Scene Modeling Using Co-Clustering

Jingen Liu
Computer Vision Lab
University of Central Florida
liujg@cs.ucf.edu

Mubarak Shah
Computer Vision Lab
University of Central Florida
shah@cs.ucf.edu

Abstract

In this paper, we propose a novel approach for scene modeling. The proposed method is able to automatically discover the intermediate semantic concepts*. We utilize Maximization of Mutual Information (MMI) co-clustering approach to discover clusters of semantic concepts, which we call* intermediate concepts*. Each* intermediate concept *corresponds to a cluster of* visterms *in the Bag of Visterms (BOV) paradigm for scene classification. MMI co-clustering results in* fewer *but meaningful clusters. Unlike* k*-means which is used to cluster image patches based on their appearances in BOV, MMI co-clustering can group the* visterms *which are highly correlated to some concept. Unlike probabilistic Latent Semantic Analysis (pLSA), which can be considered as one-sided soft clustering, MMI co-clustering simultaneously clusters visterms and images, so it is able to boost both clustering. In addition, the MMI co-clustering is an unsupervised method. We have extensively tested our proposed approach on two challenging datasets: the fifteen scene categories and the LSCOM dataset, and promising results are obtained.*

1. Introduction

The scene is defined as the physical setting of the environment where the image is taken. Some examples of scenes include outdoor, indoor, beach, mountain, forest, office and urban landscape. Image scene classification has a wide range of applications, such as intelligent image processing and content-based image indexing and retrieval (CBIR)[20]. In CBIR, an efficient and effective classification method can significantly improve the retrieval accuracy by removing the irrelevant images. Scene classification has posed a significant challenge to the research community of computer vision due to interclass variability, illumination and scale changes.

In general, we can model a scene from the hierarchical viewpoint. On the bottom level, a scene can be modeled as a statistical distribution of color of pixels or interest patches. Yet beyond the low level, we can also describe a scene by the composition of objects such as cars, buildings, and persons. The objects can be further described in terms of parts e.g. a wheel of a car, a window of a building, or a face of a person. The highest level could be the scene as a whole.

Earlier scene modeling approaches mainly focused on modeling a scene using the global statistical information of an image rather than the local details [20] [16]. However, these approaches were not extensible to multi-category classification [15]. Later some approaches were proposed to classify images into multiple categories e.g. [14] and [21]. The key idea of these approaches is to utilize "semantic concepts" to represent an image. These "semantic concepts" are objects or object patches in the scene. However, in this work "semantic concepts" were determined by *manual annotations*. Hence, these approaches are less flexible to be applicable to other sematic categories.

Local interest-points and their descriptors have attracted lots of attention. Bag of Visterms (BOV) approaches, which have achieved inspiring performance [3] [10], model images as sets of orderless local features. The key process in BOV modeling is to quantize the local image patches into *visterms* using k-means algorithm, which clusters the patches based on appearance similarity. It has been noticed that the size of the codebook affects the performance and there is an optimal codebook size which can achieve maximal accuracy [13][22]. In general, thousands of *visterms* are used to achieve better performance. However, they may contain a large amount of information redundancy. Therefore, the researchers have attempted to find a more compact representation. Winn *et al* proposed an agglomerative Information Bottleneck (IB)[19] based method to generate optimized codebook size by merging the initial large number of visterms [23]. This is a supervised procedure which needs to manually label the training regions. Latent Dirichlet Allocation (LDA) [1] or probabilistic Latent Semantic Analysis (pLSA) [7] modeling is another attempt. Fei-Fei *et al.* [5], Quelhas *et al.* [15], Sivic *et al.* [18] and Bosch *et al.* [2] have respectively applied LDA and pLSA to discover latent semantic concepts beyond the BOV. Those models were

originally used in the text processing field [1][7], then successfully applied to scene classification and object recognition in Computer Vision. Here, an image is modelled as the distribution of the *hidden concepts* that can be essentially considered similar to the "semantic concepts" in [21]. The difference is that *hidden concepts* can be discovered automatically from document-word or image-visterm co-occurrence matrix.

In this paper, we propose a novel approach for automatically discovering intermediate concepts from visterms by Maximization of Mutual Information (MMI). Recently, information-theoretic clustering has received more attention in data clustering [6][8][19]. Co-clustering via the Maximization of Mutual Information (MMI) is a successful strategy to group words into semantic concept clusters (e.g. "pitching", "score","teams" etc. can be clustered into "baseball" concept; and "biker", "wheel", "ride" may be clustered into "motorcycle" concept.), which has been successfully used in text classification area[4]. The critical point is to simultaneously maximize the mutual information (MI) of the words and documents when clustering these words into semantic concepts, which are somehow analogous to the *hidden concepts*. However, there are significant differences between them. pLSA is a generative model, which employs hidden variables; while MMI co-clustering does not use hidden variables. Secondly, pLSA assumes conditional independence, i.e. given the latent variable the image and *visterm* are independent which are not required in MMI co-clustering. Besides, MMI co-clustering performs hard clustering, and it simultaneously clusters both words and documents. Moreover, in practice we have observed that pLSA needs a considerable number of EM iterations to reach convergence.

Similarly, the IB also can preserve the MI between words and documents. However, it is one-sided clustering. Even the double IB [19] only *sequentially* clusters words followed by the documents. It does not guarantee a global minima of loss function. However, MMI co-clustering systematically gives the global minimum of the loss of MI, and it has been proved the loss function of MI is monotonic[4].

1.1. Proposed Framework

Fig.1 shows the workflow of our framework for both learning and classification. Other than using MMI co-clustering technique to automatically discover *intermediate concepts*, we also investigate ways to capture the spatial information of the *intermediate concepts*. We form a codebook from a collection of local patches sampled from the training images using k-means algorithm which can efficiently group visually similar patches into one cluster (*visterm*). And then we use MMI co-clustering to further cluster the *visterms* into some *intermediate concepts*(unsupervised). Unlike k-means, MMI co-clustering can group the *visterms* which are highly correlated to some

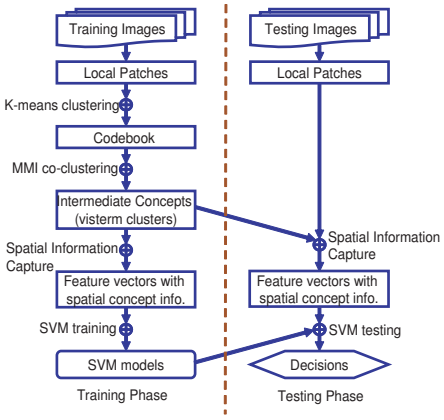

Figure 1. Work flow of the proposed scene classification framework.

concept. In order to capture the spatial information of the semantic concepts in the scene, we exploit the Spatial Pyramid Matching (SPM)[12] and weighted Spatial Concept Corellogram (SCC). Finally, we use a SVM as a classifier to train and test these models.

We have tested our approach on two diverse database: the 15 semantic scene categories [12] and the LSCOM dataset [1](we provide the details in section 3). Our results show that MMI co-clustering (Bag of concepts: BOC) can clearly achieve much better performance than clustering obtained by k-means algorithm (Bag of Visterms: BOV); this improvement is quite significant especially when the number of clusters (*visterms* or *intermediate concepts*) is small. We have also explored the different possible cases (different sampling distances, strong vs weak classifier, different number of clusters etc) under which MMI co-clustering can achieve much better or competitive results compared to original BOV. This is due to the fact that MMI Co-clustering generates fewer but more meaningful clusters of visterms called *intermediate concepts*. Besides, we also apply the learnt *intermediate concepts* to two types of spatial models: SPM and weighted SCC. The experiments verify that SCC further improves the results over MMI co-clustering, and SCC+BOC gives better performance than SCC and BOC. Besides, SPM using *intermediate concepts* can also improve the performance of SPM using *visterms* from 2% to 5% in terms of average accuracy. Finally, we would like to note that the proposed *intermediate concepts* model with spatial information can achieve competitive performance with much lower dimensions compared to that of *visterms* model (see the section 3 for the details). Lower dimension is quite important for computation speed, especially for a large dataset like LSCOM.

The rest of this paper is organized as follows: Section 2 describes the MMI co-clustering. Section 3 presents the experimental results and comparisons with other approaches.

[1]http://www.ee.columbia.edu/ln/dvmm/lscom/

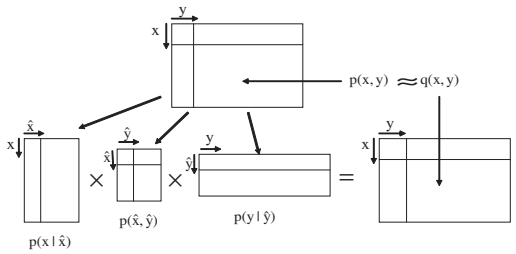

Figure 2. The graphical explanation of MMI co-clustering. The goal of MMI co-clustering is to find one clustering of X and Y which minimize the distance between the distribution matrix p(x,y) and q(x,y).

Finally, Section 4 concludes our work.

2. Co-clustering by Maximization of Mutual Information

In this section, we present details on how co-clustering of visterms and images is performed by maximizing the mutual information. Consider two discrete jointly distributed random variables X and Y, where $X \in \mathcal{X} = \{x_1, x_2, ..., x_n\}$ and $Y \in \mathcal{Y} = \{y_1, y_2, ..., y_m\}$. In practice, $\mathcal{X}$ may represent a set of words in text classification or *visterms* in image classification, and $\mathcal{Y}$ may be a set of documents or images. In scene classification based on BOV modeling, the similarity of two images can be measured by their *visterm* conditional distributions $p(x|y)$. One critical procedure for BOV modeling is to form codebook X via vector quantization using k-means algorithm, which groups the local patches by their appearance similarity. If codebook size is small, it may cause over-clustering with higher intraclass distortion. Therefore, it is common to choose an appropriate larger value of codebook size. However, this large size may introduce information redundancy in the co-occurrence matrix.

So, we seek to find a more compact representation of X, say $\hat{X}$ which is able to capture the "semantic concepts". This procedure is called "word clustering" in text classification. One criteria for $\hat{X}$ is to maximize the mutual information $I(\hat{X}; Y)$. Since our original goal is to cluster Y, we can simultaneously perform clustering on X and Y by maximization $I(\hat{X}; \hat{Y})$.

2.1. Mutual Information

Given two discrete random variables X and Y, the MI between them is defined as:-0.1in

$$I(X;Y) = \sum_{y \in Y, x \in X} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \qquad (1)$$

where $p(x, y)$ is the joint distribution of X and Y, $p(x)$ and $p(y)$ are the probability distributions of X and Y respectively. MI is used to measure the dependence of two random variables, that means how much information of variable X is contained in variable Y. Using Kullback-Leibler divergence, also known as relative entropy, the MI also can be expressed as:

$$I(X,Y) = D_{KL}(p(x,y) \parallel p(x)p(y)), \qquad (2)$$

where D_{KL} computes the distance between two distributions. Thanks to BOV image representation, we can easily make an analogy between images (*visterms*) and documents (words). As noted earlier in the context of this paper, X and Y represent *visterm* and image respectively.

2.2. Co-clustering Algorithm

Consider a training image dataset $\mathcal{Y}$ with c categories, and its associated codebook $\mathcal{X}$ with n *visterms*, we seek to simultaneously cluster Y into c categories $\hat{\mathcal{Y}} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_c}\}$, and X into w disjoint clusters $\hat{\mathcal{X}} = \{\hat{x_1}, \hat{x_2}, ..., \hat{x_w}\}$. Actually, we can consider the clustering as two mapping functions $\hat{X} = C_X(X)$ and $\hat{Y} = C_Y(Y)$. In order to evaluate the quality of clustering, we utilize the following mutual information loss:

$$\Delta MI = I(X;Y) - I(\hat{X};\hat{Y}). \qquad (3)$$

Because $I(X;Y)$ is fixed for specified data collections, the optimal co-clustering actually attempts to maximize $I(\hat{X};\hat{Y})$, given the number of clusters c for Y, and w for X respectively. It is straightforward to verify that the MI loss also can be expressed in the following form [4]:

$$\Delta MI = D_{KL}\big(p(x,y) \parallel q(x,y)\big), \qquad (4)$$

where $q(x,y) = p(\hat{x},\hat{y})p(x|\hat{x})p(y|\hat{y})$. This is the objective function when performing co-clustering. The input to co-clustering algorithm is the joint distribution $p(x, y)$, which records the probability of occurrence of a particular *visterm* x in a given image y. The aim is to determine clusters with distribution $q(x, y)$ which is as close as possible to $p(x, y)$. The process is pictorially shown in Figure 2. For each new clustering $\hat{X}$ and $\hat{Y}$, we first compute the joint distribution matrix $p(\hat{x}, \hat{y})$ as follows:

$$p(\hat{x},\hat{y}) = \sum_{x \in \hat{x}, y \in \hat{y}} p(x,y). \qquad (5)$$

Then for $x \in \hat{x}$ we compute the conditional distribution $p(x|\hat{x})$,

$$p(x|\hat{x}) = \frac{p(x)}{p(\hat{x})}, \qquad (6)$$

where the marginal distribution $p(x) = \sum_{y \in Y} p(x, y)$ and $p(\hat{x}) = \sum_{\hat{y} \in \hat{Y}} p(\hat{x}, \hat{y})$. For $x \notin \hat{x}$, $p(x|\hat{x}) = 0$. Similarly, we can get the conditional distribution $p(y|\hat{y})$. Consequently, the quality of this specified clustering is evaluated by $D_{KL}\big(p(x, y) \parallel q(x, y)\big)$.

The algorithm starts with randomly initial partitions C_X^0 and C_Y^0. The number of clusters for *X* and *Y* are specified as *w* and *c* respectively. At each iteration t of the algorithm, two phases are involved:

1. Clustering of X while keeping Y fixed. For each x, assign it to its new cluster, which means $C_X^{t+1} = argmin_{\hat{x}} D_{KL}\big(p(y|x) \parallel q(y|\hat{x})\big)$ where $q(y|\hat{x}) = p(y|\hat{y})p(\hat{y}|\hat{x})$. Update the probabilities based on the new X cluster.

2. Clustering of Y while keeping X fixed. For each y, find its new cluster such that $C_Y^{t+2} = argmin_{\hat{y}} D_{KL}\big(p(x|y) \parallel q(x|\hat{y})\big)$, where $q(x|\hat{y}) = p(x|\hat{x})p(\hat{x}|\hat{y})$. Update the probabilities based on the new Y cluster.

The iterations of the co-clustering stops when

$$\Delta^t MI - \Delta^{t+2} MI < \epsilon, \tag{7}$$

where ϵ is the threshold.

In summary, in order to assign *intermediate concepts* to each image patch, we apply two steps. We first use k-means algorithm to cluster the image patches into *visterms*. Since the criterion for k-means is based on appearance similarity, patches belonging to one *visterm* are visually similar. Further, we group the *visterms* into some semantic clusters (*intermediate concepts*) via MMI co-clustering. The number of *intermediate concepts* is much less than that of *visterms*. Our experiments show that we can do better scene classification using *intermediate concepts* than using *visterms*.

3. Experiments

We have extensively applied our proposed approach to two diverse datasets: fifteen scene categories [12] and the LSCOM (Large Scale Concept Ontology for Multimedia) dataset. For both datasets, only gray level images are used. The default experiment setting is as follows. We utilize dense features sampled using a regular grid with sampling space of M=8 pixels. The patch size is randomly sampled between scales of 10 to 30 pixels. SIFT descriptor [9] is computed for each patch. We use a support vector machine (SVM) with Histogram Intersection kernel as a classifier. For 15 scene categories, we choose the one-versus-all methodology for multi-class classification. The binary SVM classification is applied to the experiments on LSCOM. All the experiments on 15 scene categories are repeated 5 times. For each experiment the training dataset is randomly selected. The final results are reported as the average accuracy.

3.1. Classification of Fifteen Scene Categories

The fifteen scene categories are the same used by [12], which is union of the 13 scenes reported in [5] and two additional scenes added by Lazebnik et al. In fact, the thirteen categories contain 8 scenes originally reported in [14]. Each category has 212∼410 images. The average image size is about 250 × 300. We use 50 randomly selected images from each category to form the *visterm* codebook of size N_v. Further, we use MMI co-clustering to discover N_c *intermediate concepts* from the codebook. We try several N_v, and finally fix $N_v = 1,500$ which gives better performance. Then an image can be represented by *visterms* histogram (BOV model) or a *intermediate concepts* histogram (BOC model). In SVM classification phase, 100 images are randomly selected from each category as a training set, and the rest are used for testing.

N_c / N_v	20%	60	80	100	200	300
BOV	47.25	61.69	65.34	67.72	70.81	71.46
BOC	63.32	68.53	70.25	73.01	75.16	74.62

Table 1. The average accuracy (%) achieved using strong and weak classifiers.

3.1.1 Classification using orderless features

We investigate the gain of MMI co-clustering (BOC model) compared to the k-mean approach (BOV model) in two ways. One is to compare them using the same number of clusters($N_c = N_v$), and the other is to compare BOC with the original BOV with N_v=1,500 (*Original BOV means directly representing an image using bag of visterms from which the* intermediate concepts *are created.*). We conduct classification on the 15 scene categories using both BOV and BOC models by using different values of N_v or N_c from a set: $\{20, 60, 80, 100, 200, 300\}$. Table 1 shows the results. Overall, BOC is able to improve the performance between 3.16% to 16.07% compared to BOV; especially when the number of clusters is small. This is due to better clustering. K-means algorithm groups the image patches into *visterms* based on the appearance of the patches. When N_v is small, the intra-cluster variance is larger, which hurts the performance. However, when grouping the 1,500 *visterms* into semantic intermediate concept clusters, MMI co-clustering tries to preserve the mutual information between *visterms* and images, such that the *visterms* in the same cluster share certain common *intermediate concept*. So that they are not necessarily similar in visual appearance. Although in MMI co-clustering intra-cluster variance of appearance may be large, it can preserve some meaningful concepts. Therefore, MMI co-clustering can still achieve better classification performance even with small N_c. The best performance for BOC is achieved when $N_c = 200$.

The classification accuracy is 76.38% when using BOV model with $N_v = 1,500$, which is slightly better than the best performance of BOC model. This is consistent with the results of Lazebnik *et al.*[12] and Quelhas *et al.* [15]. We conjecture that this improvment may be due to the dimension reduction achieved by the MMI co-clustering technique. While Bosch *et al.* claimed in their paper [2] that compared to original BOV, pLSA which is another dimension reduction technique similar to our MMI co-clustering, performs better. We feel that the gain in performance due

to the dimension reduction depends on classifier type and the performance of original BOV. The performance of BOV can vary with the patch sampling [13], and the number of *visterms* [22]. If the patch sampling and N_v has been optimized, it is not easy to achieve higher accuracy with any dimension reduction techniques, because BOV representation is not that sensitive to noise. Another reason may be due to the performance of the specific classifier. Some weak classifiers like K-Nearest Neighborhoods (KNN) perform poorly with high dimensional features. Therefore, when the dimension is reduced, they are able to achieve better performance. However, some strong classifiers (i.e. SVM) which are good at classification of high dimensional features, may not be able to achieve better performance with dimension reduction because of certain information loss.

In order to verify our conjecture, we conducted two groups of experiments. Table 2 shows the results using different sampling spaces denoted by M. Here, multi-class SVM is used as a classifier. The first row lists the results using BOV model with N_v=1,500, and the second row shows the results using BOC model where the *intermediate concepts* are extracted from the corresponding BOV codebook. When we increase the sampling space, the difference between the performance of BOV and BOC decrease from about 3.6% to -1.33%. In particular, the sampling setting in the third column is similar to the sampling in [2], and the performance of BOC is better than BOV. In fact, large sampling space generates fewer sampling features. Space with M=4 corresponds to more than 4,000 patches for each image, while space with M=10 correspond to only about 500 patches. Therefore, we feel that with a large number of sampling patches, the BOV performs better. Our further experiments verified this. For M=8 (each image has more than 1,000 patches), we randomly select about 200 patches from each image to evaluate the performance. Then the results for BOV and BOC are 67.29% and 69.21% respectively. Therefore, the number of patches sampled from the image affects the comparison between BOC and original BOV.

We further investigated the performance of classification using different classifiers. Table 3 demonstrates the performance comparison of the SVM classifier and the KNN classifier with Euclidian distance. In both cases, 100 randomly selected images from each category were used for training. The first column with N_v=1,500 shows the BOV baseline, and the following column shows the result of BOC with different N_c. It is very clear that the KNN classifier does not work well for high dimensional data. Hence, the dimension reduction technique can improve the performance quite much. However, SVM is a strong classifier which is able to handle high dimensional data. With reasonable N_c, the SVM can still achieve competitive results. However, low dimensional features provides us much better computational efficiency, which is very important for learning/classification of a large dataset like LSCOM.

	M = 4(%)	M = 8(%)	M = 10(%)
BOV(N_v=1,500)	78.32	76.38	69.81
BOC(N_c=200)	74.69	75.16	71.14

Table 2. The results achieved under different sampling spaces.

$N_c \setminus N_v$	1500	40	60	80	100
SVM	76.38	64.60	68.53	70.25	73.01
KNN (K=12)	58.17	61.22	63.76	64.54	66.37

Table 3. The average accuracy (%) achieved using strong and weak classifiers.

Finally, we compared the performance of MMI co-clustering, pLSA and IB. For all of them, we use the default experiment setting. pLSA achieves the best performance of 71.24% at $N_c = 80$. The best performance of IB is 72.49% when $N_c = 150$, while MMI co-clustering can achieve 75.16% at $N_c = 200$. Besides, in the experiments we observed that pLSA converged after about 100 iterations, while MMI co-clustering can converge in less than 40 iterations. This is consistent with the claim in [11] that in practice it takes a considerable number of of EM iterations for pLSA to converge. The time complexity for co-clustering, pLSA and IB are $O(t \cdot R \cdot (c + k))$,$O(t \cdot R \cdot k)$ and $O(|I|^3)$ respectively (where t is number of iterations, R is the number of nonzero entries, c is number of categories, k is number of *intermediate concepts* and I is number of training images). Therefore, IB is not suitable for large dataset[19].

3.1.2 Classification using intermediate concepts and their spatial information

In order to capture the spatial information, we implement two models: Spatial Pyramid Matching (SPM)[12] and Spatial Concept Correlogram (SCC). For SPM, we repeatedly divide an image into subblocks and compute local histogram of *intermediate concepts* for each block. Finally, an image is represented by combining the local histograms from the subblocks of the pyramid. The representative vector has high dimensions of $\frac{1}{3}(4^L - 1)N_c$, where L is the number of pyramid levels. In our experiments, we set $L = 3$. From table 4, we can see thanks to *intermediate concepts*, the SPM_IC (SPM using *intermediate concepts*) can improve the performance from about 2.79% to 4.28% , especially when the number of cluster is smaller. Interestingly, we notice that when $N_c = 80$, the SPM_IC can achieve competitive performance to SPM_V (SPM using *visterms*) at $N_v = 400$, while the dimension is reduced by 5 times. The best performance achieved by SPM_V is 80.46% [2] and 83.25% for SPM_IC.

We modify the correlogram and make it fit to our SCC, which is able to capture the spatial correlation of the *intermediate concepts* in the image. It represents the probability

[2] In [12] the best performance for SPM_visterm is quoted as 81.4%.

	40	50	80	100	200	400
SPM_V	75.24	76.14	77.62	77.81	80.27	80.46
SPM_IC	79.52	80.19	80.93	81.33	83.19	83.25

Table 4. The performance (average accuracy %) of SPM using *visterms* and *intermediate concepts*. SPM_IC and SPM_V denote SPM using *intermediate concepts* and *visterm* respectively.

	40	60	80	100	200	400
BOC	65.48	68.53	70.25	73.01	75.16	74.21
SCC	65.97	69.71	72.40	74.39	77.76	78.15
BOC+SCC	71.10	73.06	75.18	78.33	81.49	81.72

Table 5. The average classification accuracy (%) obtained by various models (SCC, BOC, and SCC+BOC).

of two patches at a distance D having the same *intermediate concept*. We can define the SCC as follows,

$$\mathcal{R}(D_k, l_i, l_j) = \mathbf{Pr}\big(l(p_2) = l_j | l(p_1) = l_i, d(p_1, p_2) \in D_k\big), \tag{8}$$

where p_1, p_2 are two patches, l_i, l_j are two concept labels, and D_k represents the quantized distance. Assume $\mathcal{R}_1$ and $\mathcal{R}_2$ respectively represent SCC of image 1 and image 2, then the similarity between them is computed as,

$$Sim(\mathcal{R}_1, \mathcal{R}_2) = \sum_{k=1}^{K} \sum_{i,j=1}^{L} w_k \times min(\mathcal{R}_1(D_k, l_i, l_j), \mathcal{R}_2(D_k, l_i, l_j)), \tag{9}$$

where w_k is the weight assigned to the matches made at distance D_k. Thus, we can assign a higher weight to the match found at a smaller distance.

In our experiments, we consider autocorrelogram. When computing the SCC, we divide the image into 2 by 2 blocks, and for each block, we compute its SCC. We set $D_1 = [1\ 64]$ and $D_2 = [64\ 128]$ in term of pixels in x and y direction. Table 5 shows the classification results. We can see the combination of SCC and BOC can achieve better performance over SCC and BOC. Interestingly, *correlatons* reported in [17] performs much worse than BOV. However, our SCC performs better. This might be due to the fact that our *intermediate concepts* correlogram is computed on patches, and weighted by different quantized distances. The best performance for SCC+BOC is 81.72%, which is little worse than SPM_IC, but better than SPM_V. The number of dimensions, $9N_c$, is much lower than that of SPM_IC. Fig.3 shows the confusion table for the 15 sceene categories using SCC+BOC approach.

3.2. Classification of LSCOM Dataset

The LSCOM dataset which includes more than 400 annotated categories is a very challenging dataset and has has been explored by the TRECVID community for several years[3]. This dataset contains 61,901 keyframes extracted from a variety of real TV news programs. The

[3]http://www-nlpir.nist.gov/projects/trecvid/

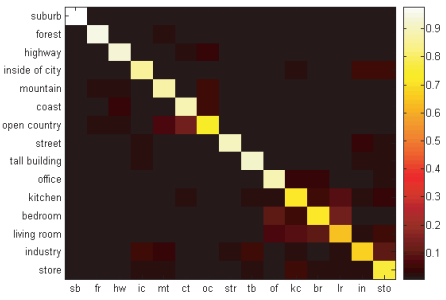

Figure 3. Confusion table for the SCC+BOC model. The average performance is 81.72%.

size of the keyframe is fixed to 240 × 352. In our experiments, the following 28 categories including scenes and objects are evaluated: airplane, animal, basketball, boat or ship, building, charts, clouds, weather, crowd, desert, flag-US, maps, meeting, military, mountain, road, studio, tennis, trees, urban, waterscape, computer_TV-screen, explosion or fire, industrial setting, car, fields, office and vegetation. In this experiment, we want to demonstrate how the different classification approaches perform. Unlike the images of 15 scene categories, the keyframes of LSCOM may contain several overlapping high level concepts. For example, in one keyframe, you probably can see crowd, buildings, cars or roads. Therefore, each keyframe may be classified into multiple categories. We use binary SVM as the classifier (The keyframes from one category are positive, and the rest are treated as negative). The average precision (AP) is adopted as the performance measure. Assume that D retrieved keyframes are ranked, and R of them are relevant ($R < D$), then we can define the AP as follows,

$$AP = \frac{1}{R} \sum_{j=1}^{D} \frac{R_j}{j} * I_j, \tag{10}$$

where $I_j = 1$ if the jth shot is relevant, otherwise 0. R_j is the number of relevant keyframes in the top j retrieved keyframes.

To form the visterm codebook, we randomly selected 50 keyframes from each of the 28 categories and 500 keyframes from categories other than the 28 categories. And finally a $N_v = 3,000$ codebook generated. Further, the "intermediate concepts" using MMI co-clustering and pLSA are generated from the $N_v = 3,000$ codebook. We tried different values of N_c, and chose the value of N_c which gave us the best results. In the SVM learning/classification phase, we randomly divided the dataset into three parts: one half for training, 1/4 for validation and 1/4 for testing. Fig 4 shows the AP of each category. Only for 3 categories, pLSA (pLSA-BOC) performs better than the MMI co-clustering (CC-BOC). Compared to the BOV with reduced dimension ($N_v = 250$), the CC-BOC always performs much better. Besides, for most cases, the CC-BOC can achieve competitive results compared to original BOV ($N_v = 3,000$). However, the gain of CC-BOC is

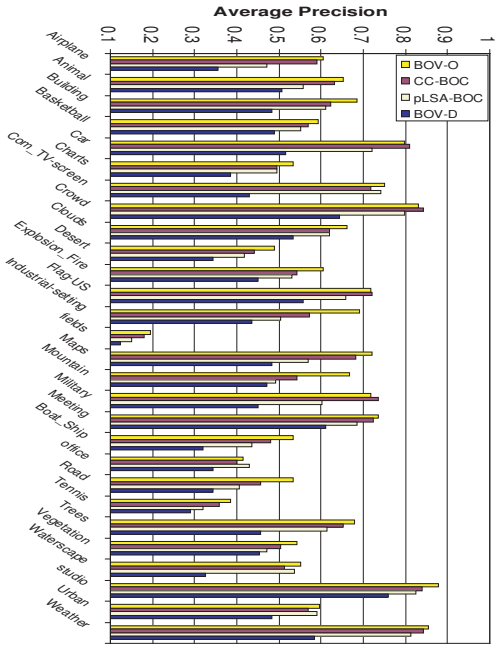

Figure 4. The AP (Average Precision) for the 28 categories. BOV-O and BOV-D represents the BOV model with $N_v = 3,000$ and $N_v = 250$ respectively. CC-BOC and pLSA-BOC denotes the BOC model created by co-clustering and pLSA.

	BOV-O	CC-BOC	BOV-D	pLSA-BOC
MAP	61.91%	59.48%	45.07%	55.77%

Table 6. The MAP (Mean Average Precision) for the 28 LSCOM categories achieved by different approaches. BOV-O and BOV-D represent the BOV models with $N_v = 3,000$ and $N_v = 250$ respectively. CC-BOC and pLSA-BOC denotes the BOC model created by co-clustering and pLSA respectively.

computational efficiency with lower dimension when performing SVM learning and classification on a large dataset. (e.g. it takes about 23 hours to learn and test the 28 categories for BOV with $N_v = 3,000$, while it only takes about 6 hours for BOC with $N_c = 250$ on a 2.99GHz machine.) The advantage of MMI co-clustering can be clearly noticed in table 6 which demonstrates the Mean Average Precision (MAP) of the 28 LSCOM categories using different approaches. Compared to BOV-D, the benefit of MMI co-clustering is about 14.4% in terms of MAP, which further verifies that MMI co-clustering can get more meaningful clusters. Even compared to pLSA, MMI co-clustering performs about 4% better.

4. Conclusion

In this paper, we propose a novel approach for scene modelling. The proposed method first extracts *intermediate concepts* from *visterms* by using MMI co-clustering. Unlike k-means clustering, MMI co-clustering can preserve the mutual information of *visterms* and images when clustering. Therefore, the more compact image representation can significantly improve the performance of classification. Besides, in order to capture the spatial information of the *intermediate concepts*, the framework uses two spatial models SPM and SCC. Experiment results show that both of models can improve the classification accuracy significantly.

5. Acknowledgments

This research was funded by the US Government VACE program.

References

[1] D. Blei, A. Ng and M. Jordan. “Latent Dirichlet Allocation”. Journal of Machine Learning Research, 3:993-1022, 2003.

[2] A. Bosch, A. Zisserman and X. Munoz. “Scene Classification via pLSA”, ECCV 2006.

[3] G. Csurka, C. Bray, C. Dance, and L. Fan. “Visual Categorization with Bags of Keypoints”, ECCV 2004.

[4] I. S. Dhillon, S. Mallela and D. S. Modha. “Information-Theoretic Co-clustering”, ACM SIGKDD 2003.

[5] L. Fei-Fei and P. Perona. “A Bayesian Hierarchical Model for Learning Natural Scene Categories”. CVPR 2005.

[6] S. Gordon, H. Greenspan and J. Goldberger. “Applying the Information Bottleneck Principle to Unsupervised Clustering of Discrete and Continues Image Representation”, ICCV, 2003.

[7] T. Hofmann. “Unsupervised Learning by Probabilistic Latent Semantic Analysis”. Machine Learning, 42, 177-196, 2001.

[8] W. H. Hsu, L. S. Kennedy and S. Chang. “Video Search Reranking via Information Bottleneck Principle”. ACM MM 2006.

[9] D. G. Lowe. “Distinctive Image Features from scale-invariant keypoints”. IJCV, 60(2):91-110,2004.

[10] F. Jurie and B. Triggs. “Creating Efficient Codebooks for Visual Recognition”, ICCV, 2005.

[11] A. Kaban and M. A. Girolami. “Fast Extraction of Semantic Features from a Latent Semantic Indexed Text Corpus”, Neural Processing Letters 15: 31-43, 2002.

[12] S. Lazebnik, C. Schmid and J. Ponce. “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories”, CVPR, 2006.

[13] E. Nowak, B. Triggs and F. Jurie. “Sampling Strategies for Bag-of-Features Image Classification”, ECCV, 2006.

[14] A. Oliva and A. Torralba. “Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope”. ICCV, 2001.

[15] P. Quelhas, F. Monay, J.-M Odobez, D. Gatica-Perez, T. Tuytelaars and L. Van Gool. “Modeling Scenes with Local Descriptors and Latent Aspects”. ICCV, 2005.

[16] A.W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. “Content-based Image Retrieval at the End of the Early Years”. PAMI, 22(12):1349-1380,2000.

[17] S. Savarese, J. Winn and A. Criminisi. “Discriminative Object Class Models of Appearance and Shape by Correlatons”, CVPR 2006.

[18] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman and W. T. Freeman. “Discovering Objects and Their Location in Images ”. ICCV, 2005.

[19] N. Slonim and N. Tishby. “Document clustering using word clusters via the information bottleneck method”,ACM SIGIR 2000.

[20] A. Vailaya, A. T. Figueiredo, A. K. Jain and H.J. Zhang. “Image Classification for Content-Based Indexing”. IEEE Transantions on Image Processing, Vol. 10, No. 1, January 2001.

[21] J. Vogel and B. Schiele. “Natural Scene Retrieval Based on a Semantic Modeling Step”. CIVR 2004.

[22] M. Varma and A. Zisserman. “A statistical approach to texture classification from single images”. IJCV, 62:61-81, Apr. 2005.

[23] J. Winn, A. Criminisi and T. Minka. “Object Categorization by Learned Universal Visual Dictionary”, ICCV 2005.