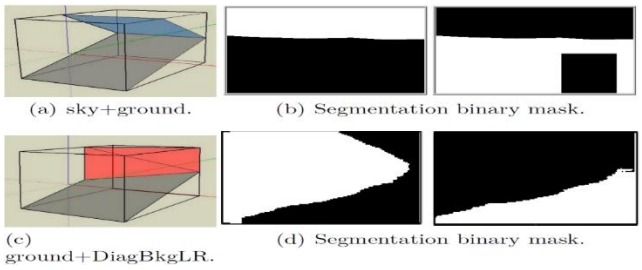


**Fig. 4:** An example of segmentation binary mask maps. Top row image belongs to *sky+ground* scene geometry. Bottom row image belongs to *ground+DiagBkgLR* scene geometry. The mask maps are of the same size as the original image.

**Spatial Layouts Derived From 3d Scene Geometry**

The SP scheme proposed by [5] represents an image by using weighted multi-resolution histograms which are obtained by repeatedly sub-dividing an image into increasingly finer sub-regions, where the spatial pyramid at level $l \in \{0, \dots, L\}$ has $R(l) = 2^{2l}$ sub-regions. For image $X$, all features are assigned to their best visual word $v$ selected from a vocabulary $V$. The frequency of $v$ inside sub-region $i$ of image $X$ is given by the histogram bin $H_X^i(v)$. The similarity or matching rate between images $X$ and $Y$ at level $l$, is given by the histogram intersection function [17]:

$$I^l(X,Y) = \sum_{i=1}^{R(l)} \sum_{v=1}^{|V|} \min(H_X^i(v), H_Y^i(v)). \qquad (3)$$

Matches found at finer resolutions are closer to each other in the image space and are therefore more heavily weighted. To accomplish this, each level is weighted to $1/2\, L - l$ which results in the final SP:

$$\kappa^L(X,Y) = \frac{1}{2^L} I^0(X,Y) + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}} I^l(X,Y). \qquad (4)$$

For the geometry-driven pyramid we use the same approach, only for geometric equivalent classes. Formally, for each scene geometry or stage $s$ of $n$ stage types, let $R(s)$ denote the number of sub-regions of $s$. Instead of using a fixed pyramid, we propose that a spatial stage pyramid is created by computing the similarity between images $X$ and $Y$ for stage $s$ by

$$I^s(X,Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v)). \qquad (5)$$

Where the different sub-regions $i_s$ for stage $s$ correspond to a scene part at a certain depth (layer). For each stage $s$ there are $R(s)$ sub-regions.

We propose two alternative approaches for selecting the appropriate spatial image representation for each category. This is achieved by learning a proper class-specific spatial model. These models encode the proper spatial partitioning

for each category; which is used further to obtain class-specific spatial models. To this end, we first exploit the use of the standard 3D geometry model as a prior for learning the best candidate template for each category. Then, we introduce an adaptive approach to generate spatial image representation that suits each category based on the ground-truth (GT) information of its training images. Lastly, we propose a learning approach for learning the most suitable category-model based on information theory.

***Generic Spatial Pyramids sets***

In this section, the 13 different scene geometries $\{S1,\ldots,S13\}$ proposed by [7] are used. It has been shown that these geometries cover most of the image partitionings encountered in real-world scenarios. Hence, 3D geometry structures are used to determine how the image should be divided; where each geometry depth corresponds to a pyramid region $i_s$. Therefore, the 13 prior stages are considered as generic models. Images of each category are classified into one of these stages in order to select the most appropriate spatial representation. More formally, the proposed method consists of the following steps: first, training images are spatially represented according to each of the 13 binary mask maps. We use the stage data set to obtain the binary mask maps. The training set is manually annotated and divided into scene geometries as in [18]. The parameters of each geometry (horizon, vanishing points) are computed to fit the underlying data. Image segmentation is based on the occurrence probability in the training set, see [18] for details. For each category, we then train 13 geometry models and learn on the validation set which geometry or even the combination of geometries that best suits each category. The whole process is demonstrated by the block diagram in Figure 5.

For a new image, it is represented using the 13 binary maps (off-line step). We evaluate the learned geometry model of each category w.r.t. its appropriate representation. Consequently, the test image will have a score towards each category and is assigned to the category with the highest score. We summarize the whole procedure in Algorithm 1.

**Algorithm 1** Generic Spatial Pyramids

1. **Require:** Binary Map *(BM)* of the 13 scene geometries $s \in \{S_1,\ldots,S_{13}\}$). Each *BM* has a number of subregions $R^s$.
2. **Training from 2-6:** Construct a histogram $H_X^{i_s}$, for each *BM* subregion $i_s$ of image $X$.
3. Train a Geometry Model *(GM)* with each *BM* representation.
4. Evaluate the performance score on the validation set.
5. Repeat steps 2 to 5 for each category.
6. Set the category geometry to *GM* with the highest score.
7. **Testing from 7-8:** For a test image, evaluate its performance for each category using its *GM* & *BM*.
8. Assign the test image to category label with the highest score.
9. The matching between images $X$ and $Y$ for a stage $s$ pyramid is given by:
$I^s(X,Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v))$.

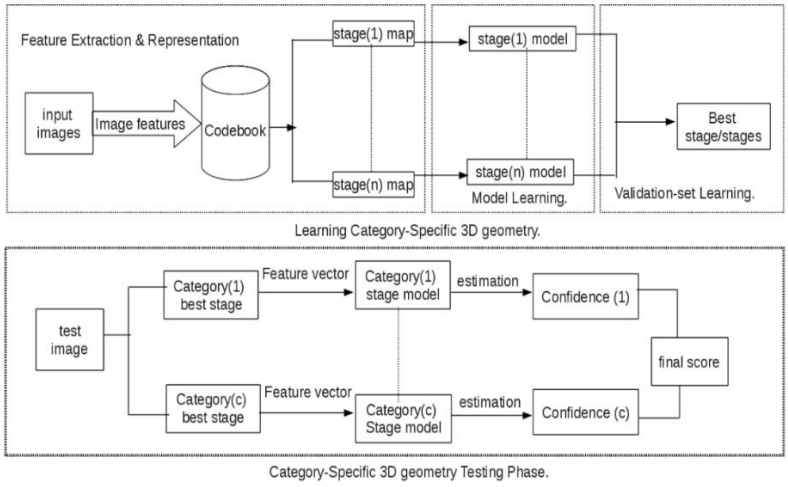


**Fig. 5:** Outline of the spatial representation using 3D scene geometry. Note that the codebook models and the stage models are obtained off-line. For each category, the proper stage model is obtained.

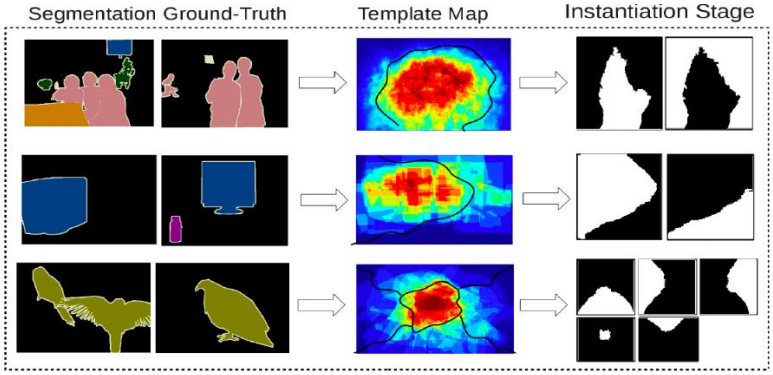


**Fig. 6:** Examples Adaptive Template maps obtained for different data set categories: person (top row), tv monitor (middle row) and bird (bottom row). Adaptive Template maps are learned on the dataset beforehand.

***Adaptive Spatial Pyramids***

In this section, an adaptive approach to generate spatial image representation that suits each category is proposed. In particular, a spatial template map for each object category c is learned based on the segmentation ground-tr uth (GT) of its training images. This template map (TM) is obtained by summing up the existence of each pixel location belonging to category c from its ground truth images:

$$\mathrm{TM}(\mathbf{x}) = \frac{\sum_{j=1}^{n} \mathrm{GT_j}(\mathbf{x})}{n}, \tag{6}$$

Where n is the total number of images in the training dataset of the $c^{th}$ category, and $GT_j(x)$ is the groundtruth map of the $j^{th}$ image. Note that $GT_j(x)$ is an indicator function: $GT_j(x) = 1$, if $x$ belongs to the $c^{th}$ category and 0 otherwise. We then apply a weighting scheme upon the generated TM in order to assign high weightings to higher frequency pixel positions, while suppressing other positions based on a Gaussian fitting function. Then, the binary mask map $TM'(x)$ is defined as follows:

$$\mathrm{TM'(x)} = \begin{cases} 1, & \mathrm{TM}(x) > 0.5, \\ 0, & otherwise. \end{cases} \tag{7}$$

The resulting class-specific binary maps, allow us to encode the spatial geometrical shape of each category. In Figure 6, we show example template maps obtained for different data set categories, together with their corresponding scene geometry instantiation.

For unknown test image, we spatially represent it using each of those class-specific binary maps. We then evaluate its performance towards each data set category using the appropriate class-specific spatial representation and the learned class-models. The whole process is demonstrated by the block diagram shown in Figure 7. We summarize the whole procedure in Algorithm 2.

**Algorithm 2** Adaptive Spatial Pyramids

1. **Require:** Binary Map *(BM)* for each category. Each $BM$ has a number of subregions $R^s$.
2. **Training from 2-4:** Construct a histogram $H_X^{i_s}$, for each $BM$ subregion $i_s$ of image $X$.
3. Train a Model with each $BM$ representation.
4. Repeat steps 2 to 4 for each category.
5. **Testing from 5-6:** For a test image, evaluate its performance for each category using its model & $BM$.
6. Assign the test image to category label with the highest score.
7. The matching between images $X$ and $Y$ for a subregion $s$ is given by:

$I^s(X,Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v))$.

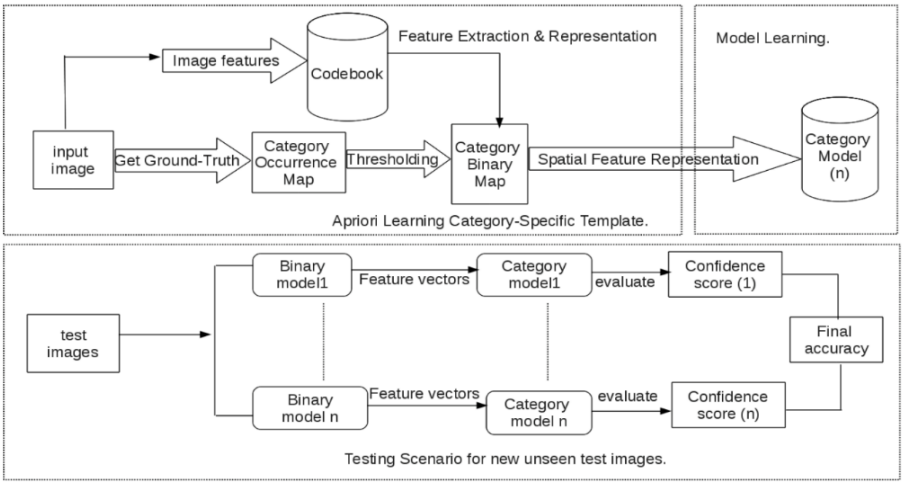


**Fig.7:** Outline of class-specific spatial template learning using Ground-truth. Note that the codebook models and the class-specific models are obtained off-line. For a test image, we evaluate its performance using the learned class-specific models.

***Selective Spatial Pyramids***

In this section, we propose an approach for learning the spatial partitioning based on the Agglomerative Information Bottleneck theory (AIB) [8], [9]. In fact, the main goal of AIB is to reduce the dictionary of visual words V required for representing the categories Z. This means generating a compact set of words $V^2$ from the original dictionary V so that the loss of mutual information to the categories Z is minimal:

$$I(V;Z) = D_{KL}\left[p(v,z)||p(v)p(z)\right] \tag{8}$$

The functional $D_{KL}\left[p(v,z)||p(v)p(z)\right]$ is the Kullback-Leiber divergence. The joint distribution $p(v,z)$ is estimated from the training set by counting the number of occurrences of each visual word in each category. The information about $v$ captured by $z$ can be measured by the mutual information. ,

$$I(V,Z) = \sum_k \sum_c p(v_k, z_c) log \frac{p(v_k, z_c)}{p(v_k)p(z_c)}, \tag{9}$$

measuring the discriminative power $I$ that one random variable carries about the other. The merging of visual words is achieved by iteratively applying AIB for fusing those two visual words $v_k$ and $v_j$ into $\hat{v}$ that causes the smallest decrease $\mathrm{I}(\hat{V};\mathrm{Z})$in the original mutual information$I(V;Z)$:

$$\delta I_z(V;\hat{V}) = I(V;Z) - I(\hat{V};Z). \tag{10}$$

At each step, AIB performs the best possible merge $argmin_{\hat{V}} = {}^{\delta} I_Z(V;\hat{V})$. We extend the original AIB to learn the most appropriate class-specific spatial structure. Concretely, we first over-segment the images as [19]. For each category, we learn on the validation set the most suitable split-up by pruning the least informative image segments based on AIB. Hence, we relate each image segment $((i_s)$ with the data set categories $Z$, such that the probabilities of each segment $p(i_s)$ is calculated by summing up the probabilities of the vocabularies $(v_u)$ it

contains as:

$$p(i_s) = \sum_{i_s=1}^{R} \sum_{u=(i_s-1)|V|+1}^{k|V|} p(v_u), \qquad (11)$$

where $i_s$ is the segment index, $R$ is the total number of image segments, $u$ is vocabulary index within segment $i_s$. To evaluate the Discriminative Power ($DP$) of the generated segments w.r.t. the categories, we use the information content criteria:

$$I(S,Z) = \sum_{i_s=1}^{R} \sum_{c=1}^{|C|} p(i_s, z_c) log \frac{p(i_s, z_c)}{p(i_s)p(z_c)}, \qquad (12)$$

where $c$ is the category index, and $C$ is the total number of data set categories. Finally, the loss in $DP$ is then obtained by:

$$\delta I_z(S;\hat{S}) = I(S;Z) - I(\hat{S};Z). \qquad (13)$$

Formally, for each category the best spatial split-up (partitioning) $s \in S$ is learned. Each split-up $s$ has a number of subregions $R(s)$. We propose that that the spatial partitioning matching function is created by computing the similarity between images $X$ and $Y$ for partitioning $s$ by:

$$I^s(X,Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v)). \qquad (14)$$

where the different sub-regions $i_s$ for the partitioning s correspond to a different image segment, such as sky, ground, etc. The whole process is demonstrated by the block diagram in Figure 8. We summarize the procedure of the selective search in Algorithm 3.

**Algorithm 3** Selective Spatial Pyramids

1. **Require:** a neighborhood matrix (N); defines the neighbors of each segment $i_s$.
2. Compute $DP$ of the current over-segmented shape, using Eq. (12).
3. Calculate the $DP$ loss to merge segment $i_s$ with its neighbors by Eq. (13).
4. Perform segment merge leading to the minimum $DP$ loss; generate split-up ($S$).
5. Evaluate $S$ score on the validation set.
6. Set the current shape to $S$ for next iteration.
7. Update the $N$ matrix to reflect the NS.
8. Repeat steps 3 to 7 until convergence (i.e., $N$ is empty).
9. Pick split-up $s \in S$ (has $R(s)$ segments) of highest score as the category model.
10. The similarity between images $X$ and $Y$ for same spatial split-up $s$ is given by: $I^s(X,Y) = \sum_{i_s=1}^{R(s)} \sum_{v=1}^{|V|} \min(H_X^{i_s}(v), H_Y^{i_s}(v))$.

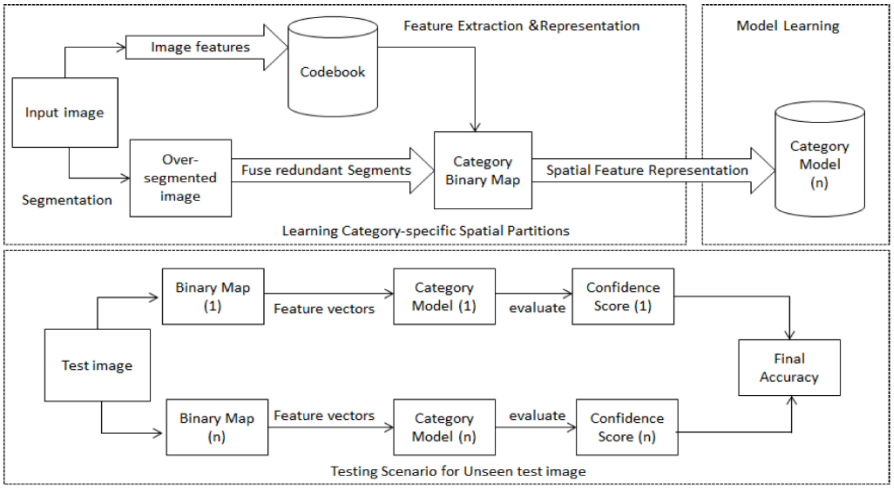


**Fig. 8:** Outline of class-specific spatial template learning using Ground-truth. Note that the codebook models and the class-specific models are obtained off-line. For a test image, we evaluate its performance using the learned class specific models.

In Figure 10, we visualize our *Selective SP* approach [32]. The Decision Tree *(DT)* shown in Fig. 10(b) represents the input image in Fig. 10(a). Each node in *DT* is equivalent to an image block (segment), while arrows indicate neighboring blocks that are candidates for fusion. Each block can be merged with either its right or its bottom neighboring block (if any). Initially, all possible block fusions indicated by arrows are considered for fusion. However, the actual fusion occurred is between $b_3$, and $b_6$ as it caused the minimum loss in discriminative power, see Fig. 10(c). As a result, $b_3$ is updated to $b_{(3,6)}$, and the neighbors of both $b_3$ and $b_6$ are inherited (i.e. $b_2$, $b_5$, $b_9$).
This iterative procedure results in generating various spatial configurations, and it terminates when all blocks are merged. Hence, it converges when it reaches the standard *BoW* representation.

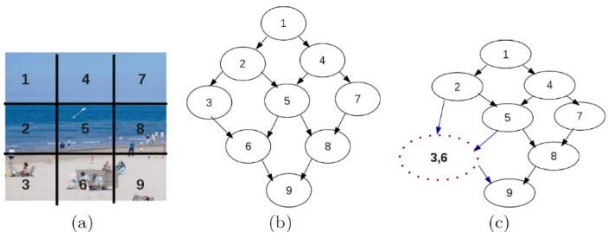


**Fig. 9:** The *Selective SP* Example. (a) Given an input (3*X*3) segmented image. (b) Each segment is represented by a node in a decision tree. (c) We calculate the discriminative power of each possible merging. See text for details.

## Experiments

In this section, the proposed methods to generate flexible spatial pyramids will be compared to the existing state-of art rigid based pyramids in the context of object recognition. In Sec. 5.1, the data sets used in all experiments are given. The experimental setup used is shown in Sec. 5.2.

**Data Sets**

Three independent data sets are used in the experiments. The first data set is a large dataset consisting of 3589 images classified as 15 different categories representing the standard generic scene geometries. 151 "sky+ background + ground", 333 "background + ground", 81 "sky + ground", 212 "ground", 139 "ground + Diag-BkgLR", 132 "ground + DiagBkgRL", 75 "diagBkgLR", 71 "diagBkgRL", 84 "box", 57 "1sidewallLR", 69 "1sidewallRL", 266 "corner", 960 "persBkg", 833 "noDepth", and 126 "tabPersBkg". Images are take are under large variety of lighting conditions and imaging conditions (including indoor, outdoor, desert, cityscape, and other settings). We refer to this data set as "stages data set". This data set is used to generate the binary mask maps (Sec. 3.2) used by the *Generic Spatial Pyramids* approach. Some example of images that are in this dataset are shown in figure 10.


**Fig. 10:** Example images of stages data set.

We also use Caltech-101 [10] and Pascal 2007 [6] datasets as benchmark data sets for evaluating our approach. The Pascal VOC 2007 data set [6] which consists of 9963 images of 20 different classes with 5011training images and 4952 testing images. The Caltech-101 data set which contains 9144 images of 102 different categories. Some example of images that are in Caltech and Pascal data sets are shown in figure 11(a) and figure 11(b), respectively.

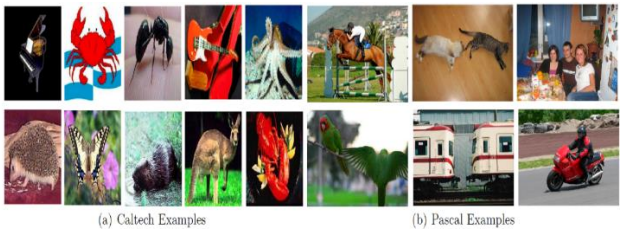


**Fig. 11:** Example images of Caltech and Pascal data sets

**Experimental Setup**

To compare the different spatial pyramids, a standard BoW image classification approach is used. SIFT features [20] of 16 × 16 pixel patches are used.

For Caltech-101, we use 30 images per category for the training and 50 for testing. The general architecture follows [5]. The SIFT descriptors are extracted on a dense grid rather than interest points, as this procedure has been shown to yield superior performance for scene classification [3]. We use a codebook of size 300. Experiments are conducted over 10 random splits of the data, and the average per-class recognition rates are recorded for each run. The final result is reported as the mean accuracy and its standard deviation from the individual runs.

For Pascal 2007, a standard multi-scale grid detector is taken together with a Harris-Laplace point detector [2], and a blob detector. SIFT descriptors are computed for all regions in the feature descriptor step, which are then quantized to a codebook of size 1000 visual-words. We used the standard average precision (AP) criteria to validate the performance on the object recognition task. The average precision is equivalent to the area under a precision-recall curve. Mean average precision (MAP) is used to evaluate the performance of the features over all the data set categories, which is obtained by averaging the AP over all categories. In our experiment, we compare our method with the standard three-levels Spatial Pyramid SP proposed by Lazebnik et al.[5] as the baseline.

**Generic Spatial Pyramids (Generic SP)**

In this section, the "Generic SP" approach proposed in Sec. 4.1 is evaluated using 3D scene geometries. For each category, the geometry with the highest score is selected for representing it. In table 1, we show the obtained scores compared with the standard SP. It is demonstrated that the Generic SP improves the results by 8.5% and 9.0% (relative to the baseline) on the Pascal and Caltech data sets, respectively. We attribute this to, (i) the generic 3D geometries contains a wide range of spatial partitionings which cover most of the "real-world" object categories; (ii) the obtained representations of our approach, are tailored for each category, and therefore can efficiently capture the variabilities that exists within each category.

**Table. 1:** Results obtained on Pascal (MAP score) and Caltech (Average per-class recognition rates) data sets using Generic SP. The proposed approach improves the scores over the standard SP proposed by Lazebnik et al.[5]. The best geometric split-up learned over multiple kernels *MKL* improves the overall performance significantly (see text).

| Method | Pascal | Caltech |
|---|---|---|
| Lazebnik et al. [5], linear kernel | 48.3 | $51.4 \pm 0.9$ |
| Lazebnik et al. [5], intersection kernel | 51.5 | $64.6 \pm 0.8$ |
| Generic SP, linear kernel | 52.5 (+8.7%) | $55.5 \pm 1.6$ (+8.0%) |
| Generic SP, intersection kernel | **55.9** (+8.5%) | $\mathbf{70.5 \pm 1.3}$ (+9.0%) |
| Generic SP + MKL | **59.7** (+16.0%) | $\mathbf{77.3 \pm 1.1}$ (+19.7%) |

In Figure 12, we show some example images for various data set categories together with their most appropriate 3D scene geometries. These quantitative results illustrate that different 3D scene geometries are selected for representing

the various data set categories. For instance, the plane category is instantiated from the sky+gnd scene geometry. While, the bird category is instantiated from the box geometry. Hence, using 3D scene geometries is important for efficiently capturing the spatial layout per category.


**Fig. 12:** Pascal data set examples with their learned geometries. Plane instantiated from sky+gnd, Bird from *box* geometry, TV monitor from *gnd+DiagBkgLR* and Cow from *sky+backgnd+gnd*

Another advantage of the "Generic SP" scheme is its ability of reducing the dimensionality of the generated histograms. The maximum number of partitions that exist for representing a category is 6 (i.e., "box geometry"+ BoW). This leads to a final representation of size 6×|V |, where |V | is the size of the vocabulary. On the other hand, the number of partitions of the standard "SP" is 21. This leads to a final representation of size 21 × |V |. Finally, we investigate the use of "Multiple Kernels Learning (MKL)" proposed by Gehler et al. [21], for the selection of the most appropriate geometry or the combination of geometries per category among multiple kernels. The results, in table1, demonstrate the importance of using multiple kernels for our approach. This improves the performance significantly by 16.0% and 19.7% (relative to the baseline) on Pascal and Caltech data sets, respectively.

***Adaptive Spatial Pyramids (Adaptive-SP)***

In this section, we evaluate the "Adaptive pyramid" method proposed in Sec. 4.2 which makes use of the Ground-Truth (GT) information of each category for learning its representative mask map. In table 2, we show the performance scores obtained on the Pascal data set based on the adaptive SP scheme w.r.t the standard three-levels Spatial Pyramid SP proposed by Lazebnik et al.[5] as the baseline. It is demonstrated that the "Adaptive SP" approach improves over the baseline SP. An improvement of 6.4% (relative to the baseline) is achieved. The main advantages of this method is the computational complexity of the learning approach as it is simple and fast to compute. However, it needs the GT, in order to obtain the mask maps needed for this approach. For this reason, we do not peruse this experiment on Caltech data set; as it does not contain the required *GT*.

**Table. 2:** MAP obtained on Pascal data set and *Average Recognition Rate* obtained on Caltech data set based on the *Adaptive SP*, the *GenericSP* and the *Selective SP* schemes.

| Method | Pascal | Caltech |
|---|---|---|
| Lazebnik et al. [5], intersection kernel | 51.5 | $64.6 \pm 0.8$ |
| Adaptive SP, intersection kernel | 54.8 (+6.4%) | - |
| Generic SP, intersection kernel | **55.9** (+8.5%) | $\mathbf{70.5 \pm 1.3}$ (+9.0%) |
| Selective SP, intersection kernel | **57.9** (+12.4%) | $\mathbf{73.6 \pm 1.5}$ (+14.0%) |
| Generic SP + MKL | **59.7** (+16.0%) | $\mathbf{77.3 \pm 1.1}$ (+19.7%) |

In the first column of Fig. 13, we show some example images from a number of object categories together with their selected partitioning based on the *Adaptive SP* approach. Interestingly, these quantitative results illustrated that the learned *adaptive SP* split-ups converge to one of the *Generic 3D* geometries. For instance, for the *Cow* category it is derived that its *TM* shape is an instantiation of its best performing *Generic* shape *(i.e., SkyBkgGnd)*. They share similar characteristics: the sky is located in the upper part of the image, the main foreground object is located within the middle part, while the lowest image part contains ground information.

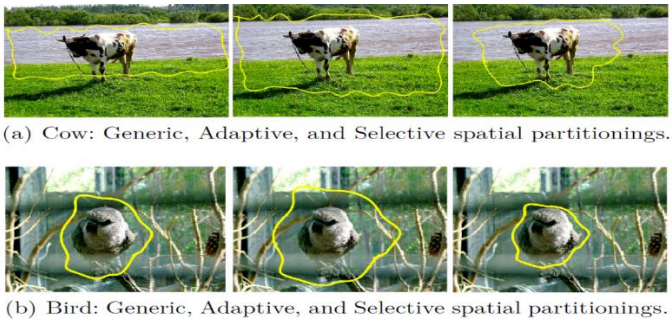


**Fig. 13:** Pascal data set examples for the spatial split-up learned based on (a)*Adaptive*, (b) *Selective* and (c) *Generic* SP approaches, respectively. Top row, shows the results on the "Cow category". Bottom row, shows the results on the "Bird category". The *Adaptive* generates coarser representations, *Selective* learns finer representations, *Adaptive* learns coarser representations, and *Generic* SP learns in-between representations. This justifies why improves over both the *Adaptive* and the *Generic* SP methods. Moreover, *Selective* and *Adaptive* partitions are instantiated from *3D* scene geometries (see text).

**Table. 3:** Average-Precision Results for all classes of the PASCAL VOC 2007 database. Comparison on the average accuracy of the original three level pyramid representation. The second and the third rows show the novel pyramid representations results using our Generic SP and Selective SP approaches, respectively.

| | plane | bike | bird | boat | bottle | bus | car | cat | chair | cow | table |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Pyramid | 68.4 | 51.8 | 37.7 | 64.4 | 19.8 | 44.2 | 71.8 | 51.9 | 47.9 | 31.7 | 39.7 |
| Generic | 74.0 | 57.1 | 45.4 | 66.7 | 26.2 | 54.2 | 75.0 | 57.2 | 46.0 | 38.3 | 41.6 |
| Selective | 75.6 | 57.9 | 46.4 | 67.2 | 28.0 | 56.0 | 77.0 | 59.0 | 48.0 | 40.0 | 42.6 |

| | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|
| Pyramid | 36.8 | 71.6 | 54.5 | 77.9 | 17.4 | 30.5 | 38.7 | 66.4 | 44.9 | 48.3 |
| Generic | 38.4 | 71.0 | 57.2 | 77.7 | 26.5 | 40.6 | 40.1 | 72.7 | 46.2 | 52.5 |
| Selective | 39.5 | 73.0 | 59.0 | 79.2 | 28.3 | 42.2 | 42.1 | 74.2 | 47.8 | 54.2 |

**Selective Spatial Pyramids (Selective SPs)**

In this experiment, we evaluate the *Selective SP* proposed in Section 4.3, in which the most appropriate "partitioning" is learned by eliminating the least informative partitions. The quantitative results, in table 1, show an improvement of 12.4% and 14% on Pascal and Caltech data sets, respectively. The main advantage of this method, is the ability to learn a compact yet discriminative partitioning (from its initially over-segmented images) which can efficiently fits each category.

The performance difference between the standard spatial pyramid and our proposed Generic and Selective pyramids using a vocabulary of size 1000 words. Table 3 shows a relative performance improvement of 8.7% and 12.2% on the Mean Average-Precision for having dynamic pyramid representations based on our approaches compared to those obtained using the standard three-level spatial pyramid on object recognition.

In Fig. 13, we show some example categories with their learned spatial partitioning based on the selective, the adaptive, and the generative SP approaches. These quantitative results demonstrate that the adaptive SP approach learns coarser spatial image representations, while the selective SP learns the finest representation. On the other hand, the Generic-SP approach captures an in-between spatial partitioning. This justifies the performance improvement obtained based on the selective-SP approach over the adaptive-SP, and the generative-SP approaches. It should also be noticed from these results that the generated spatial split-ups of both the selective, and the adaptive SP approaches are instantiations from the standard generic 3D geometries. For instance, the adaptive-SP (coarser shape) and the selective-SP (finer shape) partitionings learned for representing the "cow" and the "bird" categories; are instantiations from the "skyBkgGnd" and "box" geometries, respectively. We also show (in table 2) that our best results are based on "Generic SP + MKL" (a relative improvement of 16% vs. 12.4% for Selective SP to state-of-the-art SP results).

Finally, we compare the computational complexity between the proposed methods with respect to the standard SP in terms (i) SP levels, (ii) dimensionality and (iii) speed. The standard Lazebnik SP has 3 levels (i.e., BoW + 4+16) and a dimensionality of $21 \times V$. On the other hand, the Generic, Adaptive, and Selective SPs have an average of 2 levels (i.e., BoW + 2). This leads to a dimensionality of $3 \times V$ and 6×V for the average and worst case, respectively. Regarding the speed, experimentally a typical run using a vocabulary of size 1000 words on pascal 2007 data set, the classification time based on Lazebnik SP takes 220 seconds, while it takes around 72 seconds with our Selective SP or Generic SP approaches. Selective SP has the highest computational complexity, although it generates compact yet efficient spatial partitioning for each category. On the other hand, Generic SP approach balances between accuracy, dimensionality, and speed on both training and testing phases.

To summarize, there is a trade-off between the complexity of the learning algorithm, and the required accuracy. We conclude that the Generic SP approach balances between the accuracy and the computational complexity. Therefore, we consider it as our baseline approach. In the next section we compare this method with state-of-the-art methods.

**Comparison with State-of-the-Art**

In this section, we first compare the *Selective SP* with several recent related state-of-the-art approaches which do some implicit segmentation and build the representation of the image using these segmentations. We then investigate the *Generic SP* performance under varying vocabulary sizes. Finally, we compare our approach to state-of-the-art methods based on a single type of descriptor.

***Comparison: Related Approaches***

Comparison is done with several recent related state-of-the-art approaches of [22], [24], which incorporate some implicit segmentation and build the representation of the image using these segmentations. Concretely, [22] aims to recognize the scene type within image regions rather than the entire image, which is consistent with object detection more than object classification. On the other hand, the work of [23, 24] addresses the problem of the ambiguity of visual words which is partly caused by discarding spatial information. In particular, [23] aims to use category-specific color attention maps (using two different color descriptors, Color Name and HUE) to weight local shape features (e.g., SIFT features with 1000 visual words), obtaining a MAP score of 50.2% using three different features vs. 59.7% with SIFT features only (of 1000 visual words) using our approach (table 1). [24] also uses the idea of weighting local features. However, they adopt semantic contexts (rather than color) to generate attention maps and preserve the most discriminative context for visual word.

The method in [24] obtains an absolute improvement of 5.3% compared with the standard SP approach. While, our proposed method obtains an absolute improvement of 8.2% compared with the standard SP approach, see table 1. Moreover, the method in [24] obtains an accuracy 64.5% MAP using four different features with the standard SP representation. However, we obtain 63.6% MAP based on our proposed method with C-SIFT features only (see table 5). In summary, we demonstrated that our proposed approach outperforms the state-of-the-art methods using only one type of feature.

Further comparison, is done between the proposed approaches in this paper with recent work of [25], which also aims at addressing the fixed pre-defined partitioning assumption of the standard spatial pyramid that suits all the dataset categories. However, minor gain is obtained on Pascal VOC 2007 dataset compared to our approach (50.8% vs. 57.9% using 1000 visual words, see table 1). In particular, the learning approach of [25] recursively learn a space of grids starting with the full image as a grid of one cell. The finest grid possibly separates every pixel the image. They iteratively execute Gradient-descent to find the best grid at a certain depth. The resulting partitions are grids with pronounced boundaries. The learned grids outperform the standard SP at lower depths (i.e., depth 4: "5 partitions"). The algorithm tends to learn finer partitions (on average 5 partitions, not including BoW representation), and, hence having higher dimensionality.

**Discussion**: We attribute the better performance of our approach due to the fact that it tends to learn coarser spatial image partitions with smoother boundaries. This enables to capture the variations (scale, translation) that exist within each class. The use of the BoW representation helps in handling the background clutter and occlusions. In

particular, the learning algorithm of our Selective SP approach is initially based on an over-segmentation step. Hence, the boundaries of the generated segments are smooth in nature. The algorithm exploits the information theory in order to iteratively fuse the least informative segments. Moreover, we directly optimize over the Average Precision (AP) criteria till we reach the BoW image representation. Consequently, the spatial partitions are selected which give the highest AP score on the validation set. The resulting partitions tend to be coarse (on average 2 partitions + BoW representation) and, hence, having lower dimensionality. In conclusion, our approach has the ability to balance efficiently between invariance (not too fine) and discrimination power (not too coarse).

***Comparison: Multiple Vocabulary Sizes***

In Figure 14, we compare the performance of our approach to the state-of-art SP using various vocabulary sizes (i.e., 1k, 2k, 4k and 6k) on the Pascal data set. The results show that our approach outperforms the standard state-of-art SP over all the examined vocabularies. Similar results are obtained on Caltech datasets under various sizes. Moreover, our results confirm the experimental findings of the work of [21], where the use of MKL improves the overall performance for various vocabulary sizes.

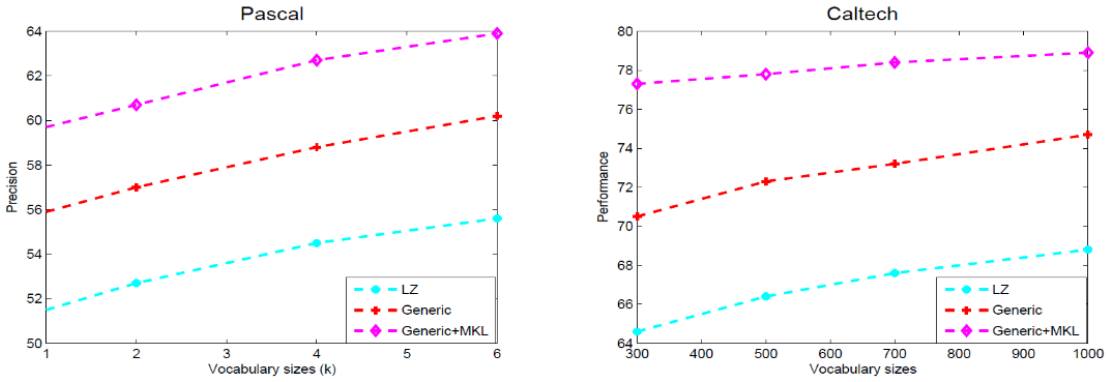


**Fig. 14:** Comparison between our *Generic SP* and *Generic SP + MKL* approaches with standard SP (denoted as *LZ*) using the SIFT descriptor under different vocabulary sizes on Pascal (left) and Caltech (right) data sets, (see text).

***Comparison: State-of-the-Art***

Comparison with previously published results obtained using one type of descriptor are shown in Table 4 and Table 5 for the Caltech and the Pascal data sets, respectively.

**Caltech.** A performance improvement of 4.5% is obtained based on our Selective SP approach w.r.t. the best performing method as shown in table 5. Moreover, an improvement of 9.8% is obtained based on our Generic SP + MKL approach.

Furthermore, we compare our approach with other recent work described in [29,26] which achieves comparable results with our approach using the standard spatial partitioning. The main difference between our work and the work in [29], [26] is that we use the classical bag-of-words approach with the standard K-means vector quantization within our novel spatial pooling approach. In contrast, they use an alternative coding scheme called Linear Coordinate Coding (based on sparse code and max pooling), which is shown recently to significantly outperforms the standard vector quantization scheme, leading to state-of-the-art performance on several benchmark datasets using a single type of descriptor. Therefore, we investigate the extension our proposed approach with the aforementioned new coding schemes. We show that the performance of our approach increases by using better coding (i.e., sparse codes) scheme. A relative performance improvement of 13.4% is obtained based on our Selective SP approach with respect to the best performing method as shown in table 5. Moreover, an improvement of 18.2% is obtained based on our Generic SP + MKL approach.

**Pascal.** Compared to the work of Van de Sande et al. [30] using the SIFT descriptor and a vocabulary of size 4k, an improvement of 7.9% is obtained based on our Generic SP. We obtain an improvement of 10.4% based on the Selective SP approach. An improvement of 15.0% is obtained based on our Generic SP + MKL approach using the SIFT features.

Compared with the best performing method using the C-SIFT descriptor (i.e., $\chi^2$ with a 2 kernel) in [30], we obtain a performance improvement of 3.0% based on our Adaptive SP approach. We also obtain an improvement of 5.5% and 8.3% based on our Generic SP and Selective SP approaches, respectively. Finally, an improvement of 12.4% is obtained based on the Generic SP + MKL approach. Note that again better performance has been reported with multiple descriptor types (e.g., SIFT, opponentSIFT, rgSIFT, C-SIFT, RGB-SIFT with $\chi^2$ kernel achieved 60.5% [30]). For this purpose, we evaluate our proposed approach using multiple descriptors (see table 5). We show that the final performance of our approach benefits from using multiple descriptors as demonstrated in [30]. In summary, we demonstrated that our proposed approach outperforms the state-of-the-art methods on both Caltech-101 and Pascal 2007 data sets using only one type of feature.We also show that the final performance of our approach increases by using multiple descriptors, and better coding (i.e., sparse codes) schemes.

**Table 4:** Results obtained by several recognition schemes using a single type descriptor and intersection kernel on Caltech data set, see text for details. The numbers shown inside brackets in [26] are the codebook sizes used in this work.

| | Method | Caltech |
|---|---|---|
| Boureau et al. [26] | SP + hard quantization [256] | $64.2 \pm 1.0$ |
| Boureau et al. [26] | SP + hard quantization + max pooling [256] | $64.3 \pm 0.9$ |
| Lazebnik et al. [5] | SP + hard quantization + kernel SVM | $64.6 \pm 0.8$ |
| Boureau et al. [26] | SP + soft quantization [512] | $66.1 \pm 1.2$ |
| Zhang et al. [27] | kNN-SVM | $66.2 \pm 0.5$ |
| Boureau et al. [26] | SP + sparse codes [1024] | $70.3 \pm 1.3$ |
| Boiman et al. [28] | Nearest neighbor + spatial correspondences | **70.4** |
| This paper | Selective SP + hard quantization [300] | $\mathbf{73.6 \pm 1.5}$ (+4.5%) |
| This paper | Selective SP + sparse codes [1024] | $\mathbf{79.8 \pm 1.2}$ (+13.4%) |
| This paper | Generic SP + hard quantization [300] + MKL | $\mathbf{77.3 \pm 1.1}$ (+9.8%) |
| This paper | Generic SP + sparse codes [1024] + MKL | $\mathbf{83.2 \pm 0.5}$ (+18.2%) |

**Table 5:** Comparison of our approach with state-of-art methods reported in literature on PASCAL VOC 2007 using a vocabulary of size $4000$ (see text).

| | Method | Descriptor | MAP |
|---|---|---|---|
| Lazebnik et al.[5] | SP | SIFT, intersection kernel | 54.5 |
| Van de Sande et al. [30] | SP | SIFT, $\chi^2$ kernel | 55.8 |
| Van de Sande et al. [30] | SP | C-SIFT, $\chi^2$ kernel | 56.6 |
| This paper | Generic SP | SIFT, intersection kernel | **58.8**% (+7.9%) |
| This paper | Generic SP | C-SIFT, intersection kernel | **59.7**% (+5.5%) |
| This paper | Generic SP | SIFT + C-SIFT, intersection kernel | **60.8**% (+11.6%) |
| This paper | Generic SP | SIFT + C-SIFT + RGB-SIFT + OpponentSIFT, *rg*SIFT | **63.6**% (+16.7%) |
| This paper | Adaptive SP | SIFT, intersection kernel | **57.2**% (+5.0%) |
| This paper | Adaptive SP | C-SIFT, intersection kernel | **58.3**% (+3.0%) |
| This paper | Adaptive SP | SIFT + C-SIFT, intersection kernel | **59.4**% (+8.9%) |
| This paper | Adaptive SP | SIFT + C-SIFT + RGB-SIFT + OpponentSIFT, *rg*SIFT | **62.2**% (+14.1%) |
| This paper | Selective SP | SIFT, intersection kernel | **60.2**% (+10.4%) |
| This paper | Selective SP | C-SIFT, intersection kernel | **61.3**% (+8.3%) |
| This paper | Selective SP | SIFT + C-SIFT, intersection kernel | **62.4**% (+14.5%) |
| This paper | Selective SP | SIFT + C-SIFT + RGB-SIFT + OpponentSIFT, *rg*SIFT | **65.3**% (+19.8%) |
| This paper | Generic SP + *MKL* | SIFT | **62.7**% (+15.0%) |
| This paper | Generic SP + *MKL* | C-SIFT | **63.6**% (+12.4%) |
| This paper | Generic SP + *MKL* | SIFT + C-SIFT | **64.7**% (+18.7%) |
| This paper | Generic SP + *MKL* | SIFT + C-SIFT + RGB-SIFT + OpponentSIFT, *rg*SIFT | **67.5**% (+23.9%) |

## Conclusion

Spatial Pyramids have been proposed which are steered by the 3D scene geometry. The geometry of a scene is measured based on image statistics taken from a single image. After the estimation of the scene geometry, the corresponding SP is selected as the geometrical representation. From large scale experiments on the Pascal VOC2007 and Caltech101, it can be derived that SPs which by selective search outperforms the standard SPs with 12.4% and 14.0% for Pascal VOC 2007 and Caltech101 respectively. The use of 3D scene geometry, to select the proper SP configuration, provides an even higher improvement of 16.0% and 19.7% respectively.

For future work, the proposed system will be extended to automatically learn a hierarchal class-specific adaptive shape model, where the highest levels will incorporate the important localization and/or segmentation knowledge for efficiently capturing the *ROI*, for restricting the objects spatial location to work with.

## References


1. Dance, L. Fan, J. Willamowski, C. Bray: Visual categorization with bags of keypoints., in: European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision, (2004).
2. K. Mikolajczyk, C. Schmid: A performance evaluation of local descriptors, Transaction on Pattern Analysis and Machine Intelligence 27 (10) 1615-1630 (2005).
3. L. Fei-Fei, P. Perona: A bayesian hierarchical model for learning natural scene categories, in: Computer Vision and Pattern Recognition (2005).
4. J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid: Local features and kernels for classification of texture and object categories: An in-depth study. A comprehensive study, International Journal of Computer Vision 73 (2), 213–218 (2007).
5. S. Lazebnik, C. Schmid, J. Ponce: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Computer Vision and Pattern Recognition (2006).
6. M. Everingham, L. V. Gool, C. K. I.Williams, J.Winn,A. Zisserman: The pascal visual object classes challenge 2007 results. (2007).
7. V. Nedovic, A. W. M. Smeulders, A. Redert, J.-M. Geusebroek.,Stages as models of scene geometry., IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (9), 1673–1687 (2010).
8. N. Slonim, N. Tishby: Agglomerative information bottleneck, Neural Information Processing Systems, (1999).
9. B. Fulkerson, A. Vedaldi, S. Soatto: Localizing objects with smart dictionaries, European Conference of Computer Vision (2008).
10. L. Fei-Fei, R. Fergus, P. Perona.: Learning generative visual models from few training examples., Computer Vision on Pattern Recognition Workshop (2004).

11. D. Hoiem, A. A. Efros, M. Hebert: Geometric context from a single image, International Conference on Computer Vision, pp. 654–661 (2005).
12. E. Delage, H. Lee, A. Y. Ng.: A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image, Computer Vision and Pattern Recognition, pp. 2418–2428 (2006).
13. E. Sudderth,Torralba, W. Freeman, A. Willsky: Depth from familiar objects: A hierarchical model for 3d scenes, Computer Vision on Pattern Recognition, pp. 2410–2417 (2006).
14. V. Nedovic, A. Smeulders, A. Redert, J.-M. Geusebroek: Depth information by stage classification, International Conference on Computer Vision (2007).
15. M. Marszalek, C. Schmid, H. Harzallah, J. van de Weijer: Learning object representation for visual object class recognition, in: Visual recognition Challenge Workshop, International Conference on Computer Vision (2007).
16. J. van Gemert: Exploiting photographic style for category-level image classification by generalizing the spatial pyramid, International Conference on Machine Learning, (2011).
17. K. Grauman, T. Darrell.,:The pyramid match kernel: Discriminative classification with sets of image features., International Conference on Computer Vision, (2005).
18. L. Rui, A. Gijsenij, T. Gevers, V. Nedovic, X. De, J. Geusebroek.,: Color constancy using 3d scene geometry, International Conference on Computer Vision (2009).
19. A.P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, G. Jones, Superpixel lattices, in: Conference on Computer Vision and Pattern Recognition (2008).
20. D. Lowe., Distinctive image features from scale invariant keypoints, International Journal of Computer Vision 60 (2), 91–110 (2004).
21. P. V. Gehler, S. Nowozin.: On feature combination for multiclass object classification, International Conference on Computer Vision, (2009).
22. J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba: Largescale scene recognition from abbey to zoo, Computer Vision and Pattern Recognition (2010).
23. F. Khan, J. van de weijer, M. Vanrell: Top-down color attention for object recognition, International Conference on Computer Vision, (2009).
24. Y. Su, F. Jurie: Visual word disambiguation by semantic contexts, International Conference on Computer Vision (2011).
25. G. Sharma, F.Jurie: Learning discriminative spatial representation for image classification, British Machine Vision Conference (2011).
26. Y. Boureau, F. Bach, Y. LeCun, J. Ponce.: Learning midlevel features for recognition, Computer Vision and Patten Recohnition (2010).
27. H. Zhang, A. C. Berg, M. Maire, J. Malik., Svm-knn: Discriminative nearest neighbor classification for visual category recognition, Computer Vision and Pattern Recognition (2006).
28. O. Boiman, I. Rehovot, E. Shechtman, M. Irani., In defense of nearest-neighbor based image classification, Computer Vision and Pattern Recognition, (2008).
29. J. Yang, K. Yu, Y. Gong, T. Huang: Linear spatial pyramid matching using sparse coding for image classification, Computer Vision and Pattern Recognition (2009).
30. K. E. A. van de Sande, T. Gevers, C. G. M. Snoek:, Evaluating color descriptors for object and scene recognition, Transaction of Pattern Analysis and Machine Intelligence, 32 (9), 1582–1596 (2010).
31. N. Elfiky: Application of Analytics in Machine Vision using Big Data, Asian Journal of Applied Sciences 7 (4) 376–385 (2019).
32. N. Elfiky: A Novel Spatial Layout Representation for Object Recognition, International Conference on Artificial Intelligent and Computer Vision (AICV), 2020.