Speeded-up and Compact Visual Codebook for Object Recognition

B. Mayurathan *barathy@jfn.ac.lk*
Postgraduate Institute of Science,
University of Peradeniya, Sri Lanka.

A. Ramanan, S. Mahesan *a.ramanan, mahesans@jfn.ac.lk*
Department of Computer Science,
University of Jaffna, Sri Lanka.

U.A.J. Pinidiyaarachchi *ajp@pdn.ac.lk*
Department of Statistics and Computer Science,
University of Peradeniya, Sri Lanka.

Abstract

The well known framework in the object recognition literature uses local information extracted at several patches in images which are then clustered by a suitable clustering technique. A visual codebook maps the patch-based descriptors into a fixed-length vector in histogram space to which standard classifiers can be directly applied. Thus, the construction of a codebook is an important step which is usually done by cluster analysis. However, it is still difficult to construct a compact codebook with reduced computational cost. This paper evaluates the effectiveness and generalisation performance of the Resource-Allocating Codebook (RAC) approach that overcomes the problem of constructing fixed size codebooks that can be used at any time in the learning process and the learning patterns do not have to be repeated. It either allocates a new codeword based on the novelty of a newly seen pattern, or adapts the codebook to fit that observation. Furthermore, we improve RAC to yield codebooks that are more compact. We compare and contrast the recognition performance of RAC evaluated with two distinctive feature descriptors: SIFT and SURF and two clustering techniques: K-means and Fast Reciprocal Nearest Neighbours (fast-RNN) algorithms. SVM is used in classifying the image signatures. The entire visual object recognition pipeline has been tested on three benchmark datasets: PASCAL visual object classes challenge 2007, UIUC texture, and MPEG-7 Part-B silhouette image datasets. Experimental results show that RAC is suitable for constructing codebooks due to its wider span of the feature space. Moreover, RAC takes only one-pass through the entire data that slightly outperforms traditional approaches at drastically reduced computing times. The modified RAC performs slightly better than RAC and gives more compact codebook. Future research should focus on designing more discriminative and compact codebooks such as RAC rather than focusing on methods tuned to achieve high performance in classification.

Keywords: Object Recognition, Codebook, K-means, RAC, fast-RNN, SIFT, SURF

1. INTRODUCTION

Object recognition is one of the major challenges in computer vision. Researchers in computer vision have been trying to understand this for many years. Individuals can look around them and recognise familiar objects without much trouble. The ability to generalise from examples and categorise objects, events, scenes, and places is one of the core capabilities of the human visual system. However, this is not an easy task for computers. Local features are used in many computer vision tasks including visual object categorisation, content-based image retrieval, and object recognition. Local features can be thought of as patterns in images that differ from the immediate neighbourhood. Such a pattern can be a corner, blob or a region. The term interest points (or keypoints) usually refer to the set of points that are used to describe these patterns. Local feature detectors are used to find areas of interest in the images.

In the state-of-the-art visual object recognition systems, the visual codebook model has shown excellent categorisation performance in large evaluations such as the PASCAL Visual Object Classes (VOC) Challenges [8] and Caltech object categories [11]. Desirable properties of a visual codebook are compactness, low computational complexity, and high accuracy of subsequent categorisation. Discriminative power of a visual codebook determines the quality of the codebook model, whereas the size of a codebook controls the complexity of the model. Thus, the construction of a codebook plays a central role that affects the model complexity. In general, there are two types of codebook that are widely used in the literature: global and category-specific codebooks. A global codebook may not be sufficient in its discriminative power but it is category-independent, whereas a category-specific codebook may be too sensitive to noise. The codebook itself is constructed by clustering a large number of local feature descriptors extracted from training data. Based on the choice of a clustering algorithm, one might obtain different clustering solutions, some of which might be more suitable than others for object class recognition.

The popular approach to constructing a visual codebook is usually undertaken by applying the traditional *K*-means method. However, clustering is a process that retains regions of high density in a distribution and it follows that the resulting codebook need not have discriminant properties. This is also recognised as a computational bottleneck of such systems. The resource-allocating codebook (RAC) approach [26] that we compare in this paper slightly outperforms more traditional approaches due to its tendency to spread out the cluster centres over a broader range of the feature space thereby including rare local features in the codebook than density-preserving clustering-based codebooks such as K-means and fast-RNN [20].

The objective of this paper is to study the performance of discriminative clustering techniques for object class recognition. Here a scale-invariant feature descriptors, SIFT and e-SURF have been included in order to study the performance of recognising objects. Consequently *K*-means, fast-RNN and RAC methods are used to cluster the extracted descriptors and these clustering techniques performances are compared.

Following the introductory section, the rest of this paper is organised as follows. In section 2, we summarise the background information that are closely related to this paper. This includes visual descriptors that are widely used in codebook model-based object recognition and various clustering techniques that have been used in constructing a codebook. Section 3 provides a summary of previous work on object recognition that has used a codebook model-based approach. Section 4 provides the experimental setup and testing results of our work. In section 5, we discuss the extension of RAC in constructing more compact codebooks for object recognition. Finally, section 6 concludes our work.

2. BACKGROUND

Several combinations of image patch descriptors, different features, matching strategies, various clustering methods and classification techniques have been proposed for visual object recognition. Assessing the overall performance of the individual components in such systems is difficult, since the computational requirements and the fine tuning of the different parts become crucial. The well-known framework in the literature uses the SIFT descriptors to describe the patches and cluster them using the standard K-means algorithm, in order to encode the images as a histogram of visual codewords. This section briefly describes several descriptors, clustering techniques and the well known classification method that were used in comparing different codebook models.

2.1 Local Invariant Features

Usually images are composed of different sets of colours, a mosaic of different texture regions, and different local features. Most previous studies have focused on using global visual features such as edge orientation, colour histogram and frequency distribution. Recent studies use local features that are more robust to occlusions and spatial variations. This new way of looking at local features has opened up a whole new range of applications and has brought us a step closer to cognitive level image understanding. Even though many different methods for detecting and describing local image regions have been developed, the simplest descriptor is a vector of image pixels. In this subsection we summarise the well known patch-based scale-invariant feature transform (SIFT) descriptors [21] and its follow up technique, the Speeded-Up Robust Features (SURF) descriptors [2].

2.1.1 Scale-Invariant Feature Transform (SIFT)

SIFT is a method to extract distinctive features from gray-value images, by filtering images at multiple scales and patches of interest that have sharp changes in local image intensities. The SIFT algorithm consists of four major stages: Scale-space extrema detection, keypoint localisation, orientation assignment, and representation of a keypoint descriptor. The features are located at maxima and minima of a difference of Gaussian (DoG) functions applied in scale space. Next, the descriptors are computed based on eight orientation histograms at a 4×4 sub region around the interest point, resulting in a 128 dimensional vector. In PCA-SIFT [15], the principal component analysis (PCA) is used instead of weighted histograms at the final stage of the SIFT. The dimensionality of the feature space is reduced from 128 to 20 which require less storage and increases speed in matching images. Although the size of the feature vector is significantly smaller than the standard SIFT feature vector, it has been reported that PCA-SIFT is less distinctive than SIFT [13].

2.1.2 Speeded-Up Robust Feature (SURF)

SURF is partly inspired by SIFT that makes use of integral images. The scale space is analysed by up-scaling the integral image-based filter sizes in combination with a fast Hessian matrix-based approach. The detection of interest points is selected by relying on the determinant of the Hessian matrix where the determinant is maximum. Next, the descriptors are computed based on orientation using 2D Haar wavelet responses calculated in a 4×4 sub region around each interest point, resulting in a 32 dimensional vector. When information about the polarity of the intensity changes is considered, this in turn results in a 64 dimensional vector. The extended version of SURF has the same dimension as SIFT. SURF features can be extracted faster than SIFT using the gain of integral images and yields a lower dimensional feature descriptor (i.e. 64 dimensions) resulting in faster matching and less storage space but it is not stable to rotation and illumination changes. In [13], e-SURF (i.e. 128 dimensions) is proved to have better performance than SURF.

2.2 Codebook Construction

A simple nearest neighbour design for patch-based visual object recognition is a possible way forward, but is computationally not feasible for large scale data. Hence, a way to cope with the enormous amount of the patch-based descriptors and their higher dimensionality is to cluster them by using an appropriate clustering method that captures the span of the feature space. Instead of the features themselves, the cluster centroids or representative points are used for the different cluster members. Interest points are detected in training images and a visual codebook is constructed by a vector quantization technique that groups similar features together. Each group is represented by the learnt cluster centres referred to as 'codewords'. The size of the codebook is the number of clusters obtained from the clustering technique. Each interest point of an image in the dataset is then quantized to its closest codeword in the codebook, such that it maps the entire patches of an image in to a fixed-length feature vector of frequency histograms, i.e. the visual codebook model treats an image as a distribution of local features. In this subsection we describe the traditional K-means method and two other techniques known as RAC and fast-RNN that mainly focuses in constructing compact codebooks.

2.2.1 *K*-means

K-means is one of the simplest unsupervised learning algorithm that solves the well known clustering problem. Given a matrix $\mathrm{X} \in \mathbb{R}^{\mathrm{N}\times\mathrm{d}}$ (representing *N* points described with respect to *d* features, then K-means clustering aims to partition the *N* points into K disjoint sets or clusters by minimizing an objective function, which is the squared error function, that minimizes the within-group sum of squared errors. K-means is a Gaussian mixture model with isotropic covariance matrix the algorithm is an expectation-maximization (EM) algorithm for maximum likelihood estimation.

There are several known difficulties with the use of K-means clustering, including the choice of a suitable value for K, and the computational cost of clustering when the dataset is large. It is also significantly sensitive to the initial randomly selected cluster centres. The time complexity of the traditional K-means method is $\mathrm{O(NdKm)}$, where the symbols in parentheses represent number of data, dimensionality of features, the number of desired clusters and the number of iterations of the EM algorithm.

2.2.2 fast-RNN

In [20] the authors have presented a novel approach for accelerating the popular Reciprocal Nearest Neighbours (RNN) clustering algorithm and named as fast-RNN. A novel dynamic slicing strategy is used to speed up the nearest neighbour chain construction. When building nearest neighbour (NN) chains, it finds all the points that lie within a slice of the d-dimensional space of width 2ε centred at query point. To determine the nearest neighbour of $\mathrm{x_i}$ in S, i.e. $\mathrm{x_j} = \mathrm{NN(x_i)}$ where S is a set of N points $\mathrm{S} = \{\mathrm{x_1}, \mathrm{x_2}, \ldots.\mathrm{x_N}\}$, it builds the first slice of width 2ε centred at $\mathrm{x_i}$. Then, it performs a search for the NN of $\mathrm{x_i}$ considering only the points inside this slice. Once $\mathrm{x_j}$ is identified, it searches for its NN via slicing again, and so on.

Thereafter, agglomerative clustering builds the codebook by initially assigning each data point to its own cluster after that repeatedly selecting and merging pairs of clusters. Thus, it builds a hierarchical tree merging from the bottom (leaves) towards the top (root). The authors in [17] have improved the agglomerative clustering method based on the construction of RNN pairs.

2.2.3 Resource-Allocating Codebook

In [26], a Resource-Allocating Codebook (RAC) has been proposed for constructing a discriminate codebook. RAC is a much simplified algorithm that constructs a codebook in a one-pass process which simultaneously achieves increased discrimination and a drastic reduction in the computational needs. It is initialised by a random seed point selected from the set of visual descriptors. When a subsequent data item is processed, its minimum distance to all entries in the current codebook is computed using an appropriate distance metric. If this distance is smaller than a predefined threshold r (radius of the hypersphere) the current codebook is retained and no action is taken. If the threshold is exceeded by the smallest distance to codewords (i.e. it is a group represented by the learnt cluster centres), a new entry in the codebook is created by including the current data item as the additional entry. This process is continued until all data items are seen only once. The pseudocode of this approach is given in Algorithm 1. The RAC partitions the feature space into a set of overlapping hyperspheres when the distance metric used is the Euclidean norm.

Algorithm 1: Resource-Allocating Codebook

```
Input: Visual descriptors (D) and radius (r) of the hyperspheres.
Output: Centres of the hyperspheres (C)
Step 1: $C_1 \leftarrow D_1$
        $i \leftarrow 2$   // to the next descriptor
        $j \leftarrow 1$   // size of the present C
Step 2: Repeat Steps 3 to 4 while $i \leq size(\mathbf{D})$
Step 3: If $\min\|D_i - C_j\|^2 \geq r^2 \quad \forall_j$
        then create a new hypersphere of r such that
            $C_j \leftarrow D_i$
            $j \leftarrow j + 1$
        endif
Step 4: $i \leftarrow i + 1$
Step 5: return C
```

2.3 Classification Using Support Vector Machine (SVM)

SVM is a supervised learning technique based on a statistical learning theory that can be used for pattern classification. In general SVMs outperform other classifiers in their generalisation performance [3]. A linear SVM finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. SVMs were originally developed for solving binary classification problems [5] and then binary SVMs have also been extended to solve the problem of multi-class pattern classification. There are four standard techniques frequently employed by SVMs to tackle multi-class problems, namely One-Versus-One (OVO) [7], One-Versus-All (OVA) [29], Directed Acyclic Graph (DAG) [25], and Unbalanced Decision Tree (UDT) [28].

OVO method is implemented using a “Max-Wins” voting strategy. This method constructs one binary classifier for every pair of distinct classes and in total it constructs N(N-1)/2 binary classifiers, where N is the number of classes. The binary classifier C_{ij} is trained with examples from the i^{th} class and the j^{th} class only. The max-wins strategy then assigns a test data X to the class receiving the highest voting score.

OVA method is implemented using a “Winner-Takes-All” strategy. It constructs N binary classifier models. The i^{th} binary classifier is trained with all the examples in the i^{th} class with positive labels, and the examples from all other classes with negative labels. For a test example X, the winner-takes-all strategy assigns it to the class with the highest classification boundary function value.

DAG-SVMs are implemented using a “Leave-One-Out” strategy. The training phase of the DAG is the same as the OVO method, solving N(N-1)/2 binary classifiers. In the testing phase it uses a rooted binary directed acyclic graph. Each node is a classifier C_{ij} from OVO. A test example X is evaluated at the root node and then it moves either to the left or the right depending on the output value.

UDT-SVMs are implemented using a “knock-out” strategy with at most $(N - 1)$ classifiers to make a decision on any input pattern. Each decision node of UDT is an OVA-based optimal classification model. Starting at the root node, one selected class is evaluated against the rest by the optimal model. Then the UDT proceeds to the next level by eliminating the selected class from the previous level of the decision tree. UDT terminates when it returns an output pattern at a level of the decision node.

3. PREVIOUS WORK

In [6], the authors used the Harris affine region detector to identify the interest points in the images which are then described by SIFT descriptors. A visual codebook was constructed by clustering the extracted features using K-means method. Images are then described by histograms over the learnt codebook. K-means were repeated several times over a selected size of K and different sets of initial cluster centres. The reported results were the clusters that gave them the lowest empirical risk in classification. The size of the codebook used in reporting the results is 1000. The authors compared Naive Bayes and SVM classifiers in the learning task and found that the OVA SVM with linear kernel gives a significantly (i.e. 13%) better performance. The proposed framework was mainly evaluated on their 'in-house' database that is currently known as 'Xerox7' image set containing 1,776 images in seven object categories. The overall error rate of the classification is 15% using SVMs. It has been reported that RAC approach in [26] performs slightly better than the method in [6] but was achieved in a tiny fraction of computation time.

In [14], the authors proposed a mean-shift based clustering approach to construct codebooks in an under sampling framework. The authors sub-sample patches randomly from the feature set and allocate a new cluster centroid for a fixed-radius hypersphere by running a mean-shift estimator [4] on the subset. The mean-shift procedure is achieved by successively computing the mean-shift vector of the sample keypoints and translating a Gaussian kernel on them. In the next stage, visual descriptors that fall within the cluster are filtered out. This process is continued by monitoring the informativeness of the clusters or until a desired number of clusters is achieved. The features used in their experiments are the gray level patches sampled densely from multi-scale pyramids with ten layers. The size of the codebook was 2,500. The proposed method was evaluated on three datasets: Side views of cars from [28], Xerox7 image dataset [6] and the ETH-80 dataset [17]. Naive Bayes and linear SVM classifiers were compared in all their experiments. The authors' mean-shift based clustering method is computationally intensive in determining the cluster centroid by mean-shift iterations at each of the sub samples. The convergence of such a recursive mean-shift procedure greatly depends on the nearest stationary point of the underlying density function and its utility in detecting the modes of the density. In contrast, the RAC approach pursued in [26] has a single threshold that takes only one-pass through the entire data, making it computationally efficient.

In [34], the authors optimized codebooks by hierarchically merging visual words in a pair-wise manner using the information bottleneck principle [1] from an initially constructed large codebook. Training images were convolved with different filter-banks made of Gaussians and Gabor kernels. The resulting filter responses were clustered by the K-means method with a large value of K in the order of thousands. Mahalanobis distance between features is used during the clustering step. The learnt cluster centres and their associated covariances define a universal visual codebook. Classification results were obtained on photographs acquired by the authors, images from the web and a subset of 587 images in total that were selected from the PASCAL VOC challenge 2005 dataset. The initial codebook construction of this method which is based on hierarchically merging visual words in a pair-wise manner is extremely faster than K-means clustering on large number of features. However, if two distinct visual words are initially grouped in the same cluster, they cannot be separated later. Also the vocabulary is tailored according to the categories under consideration, but it would require fully retraining the framework on the arrival of new object categories.

In [22], local features are found by extracting edges with a multi-scale Canny edge detector with Laplacian-based automatic scale selection. For every feature, a geometry term gets determined, coding the distance and relative angle of the object centre to the interest point, according to the dominant gradient orientation and the scale of the interest point. These regions are then described with SIFT features that are reduced to 40-dimension via PCA. The visual codebook is constructed by means of a hierarchical K-means clustering. Given a test image, the features were extracted and a tree structure is built using the hierarchical K-means clustering method in order to compare with the learnt model tree. Classification is done in a Bayesian manner computing the likelihood ratio. Experiments were performed on a five class problem taken from the PASCAL VOC 2005 image dataset.

In [33], a novel method is proposed for constructing a compact visual codebook using a sparse reconstruction technique. Initially a large codebook is generated by K-means method. Then it is reformulated in a sparse manner, and weight of each word in the old visual codebook is learnt from the sparse representation of the entire action features. Finally, a new visual codebook is generated through six different algorithms. These algorithms are mainly based on L_0 and L_1 distances. The authors approach has been tested on the Weizmann action database [10]. As a result, the obtained codebook which is half the size of the old visual codebook has the same performance as the one built by K-means.

In [24], images are characterised by using a set of category-specific histograms generated one per object category. Each histogram describes whether the content can be best modelled by a universal codebook or by its corresponding category-specific codebook. Category-specific codebooks are obtained by adapting the universal codebook using the class training data and a form of Bayesian adaptation based on the maximum a posteriori criterion. The maximum number of Gaussians in the universal codebook was set to 2048. An image is then characterised by a set of histograms called bipartite as they can be split into two equal parts. Each part describes how well one codebook accounts for an image compared to the other codebook. Local patches were described by SIFT and colour features. PCA was applied to reduce the dimensionality of SIFT from 128 to 50, and the RGB colour channels from 96 to 50. Evaluations were performed on their own in-house database containing 19 classes of object categories and scenes, and the PASCAL VOC 2006 dataset. Classification was performed using linear SVMs and a logistic regression with a Laplacian prior. However, if two visual object classes are visually close, there is no guarantee that a distinctive visual word will be obtained. The process that generates bipartite histograms is also computationally expensive.

4. EMPIRICAL EVALUATION

4.1 Datasets

We evaluate the combination of descriptors and codebook construction techniques on three benchmark datasets: PASCAL visual object classes challenge 2007 [8], UIUC texture [16], and MPEG-7 Part-B silhouette image [12] datasets.

4.1.1 PASCAL VOC Challenge 2007

The dataset includes data with all possible viewing conditions such as different viewpoints, scales, illumination conditions, occluded/truncated and poor quality. The goal of this challenge is to recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects).

There are twenty object classes: person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and TV/monitor. The database contains a total of 9,963 annotated images. The dataset has been split nearly into 50 percent for training/validation and 50 percent for testing dataset. Figure1 shows the object categories that are in the PASCAL VOC 2007 dataset.

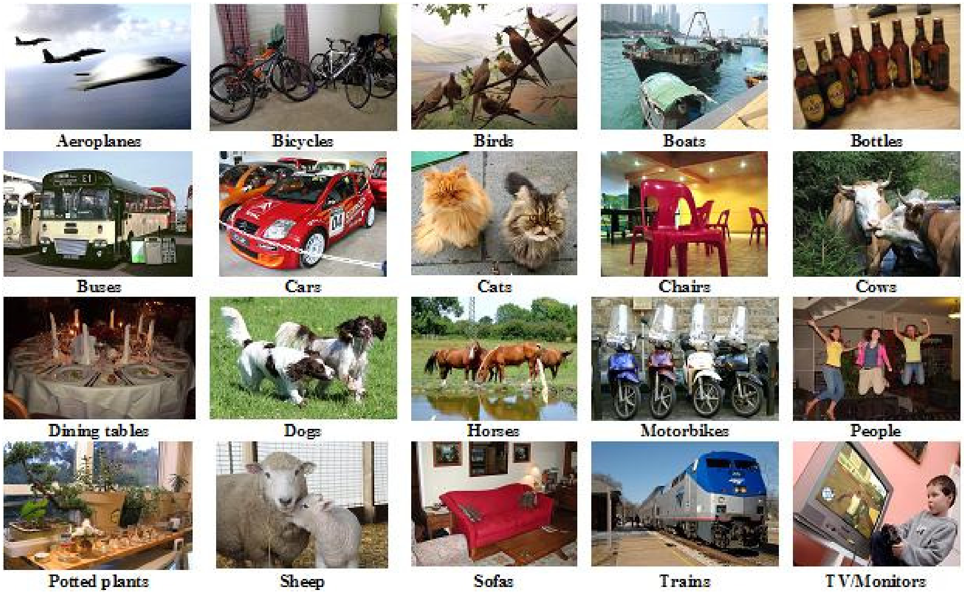


FIGURE 1: One Image from Each of the Object Categories In PASCAL VOC Challenge 2007 Image Dataset.

4.1.2 UIUC Texture Dataset

The dataset contains 25 texture classes with 40 images per class. Each of the images is of size 640×480 pixels. This dataset has surfaces whose texture is mainly due to albedo variations (e.g. wood and marble), 3D shape (e.g. gravel and fur), as well as a mixture of both (e.g. carpet and brick). It also has significant viewpoint changes, uncontrolled illumination, arbitrary rotations, and scale differences within each class. Figure 2 shows some of the example images of the UIUCTex dataset.

4.1.3 Silhouette Images

The MPEG-7 Part B silhouette database is a popular database for shape matching evaluation consisting of 70 shape categories, where each category is represented by 20 different images with high intra-class variability. The shapes are defined by a binary mask outlining the objects. Figure 3 shows some example images of the dataset.

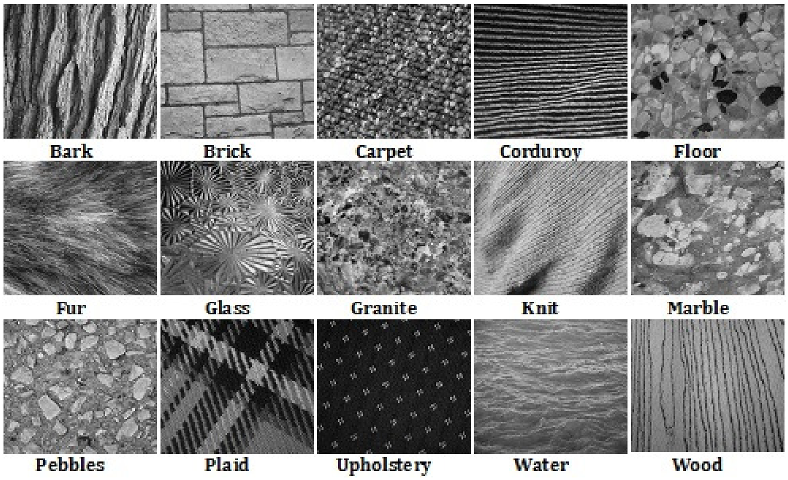


FIGURE 2: Example Images of the UIUCTex Dataset.

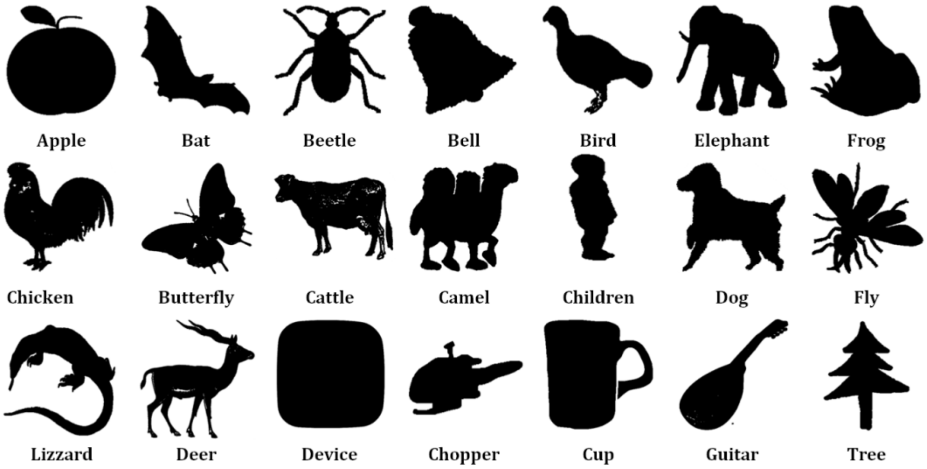


FIGURE 3: Example Images of the MPEG-7 Silhouette Dataset.

4.2 Vocabulary Construction

Popular approaches in the literature of codebook models [6, 9, 19, 31] have used K-means method with K=1000 to construct a codebook which shown better performance. For this reason when comparing the RAC and fast-RNN with K-means, we maintain the codebook size to be 1000. Moreover, K-means method was run three times with the same number of desired representative vectors and different sets of initial cluster centres.

The hyperparameter *r* of RAC is set to 0.8 in PASCAL VOC 2007 and 0.89 in silhouette and UIUCTex datasets. The choice of the radius *r* has the same set of difficulties associated with the choice of K in K-means. In [26], the approach to setting *r* is to take a small sample of the data, compute all pairwise distances between these samples and set the threshold, so that an approximate target codebook size is achieved.

When building NN chains in fast-RNN algorithm, we follow the authors [20] experimental setup in finding all the points that lie within a slice of the d-dimensional space of width 2ε centred at a query point instead of building a hypercube. The ε is set to 0.05.

4.3 Classification

In classification, we have used the OVA-based linear SVMs. The regularization parameter *C* was tuned with a range of values $[2^{-2}, 2^{-1}, \ldots. 2^{11}, 2^{12}]$.

4.4 Evaluation Criterion

Average precision is used as performance evaluator for the binary object recognition tasks which has been widely used in recent PASCAL VOC challenges. Average precision is a single-valued measure that is proportional to the area under a precision-recall curve.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{True positive rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{False positive rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

where TP, FP, TN and FN are true positive, false positive, true negative, and false negative respectively.

Receiver operating characteristics (ROC) curve shows how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. For multi-class classification, we report the classification rate as follows:

$$\text{rate} = \frac{\text{Number of correctly classified images}}{\text{Total number of testing images}} \times 100\%$$

4.5 Testing Results

Experiments in this work were mainly carried out to compare and contrast the RAC technique with traditional K-means method and the speeded-up RNN clustering algorithm in terms of classification rate, compactness of codebook, and time for constructing codebook. We have tested those algorithms on three benchmark datasets. Furthermore, we improve the standard RAC to yield more compact codebook.

For the PASCAL VOC 2007 dataset, we extracted features within the provided bounding box information from the combination of training and validation (trainval) set. Truncated objects were also included when extracting features. For evaluation purpose we have selected ten binary classes from PASCAL VOC 2007 image set (in Table 1). The classification results are shown (in Tables 1 and 2) as means of average precision and standard deviation.

In the UIUCTex dataset, we used ten-fold cross-validation. The dataset was split into ten partitions each containing four images from each of the 25 classes. Ten rounds of training and testing were performed in which nine partitions were used for training and the remaining partition was used for testing. According to [27], $2500 \times \mathbb{R}^{128}$ SIFT and e-SURF keypoints were randomly selected from each image of the training set and were individually clustered by K-means, RAC and fast-RNN techniques in order to form a locally merged global codebook. The random selection of a subset of SIFT and e-SURF descriptors was due to our previous experience of the prohibitive memory requirement by the traditional K-means in such a large number of keypoints (approx. 5000 keypoints per image). The K-means method in constructing 40 clusters per class was fed with a total of 36×2500× $\mathbb{R}^{128}$ SIFT and e-SURF descriptors. The resulting histogram of an image was of size 1000 in K-means method. The classification results are shown (in Tables 1 and 2) as means of average precision and standard deviation, over the ten runs.

In the Silhouette dataset, we used two-fold cross-validation. The Silhouette dataset was split into two partitions each containing ten images from each of the 25 classes. Two rounds of training and testing are performed in which one partition is used for training and the other was used for testing. We report the mean classification rate in Tables 1 and 2, together with the standard deviation, over the two runs.

All of our experiments were implemented in Matlab and executed on a desktop computer with an Intel Core 2 running at 2.4 GHz and 8GB of RAM.

4.5.1 Classification Rate

Table 1 details the classification rate of three independent runs of the proposed experiment using K-means, RAC and fast-RNN with SIFT descriptors respectively, whereas Table 2 details the rate for using SURF descriptors. Based on our testing results, when using SIFT descriptors, K-means performs better in three binary tasks, whereas RAC performs better in six binary tasks of the PASCAL VOC 2007 imageset. Fast-RNN performs better in only one task which can be observed in Table 1.

Dataset		SIFT		
		K-means	RAC	fast-RNN
PASCAL VOC 2007	Bird vs Aeroplane	81.86 ± 1.27	**82.10 ± 0.80**	75.29 ± 1.24
	Aeroplane vs Horse	**91.41 ± 2.40**	87.82 ± 0.83	88.04 ± 0.09
	Bicycle vs Motorbike	**77.40 ± 0.04**	76.10 ± 0.60	74.53 ± 0.99
	Bus vs Train	**79.80 ± 0.55**	76.00 ± 0.60	74.31 ± 0.69
	Dog vs Cat	68.22 ± 1.51	70.73 ± 0.65	**71.10 ± 1.40**
	Cow vs Sheep	67.77 ± 1.20	**71.10 ± 1.40**	70.89 ± 0.78
	Pottedplant vs Dining table	72.97 ± 0.53	**76.00 ± 1.19**	74.00 ± 1.60
	Bottle vs Potted plant	59.69 ± 2.15	**67.72 ± 0.31**	62.54 ± 1.30
	Boat vs TV/monitor	74.85 ± 1.20	**81.20 ± 1.80**	79.91 ± 1.10
	Aeroplane vs Boat	80.00 ± 0.90	**87.80 ± 0.80**	80.16 ± 0.81
UIUCTex		**98.77 ± 0.97**	98.10 ± 0.99	95.12 ± 0.65
MPEG 7 Part B (Silhouette)		**77.60 ± 2.26**	77.20 ± 3.39	75.19 ± 0.40

TABLE 1: Classification Rate as Mean Average Precision With Standard Deviation When Using SIFT Descriptors.

In the case of SURF descriptors, K-means performs better in eight binary tasks, whereas RAC performs better in three binary tasks of the PASCAL VOC 2007 imageset and fast-RNN performs better in only one task which can be observed in Table 2.

We carried out F-tests (one-way ANOVA) to compare the classification rates of K-means, RAC and fast-RNN techniques when applied on the PASCAL VOC 2007 imageset using SIFT and SURF descriptors. We may conclude that those three techniques are equally comparable in classification rates for the SIFT and SURF descriptors with the p-values 0.71 and 0.94, respectively.

Even though K-means slightly outperforms RAC in classification rate when applied on UIUCTex and silhouette image classification tasks, the negligible increase in performance was achieved at a huge computational time which can be observed in Table 3. Fast-RNN not only shows less classification rate compared to RAC but also higher codebook construction time to RAC. Even though fast-RNN has some demerits to RAC it still constructs more compact codebooks than the other approaches.

Dataset		e-SURF		
		K-means	RAC	fast-RNN
PASCAL VOC 2007	Bird vs Aeroplane	86.22 ± 0.21	**86.48 ± 1.02**	85.36 ± 0.21
	Aeroplane vs Horse	**91.39 ± 0.28**	85.36 ± 0.85	85.58 ± 0.42
	Bicycle vs Motorbike	**81.80 ± 0.22**	79.47 ± 0.32	75.98 ± 0.79
	Bus vs Train	69.74 ± 1.66	**79.61 ± 2.25**	76.13 ± 1.38
	Dog vs Cat	**70.31 ± 0.92**	68.83 ± 0.80	64.70 ± 0.12
	Cow vs Sheep	**67.26 ± 0.97**	65.06 ± 1.65	66.24 ± 1.81
	Pottedplant vs Dining table	**74.06 ± 0.31**	72.18 ± 1.53	70.32 ± 0.41
	Bottle vs Potted plant	60.92 ± 0.67	**61.04 ± 1.65**	60.04 ± 2.41
	Boat vs TV/monitor	56.82 ± 1.51	71.96 ± 3.12	**73.45 ± 1.37**
	Aeroplane vs Boat	**75.27 ± 1.80**	71.20 ± 2.04	69.13 ± 1.30
UIUCTex		**97.88 ± 0.92**	97.05 ± 1.47	94.52 ± 0.03
MPEG 7 Part B (Silhouette)		**75.03 ± 0.05**	74.00 ± 0.56	70.71 ± 0.23

TABLE 2: Classification Rate as Mean Average Precision With Standard Deviation Using SURF Descriptors.

The classification performance for K-means, RAC and fast-RNN clustering techniques with SIFT and e-SURF descriptors is also represented using ROC curves. Figure 4 illustrates some example of the ROC curves which are randomly selected from our results. The ROC curve details how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. In these figures, Fig. 4(a), (d) and (e) lie in the upper-left-hand corner representing a good classification result and SIFT + RAC performs better than other combinations.

In addition to this, we carried out limited experiments to see the influence of the order of presentation of data to RAC. Based on the limited experiments, we found that RAC is slightly sensitive to the order of presentation of data, similar to the random initial selection of cluster centres in the K-means method.

Dataset		SIFT			e-SURF		
		K-means	RAC	fast-RNN	K-means	RAC	fast-RNN
PASCAL VOC 2007	Aeroplane	34712	495	34735	1845	12	1664.8
	Bird	89538	780	74695	6540	12	2117
	Bottle	17824	66	14097	536	2	124
	Boat	12652	123	11202	2670	6	709
	Bus	27657	489	27657	5927	12	2092
	Bicycle	34906	270	31140	10269	14	2951
	Cat	248810	334	173563	3809	14	2588
	Cow	27999	116	22440	1800	5	583
	Dining table	35544	252	33263	5338	10	1490
	Dog	271695	452	265264	7380	21	5027
	Horse	86639	793	79988	5587	18	3622
	Motorbike	80002	619	83496	10172	23	6236
	Potted plant	32137	195	27274	8398	9	1481
	Sheep	10489	50	9256	900	2	264
	Train	78621	969	75324	16497	22	5629
	TV/Monitor	7608	69	24023	466	2	172
UIUCTex		70585	10050	40100	60141	79	57125
MPEG 7 Part B (Silhouette)		11597	20	10200	6135	4	5135

TABLE 3: Estimated Time for Codebook Construction Using SIFT and SURF Descriptor.

4.5.2 Compactness of Codebook

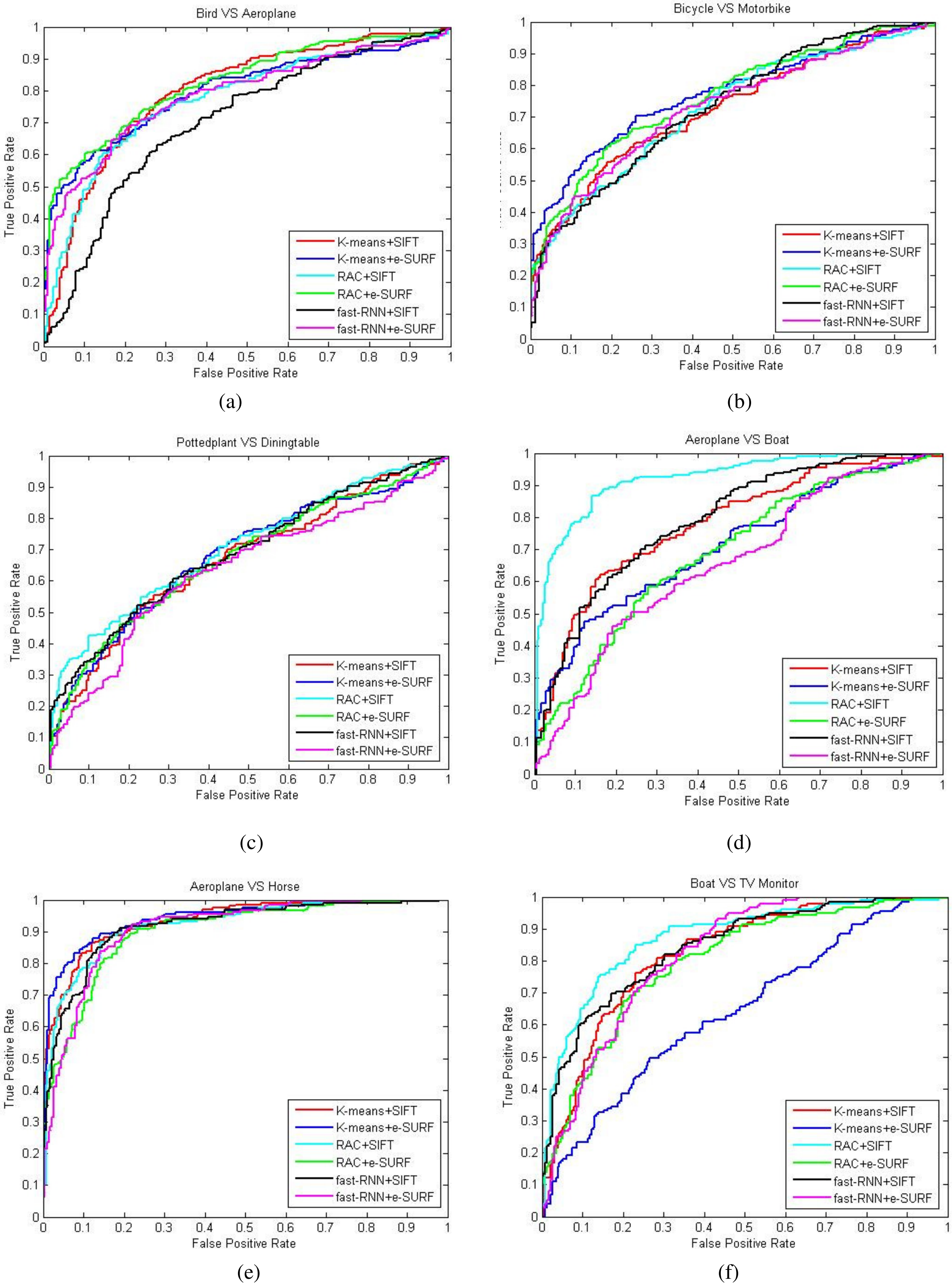
A compact codebook may be achieved directly by reducing the codebook size or by carefully selecting the codebook elements. RAC sizes are slightly greater than others. Table 4 indicates average size of the codebooks by using K-means, RAC and fast-RNN with SIFT and e-SURF descriptors in that order. As shown in Table 4 the size of the codebook obtained by e-SURF with fast-RNN yields codebooks of an average size of 131 while SIFT with fast-RNN produces codebooks of an average of 280.

4.5.3 Time For Constructing Codebook

The construction of a visual codebook is often performed from thousands of images and each image contains hundreds or even one thousands of patch based interest points described in a higher dimensional space, in order to capture sufficient information for efficient classification.

The time complexity of the traditional K-means method is $\mathrm{O(NdKm)}$, whereas for the fast-RNN technique it is $\mathrm{O(N \log N)}$ where, *N* -number of descriptors, *d*-dimensionality of features, *K*-number of desired clusters and *m*-number of iterations of the expectation-maximization (EM) algorithm. The time complexity of RAC depends on the size of the candidate cluster set C, i.e., it compares each newly seen pattern to all existing clusters. Thus, RAC has far lower computational cost than K-means and fast-RNN clustering techniques.

In our experiments codebook construction time is quantified in two cases: SIFT and e-SURF and have been tested with those three algorithms. In all cases, e-SURF is faster than SIFT based on the outcomes of the results reported in Table 4 and as found by [2]. When commenting on the performance of descriptors, we conclude that SIFT performs better than SURF descriptors, which has also been proved by [13].



(a) (b)

(c) (d)

(e) (f)

FIGURE 4: Classification of the Test Set in PASCAL VOC 2007 Represented Using the ROC Curves for Binary Classes.

Since the total number of interest points detected by e-SURF is much less than that of SIFT descriptors as shown in Figure 5, K-means contradicts with fast-RNN in both cases. Based on our computational time to construct a codebook, RAC seems particularly suitable for the codebook construction due to its speed than other methods compared in this paper.

Objects		SIFT			e-SURF		
		K-means	RAC	fast-RNN	K-means	RAC	fast-RNN
PASCAL VOC 2007	Aeroplane	1000	1325	342	1000	526	140
	Bird		1165	291		461	120
	Bottle		945	220		321	97
	Boat		1032	251		383	109
	Bus		1233	286		492	135
	Bicycle		1148	265		423	105
	Cat		1167	260		506	260
	Cow		932	210		369	100
	Dining table		1219	302		452	111
	Dog		1305	268		540	120
	Horse		1297	303		541	119
	Motorbike		1399	316		510	120
	Potted plant		1074	245		376	103
	Sheep		711	174		277	75
	Train		1410	347		531	147
	TV Monitor		943	248		358	129
UIUCTex			1524	375		1491	245
MPEG 7 Part B			1242	340		1370	125

TABLE 4: Average Size of the Codebook Using SIFT and e-SURF Descriptors.

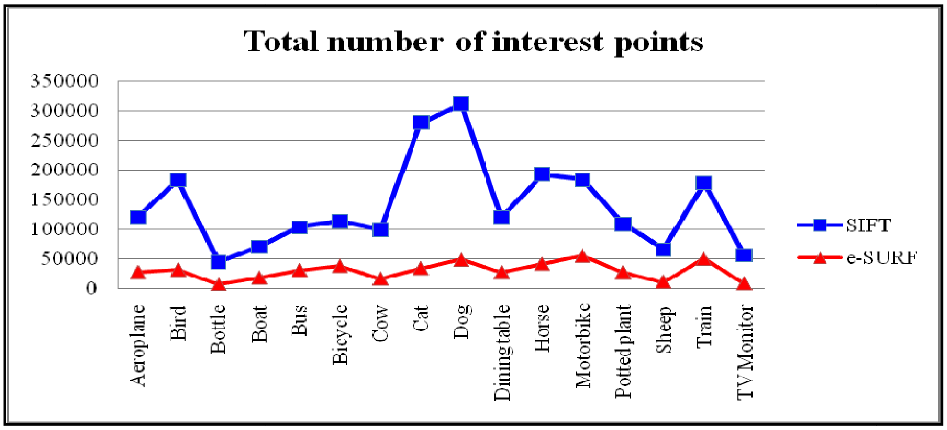


FIGURE 5: Total Number of Interest Points Detected in PASCAL VOC 2007 Dataset.

4.6 Further Improvement

Based on our testing results, we can easily conclude that RAC outperforms to the traditional K-means and fast-RNN but the codebook sizes constructed by RAC are slightly greater than others. To overcome this issue, we have reduced the codebook sizes of RAC by removing hyperspheres that contain less number of members. That is if any visual word in codebook contains very small number of local descriptors (i.e., $\mathrm{n} \leq 10$) that hypersphere is removed from the codebook. We refer this method as compact RAC.

Table 5 details the classification rate of three independent runs of the proposed experiment using Compact-RAC that is compared with the standard RAC. Recognition results are shown as means of average precision with standard deviation.

Dataset		SIFT		e-SURF	
		RAC	Compact RAC	RAC	Compact RAC
PASCAL VOC 2007	Bird vs Aeroplane	**82.10 ± 0.80**	81.12 ± 0.42	86.48 ± 1.02	**87.20 ± 0.50**
	Aeroplane vs Horse	87.82 ± 0.83	**87.00 ± 0.04**	**85.36 ± 0.85**	85.18 ± 1.79
	Bicycle vs Motorbike	76.10 ± 0.60	**77.37 ± 0.95**	**79.47 ± 0.32**	79.27 ± 2.12
	Bus vs Train	76.00 ± 0.60	**78.91 ± 0.16**	79.61 ± 2.25	**79.61 ± 2.36**
	Dog vs Cat	**70.73 ± 0.65**	69.65 ± 1.42	**68.83 ± 0.80**	68.28 ± 2.04
	Cow vs Sheep	71.10 ± 1.40	**71.33 ± 0.72**	**65.06 ± 1.65**	**65.27 ± 0.21**
	Potted plant vs Dining table	**76.00 ± 1.19**	75.62 ± 0.97	72.18 ± 1.53	**71.48 ± 0.99**
	Bottle vs Potted plant	**67.72 ± 0.31**	63.37 ± 2.13	**61.04 ± 1.65**	60.64 ± 0.88
	Boat vs TV/monitor	**81.20 ± 1.80**	76.37 ± 3.70	71.96 ± 3.12	**74.03 ± 2.99**
	Aeroplane vs Boat	**87.80 ± 0.80**	73.19 ± 2.80	71.20 ± 2.04	**71.34 ± 1.70**
UIUCTex		**98.10 ± 0.99**	97.04 ± 0.96	**97.05 ± 1.47**	96.00 ± 2.67
MPEG 7 Part B		**77.20 ± 3.39**	73.80 ± 0.84	**74.00 ± 0.56**	74.00 ± 1.13

TABLE 5: RAC vs Compact-RAC: Recognition Results as Mean Average Precision With Standard Deviation.

Figure 6 shows the average size of the codebook by using Compact-RAC. Based on the results of Table 5 and Figure 6, recognition results using Compact-RAC are reduced or increased by one percentage compared with RAC. But the size of the codebook is significantly compact when using Compact-RAC.

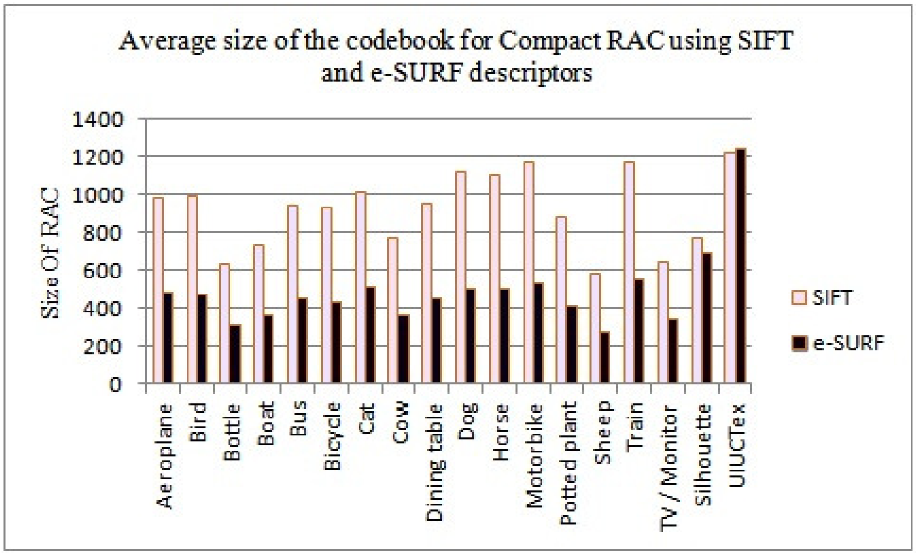


FIGURE 6: Average Size of the Codebook for *Compact*-RAC Using SIFT and e-SURF Descriptors.

5. DISCUSSION AND CONCLUSION

This paper mainly focuses on the evaluation of codebook construction techniques: K-means, fast-RNN and RAC algorithms that are used in patch-based visual object recognition. Our work suggests the need for an online codebook construction technique such as RAC in constructing discriminant and compact codebooks at a drastically reduced time. We also compare the well known patch-based descriptors SIFT and SURF in classification rate. In practice, the construction of a visual codebook is often performed from thousands of images and each image on average contains hundreds or even one thousand of patch-based interest points described in a higher dimensional space of one hundred, in order to capture sufficient information for efficient classification. While clustering algorithms and their performance characteristics have been studied extensively over recent years, a major bottleneck lies in handling the massive scale of the datasets. The Caltech and PASCAL VOC Challenge image datasets are becoming gold standard for measuring recognition performance in recent vision papers but the size of these datasets nearly grows exponentially over the years. The size of the codebooks that have been used in the literature ranges from 102 to 104, resulting in very high-dimensional histograms. A larger size of codebook increases the computational needs in terms of memory usage, storage requirements, and the computational time to construct the codebook and to train a classifier. On the other hand, a smaller size of codebook lacks good representation of true distribution of features. Thus, the choice of the size of a codebook should be balanced between the recognition rate and computational needs.

Based on our testing results the fast-RNN method constructs codebooks that are more compact but shows less classification rate than K-means and RAC algorithms. Even though K-means slightly performs better than RAC, it requires more computational resources such as memory, disk space and huge time in constructing a codebook. In contrast, RAC sequentially processes large number of descriptors in a higher dimensional feature space to constructing compact codebooks for reliable object categorisation performance at drastically reduced computational needs.

SIFT and SURF descriptors are invariant to common image transformations, such as scale changes, image rotation, and small changes in illumination. These descriptors are also invariant to translations as from the use of local features. SURF features can be extracted faster than SIFT using the gain of integral images and yield a lower dimensional feature descriptor resulting in faster matching and less storage space. SIFT descriptors have been found highly distinctive in performance evaluation [23] which has also been proved in our experiments.

The RAC discussed in this paper seems particularly suitable for the codebook construction owing to its simplicity, speed and performance and its one-pass strategy that requires relatively little memory. RAC is also fundamentally different from traditional approaches where it is not the density of detected patches one needs to retain in the codebook but the coverage across the feature space. We have demonstrated RAC with a computationally much simplified algorithm compared to what others have achieved.

6. REFERENCES

[1]. S. Agarwal, A. Awan, and D. Roth, "Learning to Detect Objects in Images via a Sparse, Part-based Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, pp.1475–1490, 2004.

[2]. H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features", In Computer Vision and Image Understanding, Vol. 110, pp. 346–359, 2008.

[3]. C. J. Burgues, "A Tutorial on Support Vector Machines for Pattern Recognition", Knowledge Discovery and Data Mining, Vol. 2, pp. 121–167, 1998.

[4]. D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis", In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, pp. 603–619, 2002.

[5]. N. Cristianini and J. Shawe-Taylor, "An introduction to Support Vector Machines and other Kernel-based Learning Methods", Cambridge University Press, 2000.

[6]. G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual Categorization with Bags of Keypoints", In Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22, 2004.

[7]. R. Debnath, N. Takahide, and H. Takahashi, "A Decision based One-Against-One Method for Multi-class Support Vector Machine", Pattern Analysis Application, Vol. 7, pp. 164–175, 2004.

[8]. M. Everingham, L. Van-Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results".

[9]. L. Fei-Fei, and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories", In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, pp. 524–531, 2005.

[10]. L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes", In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 29, pp. 2247–2253, 2007.

[11]. G. Griffi, A. Holub and P. Perona, "The Caltech-256 Object Category Dataset", Technical Report, California Institute of Technology, 2007.

[12]. L. Jan Latecki, R. Lakamper and U. Eckhardt, "Shape Descriptors for Non-rigid Shapes with a Single Closed Contour", In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 424–429, 2000.

[13]. L. Juan and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF", In International Journal of Image Processing, Vol. 3, pp. 143–152, 2009.

[14]. F. Jurie and B. Triggs, "Creating Efficient Codebooks for Visual Recognition", In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), Vol. 01, pp. 604 – 610, 2005.

[15]. Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pp. 511–517, 2004.

[16]. S. Lazebnik, C. Schmid, and J. Ponce, "A Sparse Texture Representation using Local Affine Regions", In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 27, pp. 1265–1278, 2005.

[17]. B. Leibe and B. Schiele, "Interleaved Object Categorization and Segmentation", In Proceedings of the British Machine Vision Conference (BMVC’03), pp. 759–768, 2003.

[18]. D. Li, L. Yang, X. Hua and H. Zhan, "Large-scale Robust Visual Codebook Construction", ACM International Conference on Multimedia (ACM-MM), pp. 1183–1186, 2010.

[19]. T. Li, T. Mei and I.-S. Kweon, "Learning Optimal Compact Codebook for Efficient Object Categorization", In IEEE workshop on Applications of Computer Vision, pp. 1–6, 2008.

[20]. R. J. Lopez-Sastre, D. Onoro-Rubio, P. Gil-Jimenez, and S. Maldonado-Bascon, "Fast Reciprocal Nearest Neighbours Clustering", Signal Processing, Vol. 92, pp. 270–275, 2012.

[21]. D. Lowe, "Distinctive Image Features from Scale-invariant Keypoints", International Journal of Computer Vision, Vol. 60 (2), pp. 91–110, 2004.

[22]. K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple Object Class Detection with a Generative Model", In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), Vol. 1, pp. 26–36, 2006.

[23]. K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors", In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, pp. 1615–1630, 2005.

[24]. F. Perronnin, "Universal and Adapted Vocabularies for Generic Visual Categorization", In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, pp. 1243–1256, 2008.

[25]. J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large Margin DAGs for Multiclass Classification", In Advances in Neural Information Processing Systems (NIPS’00), Vol. 12, pp. 547–553, 2000.

[26]. A. Ramanan and M. Niranjan, "A One-Pass Resource-Allocating Codebook for Patch-based Visual Object Recognition", In Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP'10), pp. 35 – 40, 2010.

[27]. A. Ramanan, R. Paheerathy, and M. Niranjan, "Speeding Up Multi-Class Texture Classification by One-Pass Vocabulary Design and Decision Tree", In Proceedings of the Sixth IEEE International Conference on Industrial and Information Systems (ICIIS’11), pp. 255-260, 2011.

[28]. A. Ramanan, S. Suppharangsan, and M. Niranjan. "Unbalanced Decision Trees for Multi-class Classification", In Proceedings of the IEEE International Conference on Industrial and Information Systems (ICIIS’07), pp. 291–294, 2007.

[29]. R. Rifkin and A. Klautau, "In Defense of One-vs-All Classification", Journal of Machine Learning Research, Vol. 5, pp. 101–141, 2004.

[30]. E. B. Sudderth, A. Torralba, W. T. Freeman and A. S. Willsky, "Describing Visual Scenes using Transformed Objects and Parts", International Journal of Computer Vision, Vol. 77, pp. 291–330, 2008.

[31]. N. Tishby, F. C. Pereira, and W. Bialek, "The Information Bottleneck Method", In the 37th Annual Allerton Conference on Communication, Control and Computing, pp. 368–377, 1999.

[32]. Q. Wei, X. Zhang, Y. Kong, W. Hu and H. Ling, "Compact Visual Codebook for Action Recognition", In International Conference on Image Processing (ICIP), pp. 3805–3808, 2010.

[33]. J. Winn, A. Criminisi, and T. Minka, “Object Categorization by Learned Universal Visual Dictionary", In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1800–1807, 2005.

[34]. C. Zhang, J. Liu, Y. Ouyang, Q. Tian, H. Lu, S. Ma, "Category Sensitive Codebook Construction for Object Category Recognition", In International Conference on Image Processing (ICIP), pp. 329–332, 2009.

[35]. H. Zhang, A. Berg, M. Maire, and J. Malik. "SVM-KNN Discriminative Nearest Neighbour Classification for Visual Category Recognition”, In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), pp. 2126-2136, 2006.