SCALABLE ACTIVE LEARNING STRATEGY FOR OBJECT CATEGORY RETRIEVAL

David GORISSE[1], *Matthieu CORD*[2], *Frederic PRECIOSO*[1]

[1] ETIS, CNRS/ENSEA/UCP, Univ Cergy-Pontoise, France, gorisse, precioso@ensea.fr
[2] LIP6, UPMC-P6, Paris, France, matthieu.cord@lip6.fr

ABSTRACT

Since the digital revolution, the volume of images to be processed has grown exponentially. Interactive search systems have to deal with these huge databases to remain effective. As the complexity of on-line learning methods is at least linear in the size of the database, scalability is the major problem for these methods. Fast retrieval systems, with index structures for fast navigation, have hence become like a Holy Grail. In this article, we propose a strategy to overcome this scalability limitation. Our technique exploits ultra fast retrieval methods as Locally Sensitive Hashing to speed up active learning system. Experiments on database of 180K images are reported. The results show that our method is 45 times faster than state of the art approaches for similar accuracy.

Index Terms— Image databases, Interactive systems, Learning systems, Image classification

1. INTRODUCTION

In Content-Based Image Retrieval (CBIR) classification framework, retrieving classes of images is usually considered as a two-class problem: the relevant class, the set of images corresponding to the user query concept, and the irrelevant class composed by the remaining of the database. Active learning is a framework of CBIR [1, 2] where a user refines his query by iteratively annotating carefully selected images proposed by the system, in a so-called *relevance feedback loop*. Such selection strategies are particularly relevant in image interactive retrieval. The training set is small since only few annotations should be required from the user [3].

As a consequence, the annotations must provide the best possible classification. This specific process is called *sampling strategy* in [4] and it consists in labeling the most uncertain images. However, the computational complexity is usually at least linear in the number of images in the database for each relevance feedback loop. When the database becomes very large, this scheme becomes intractable and represents a scalability lock for interactive retrieval systems.

Additionally, interactive approaches [5] require to sort the whole database with respect to the classification function in order to focus on the top rank images (which should be most relevant). This *database ranking* process also presents a computational complexity linear in the size of the database and represents another scalability lock of CBIR systems.

Most of existing approaches focus only on addressing the scalability issue of *sampling strategy*. In [6], Crucianu et al. propose a method which associates an M-tree built in the feature space with the kernel of the Support Vector Machine (SVM) in order to quickly retrieve the most uncertain images. In [7], Panda et al. cluster the feature space. They focus on clusters which are close to SVM boundary in order to select the most uncertain data for annotation. This last approach is problematic when considering very large database without control over the number of clusters.

Concerning *database ranking* issue, some approaches based on One-class SVM [8] have been proposed to speed up this process. However, when considering the number of relevance feedback loops needed to identify the target class, 2-class SVM with active learning significantly outperforms One-class SVM. In [7], in order to avoid ranking all the database, Panda et al. build a new index structure KDX designed for quickly retrieving the most relevant data farthest from the SVM boundary.

To the best of our knowledge, only Panda et al. [7] address both scalability issues of *sampling selection* and *database ranking*. In our previous work [9], we proposed a strategy to quickly select, in a large database, relevant images to be annotated using an Euclidean Locally Sensitive Hashing scheme (LSH). Our first results on addressing both scalability locks of CBIR systems for large databases, sample selection and data ranking, were promising.

In this paper, we present a new approach which outperforms the previous one: we build a pool of relevant data on which focusing both sample selection and ranking to make active learning techniques scalable in very large database context; we propose a strategy to quickly update this pool to explore the feature space; we propose a brand new LSH scheme defining new hash functions dedicated to χ^2 distance which proved to often lead to better results for image retrieval task [10]; we test our approach on a database of 180K images and show how our active learning strategy combined with a kernel-based SVM can be powerful to address interactive content-based image retrieval in very large databases.

2. SUB-LINEAR RETRIEVAL SCHEME

In active learning, the user is not interested in the ranking of the whole database $\mathcal{B}$. Indeed, only top rank of the N most relevant images, called TOPN, is useful (usually, N is fixed by the user). Why thus sorting the whole database while the user is only interested in the top of the search? Our idea is to shortcut this whole ranking process by selecting a pool of images, called $\mathcal{S}$, which, thanks to heuristics, are more than likely to be among the TOPN.

2.1. Selection strategy

Let $\{\mathbf{x}_i\}_{1,n}$ be the n image indexes of the database. The set of unlabeled images is denoted by $\mathcal{U} = \{(\mathbf{x}_i, y_i)_{i=1,n} \mid y_i = 0\}$. A training set $\mathcal{A} = \{(\mathbf{x}_i, y_i)_{i=1,n} \mid y_i \neq 0\}$ is then iteratively built, where $y_i = 1$ if the image $\mathbf{x}_i$ is labeled as relevant, $y_i = -1$ if the image $\mathbf{x}_i$ is labeled as irrelevant. The classifier is then trained using these labels at each iteration, and a classification function $f_{\mathcal{A}}(\mathbf{x})$ is determined in order to be able to rank the data. Looking for TOPN images means finding the N highest values for $f_{\mathcal{A}}(\mathbf{x})$. This function may be split as:

$$\begin{aligned} f_{\mathcal{A}}(\mathbf{x}) &= \sum_{p=1}^{|\mathcal{A}^+|} \alpha_p K(\mathbf{x}_p, \mathbf{x}) - \sum_{n=1}^{|\mathcal{A}^-|} \alpha_n K(\mathbf{x}_n, \mathbf{x}) \\ &= f_{\mathcal{A}^+}(\mathbf{x}) - f_{\mathcal{A}^-}(\mathbf{x}) \qquad (1) \end{aligned}$$

where $\mathcal{A}^+$ denotes positive labeled images and $\mathcal{A}^-$, the negative ones. K is a similarity kernel function. Our strategy is to replace the $f_{\mathcal{A}}$ optimization by focusing on $f_{\mathcal{A}^+}$. More precisely, we assume that if an image $\mathbf{x}$ is close to one positive training example $\mathbf{x}_p \in \mathcal{A}^+$, $\mathbf{x}$ has good chances to get a high $f_{\mathcal{A}}(\mathbf{x})$ score. The first step of our process, called *selection*, is to get images from the unlabeled image set $\mathcal{U}$, which are close to one of positive training examples $\mathcal{A}^+$. We carry out this step by using a k-NN search for all data in $\mathcal{A}^+$ in order to fast select a pool of images called $\mathcal{S}'$. Although many of images in $\mathcal{S}'$ are not good, they can be easily filtered. The second step of our strategy, called *pruning*, aims at filtering $\mathcal{S}'$. As already noted, a candidate $\mathbf{x}$ may finally have a poor $f_{\mathcal{A}}(\mathbf{x})$ score. This step consists in computing the exact relevance of the selected images by using $f_{\mathcal{A}}$ on the selected images and keeping only the N most relevant images to build $\mathcal{S}$. As aforementioned, in order to provide an effective scheme, the k-NN search must be very fast. Instead of doing a linear scan, we use an efficient indexing scheme based on LSH, which will be detailed in the next section. As LSH is based on l_1 and l_2 metrics, to be consistent the kernel function has to rely on these distances. However, it was observed that the quality of retrieval achieved using l_2 is not always satisfactory while χ^2 distance proved to often lead to better results for image and video retrieval tasks [11, 10]. In this paper, we design a brand new LSH scheme for the χ^2 distance which allows us to combine this fast k-NN search with our kernel based on χ^2 distance.

2.2. Annotation strategy

The second scalability issue to address is the sampling stage. The aim is to select from the unlabeled dataset $\mathcal{U}$ the best image $\mathbf{x}_{i^\star}$ to label. Once the label is assigned, the relevance function should be improved for the next search run. We consider in this paper the angle diversity strategy [12]:

$$i^\star = \arg\min_{\mathbf{x}_i \in \mathcal{S}} (|f_{\mathcal{A}}(\mathbf{x}_i)| + (\max_{\mathbf{x}_j \in \mathcal{A}} \frac{|K(\mathbf{x}_i, \mathbf{x}_j)|}{\sqrt{K(\mathbf{x}_i, \mathbf{x}_i) K(\mathbf{x}_j, \mathbf{x}_j)}}))$$

This strategy consists in looking in the whole database and to select the most uncertain and the most diversified image. In order to be scalable, we propose to take benefit from our previous process and to only look for $\mathbf{x}_{i^\star}$ in $\mathcal{S}$. Indeed, as long as the user is not satisfied by the results, this means that the relevance function $f_{\mathcal{A}}$ is not accurate enough to retrieve relevant enough images. Then, the pool $\mathcal{S}$ contains several uncertain images which will help to improve the relevant function $f_{\mathcal{A}}$. Moreover, in huge databases, the classification problem considered here is very unbalanced: the size of the relevant image subset is very small in comparison with the size of the irrelevant image subset. It follows that a positively annotated image is most likely to be close to the center of the relevant class than a negatively annotated image. By focusing not too far from positive examples, *i.e.* in $\mathcal{S}$, we then increase our chances to select positive images and then we can hope to rebalance the problem.

2.3. Search algorithm

The scheme of our strategy is summarized on Fig.1 and each block is described as follows: the system is initialized with a *query image*

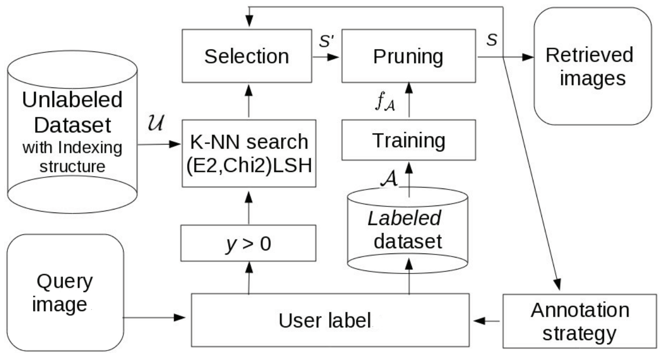

Fig. 1. scheme of fast active learning

provided by the user which is added to the training set $\mathcal{A}$ (*labeled dataset*) to learn a relevant function $f_{\mathcal{A}}$ with a SVM classifier (*training*). At the same time, this image is used to perform a fast *k-NN search* in order to quickly retrieve images of the *unlabeled dataset* $\mathcal{U}$ close to this first positive example. For the initial loop, the *selection* step consists in initializing the pool $\mathcal{S}'$ with the images retrieved by the k-NN search. The pool $\mathcal{S}'$ is then ranked in the *pruning* block using the relevant function $f_{\mathcal{A}}$. The pool $\mathcal{S}$ is built keeping only the N most relevant images from $\mathcal{S}'$. The *pruning* block ensures that the size of $\mathcal{S}$ remains constant by removing the most irrelevant images. The TOPN is shown to the user (*Retrieved images*). The *Annotation strategy* block selects the most uncertain image of $\mathcal{S}$ to be labeled by the user. The system iterates as long as the user (*User label*) is not satisfied by the search.

After each loop, the relevant function $f_{\mathcal{A}}$ is retrained, and thus the pool $\mathcal{S}$ has to be updated. This update of $\mathcal{S}$ is done in the *selection* block. Images of $\mathcal{U}$ close to one of the positive training examples from $\mathcal{A}^+$ should hence be retrieved. However, noticing that $\mathcal{S}$ already contains such images of $\mathcal{U}$ from previous iterations, this step is speeded up by only adding the k-NN of the image if it has been positively labeled ($y > 0$) to previous $\mathcal{S}$ in order to build the new $\mathcal{S}'$. Then, the *pruning* step, filtering the pool $\mathcal{S}'$, provides a new pool $\mathcal{S}$, ranked, which can be presented to the user as intermediate result. Finally, the *Annotation strategy* block selects the most uncertain images of $\mathcal{S}$ to be labeled by the user. The system iterates as long as the user is not satisfied by the search.

3. INDEXING STRUCTURE

3.1. Indexing scheme

The fundamental principle of LSH relies on the construction of a hash table using a hash function instead of sorted data. The hash function is used to map vectors into buckets, such that nearby vectors are much more likely to map into the same bucket than vectors that are far apart [13]. To avoid boundary effects, many hash tables are generated. A search by similarity to query q consists in: -1- finding the bucket B in which q hashes (for each hash table), -2- selecting vector candidates in B, -3- returning the k-Nearest Neighbors (kNN) candidates of q.

3.2. Hash function for Euclidean metric

Datar et al. [14] proposed hash function for the Euclidean metric (E2LSH). The hashing function works on tuples of random projections of the form: $h_{\mathbf{a},c}(\mathbf{p}) = \lfloor \frac{\mathbf{a}.\mathbf{p}+c}{W} \rfloor$ where $\mathbf{a}$ is a random vector whose each entry is chosen independently from a Gaussian distribution, c is a real number chosen uniformly in the range $[0, W]$ and W specifies a bin width (the size of search window). Each projection splitting the space by a random set of parallel hyperplanes, the value of the h indicates in which slice of the space, between two hyperplanes the vector has fallen. A hash function is then defined by the concatenation of M functions h. The two parameters chosen for this hash function are the size of search window W and the number of projections M.

3.3. Hash function for χ^2 distance

In this section, we introduce a new LSH family adapted to χ^2 metric. The aim is to work with the χ^2 distance keeping the same efficiency than E2LSH. All the vectors are mapped and clustered in a space of smaller dimension. The clusterization must ensure that the probability of two points falling into the same bucket (a cluster cell) is high when the distance between these two points is small (and conversely). All the points are projected on random vectors $\mathbf{a}$, but unlike E2LSH, we split the projected lines $(\mathbf{a})$ with respect to χ^2. The distance between two consequential bounds X_i and X_{i+1} on any projected line is constant considering the distance χ^2:

$$\forall i \; \chi^2(X_i, X_{i+1}) = \sqrt{\frac{(X_i - X_{i+1})^2}{X_i + X_{i+1}}} = W \qquad (2)$$

Of course, these distances are not constant if considering l_2 distance.

This simple modification of the slicing on the projected lines will produce very different partitions of the whole space that the ones produced by uniform l_2 cutting.

This partition in the sense of distance χ^2 allows us to ensure that when two vectors are close (at a distance less than W after mapping), the probability of collision is higher. We are looking for a sensitive function $h_{\mathbf{a}}$ such as: $h_{\mathbf{a}}(\mathbf{p}) = n$ iif $X_{n-1} \leq \mathbf{a}.\mathbf{p} < X_n$ where the sequence $(X_n)_n$ satisfies Eq. (2) with initial value set to zero: $X_0 = 0$. $\mathbf{a}$ is a random vector whose each entry is chosen independently from a Gaussian distribution with positive value $\mathcal{N}^+(0, 1)$. Eq. (2) leads to the relation between X_n and X_{n-1}:

$$X_n = X_{n-1} + W^2 \frac{\sqrt{8X_{n-1}/W^2 + 1} + 1}{2} \qquad (3)$$

With adding an offset b on the projected line to avoid boundary effect, we then obtain our sensitive hashing function:

$$h_{\mathbf{a},b}(\mathbf{p}) = \frac{\sqrt{\frac{8\mathbf{a}.\mathbf{p}}{W^2} + 1} - 1}{2} + b \qquad (4)$$

for any point $\mathbf{p}$ in the space and with b a real number chosen uniformly at random in the range $[0, 1[$. As in E2LSH, W and M (number of projections) are experimentally set.

4. EXPERIMENTS

Our experiments aim to evaluate how our active learning scheme is competitive in comparison to state-of-the-art angle diversity (AD) approach [12] while decreasing the computational complexity of the search. We also show that our extension of LSH to χ^2 distance allows to use kernel on χ^2 distance and then to improve search.

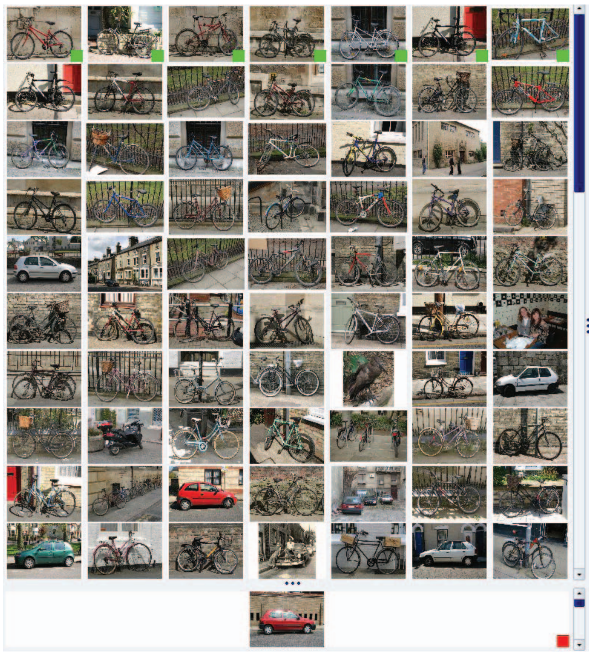

Fig. 2. Graphical interface of our system. Top part: retrieved images; Bottom part: Images selected by the active learner; Green square: image labeled positively; Red square: image labeled negatively

4.1. Experimental setup

We perform evaluation of our method on a 178,500 images database obtained by mixing 6 databases: VOC2006 [15], 2007 and 2008 and TrecVid 2007, 2008 and 2009. We only consider the 10 classes of VOC2006. We consider all images of VOC belonging to theses 10 classes as relevant and all images of TrecVid as irrelevant. The goal of our system is to learn a category of images through a relevance feedback process. The SVM Active learner has no prior knowledge on the image categories. Each image is represented by a 128-dimension vector concatenating 2 histograms: 64 chrominance $CIEL^*a^*b^*$ values and 64 textures from Gabor filters. We evaluate Gaussian RBF kernel function ($K(x, y) = e^{\frac{-d(x,y)^2}{2\sigma^2}}$) with both l_2 and χ^2 distance. Each retrieval session is initialized with one relevant and one irrelevant images. At each iteration, one image selected by the active learner is annotated according to its true label. An illustration of our system RETIN [11] is given on Fig.2. Performances are evaluated with Mean Average Precision on the $TOP200$ which is computed as follows: for each query image, we evaluate the average precision of the TOPN (AP_N), (actually this process is averaged on 100 queries picked at random by class). This is the average of the precision after that the N first images are retrieved by the system. Let $R^N = \{r_1, r_2, \ldots, r_N\}$ be a ranked version of the results. At any given rank j, let $|C \cap R^j|$ be the number of relevant images in the top j of R^N, and C the total number of relevant images in the whole database $\mathcal{B}$. Then AP_N is defined as: $AP_N = \frac{1}{N} \sum_{j=1}^{N} \frac{|R^j \cap C|}{j} \Delta(r_j)$ where $\Delta(r_j) = 1$ if $r_j \in C$ and 0 otherwise. We first compute this mean value over the queries for each class then we average over the 10 class results to compute the MAP_N. In our experiments, we fixed $N = 200$. We chose as parameter $|S| = 200$, $k = 100$ NN by search run and $L = 100$ hash tables of $M = 24$ projections. For the l_2 distance, we take

$W = 1.75 * 10^5$ and $\sigma = 10^5$ for the RBF kernel. For the χ^2 distance, we take $W = 400$ and $\sigma = 200$ for the RBF kernel.

4.2. Results

We first evaluate the accuracy of our fast retrieval method using comparison with exhaustive (but slow) linear search. Four results are reported on Table 1: linear search[1](LIN) *vs* our fast scheme (Fast) combined with two kernels l_2-RBF (l2) and χ^2-RBF (CHI2).

	$MAP_{N=200}$				
Iterations	0	10	20	30	50
LIN l2	9.91	14.83	21.38	25.61	32.05
Fast l2	9.27	16.84	21.61	25.00	30.26
LIN CHI2	13.21	21.62	28.25	33.45	40.42
Fast CHI2	12.87	23.32	28.83	33.07	39.10

Table 1. Performances of our Fast CHI2 method and comparison with linear scanning (LIN) methods

Whatever the kernel (l2 or CHI2), our fast scheme (Fast) provides results almost as good as the linear search (LIN), or sometimes even better. At the 50th iteration, our method has an accuracy of 39.10% and 30.26% respectively for χ^2 and l_2 distances that is less than 2% worse than the linear scheme (respectively 40.42% and 32.05%). We can also see that the χ^2 distance allows to improve quality of the ranking about 8% both for linear approach than fast scheme.

Meanwhile, search time is very different as shown on Fig.3 and Fig.4. The search speedup is always very high. For a classification consisting of 50 iterations, with the l_2 distance, our algorithm takes 1.79s against 13.91s for the linear method. With the χ^2 distance, the speedup is even greater (0.47s against 21.21s). Finally, our method is about 8 and 45 times faster than linear method respectively for the l_2-RBF and χ^2-RBF kernels.

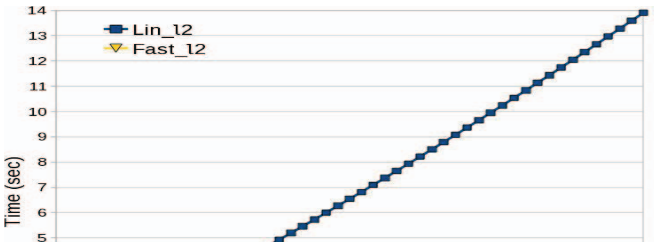

Fig. 3. Time vs number of iterations for l_2-RBF kernels

5. CONCLUSION

In CBIR, image selection and ranking are the two keys for scalability issue. In this paper, we proposed a scalable interactive retrieval strategy. Based on LSH indexing, we quickly select and update a

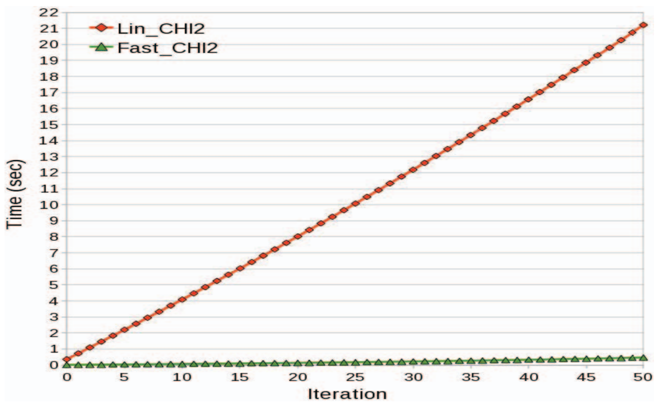

Fig. 4. Time vs number of iterations for χ^2-RBF kernels

pool of relevant images, speeding up both sampling and ranking. We also design a new sub-linear LSH scheme adapted for χ^2 distance. Experimental results on a huge database show that our algorithm achieves the same accuracy than the reference active methods, while dividing the computational complexity by 45.

6. REFERENCES

[1] Steven C.H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu, "Semi-supervised svm batch mode active learning for image retrieval," in *IEEE CVPR*, 2008.

[2] PH. Gosselin and M.. Cord, "Active learning methods for interactive image retrieval," *IEEE Trans. on Image Processing*, vol. 17, pp. 1200–1211, 2008.

[3] PH. Gosselin and Cord., "Semantic kernel updating for content-based image retrieval," in *ISMSE*. dec 2004, IEEE.

[4] E.Y. Chang, S. Tong, KS Goh, and C.W. Chang, "Support vector machine concept-dependent active learning for image retrieval," *IEEE Trans. on Multimedia*, vol. 2, 2005.

[5] E. Chang, B. T. Li, G. Wu, and K. Goh, "Statistical learning for effective visual information retrieval," in *IEEE ICIP*, 2003, pp. 609–612.

[6] M. Crucianu, D. Estevez, V. Oria, and J.-P. Tarel, "Hyperplane queries in a feature-space m-tree for speeding up active learning," in *BDA*, 2007.

[7] N. Panda, K. Goh, and E. Y. Chang, "Active learning in very large databases," *Multimedia Tools and Applications*, vol. 31(3), pp. 249–267, 2006.

[8] J. Peng and D. R. Heisterkamp, "Kernel indexing for relevance feedback image retrieval," in *IEEE ICIP*, 2003.

[9] D. Gorisse, M. Cord, and F. Precioso, "Optimization on active learning strategy for object category retrieval.," in *IEEE ICIP*, nov 2009.

[10] O. Chapelle, P. Haffner, and V.N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE trans. Neural Networks*, pp. 1055–1064, 1999.

[11] P.H. Gosselin, M. Cord, and S. Philipp-Foliguet, "Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval," *CVIU*, 2008.

[12] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *ICML*, 2003, pp. 59–66.

[13] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," *ACM STC*, pp. 604–613, 1998.

[14] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," *SCG*, pp. 253–262, 2004.

[15] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool, "Pascal voc2006," .

[1]using angle diversity sampling