Efficient Category Mining by Leveraging Instance Retrieval

Abhinav Goel Mayank Juneja C. V. Jawahar
Center for Visual Information Technology, International Institute of Information Technology, Hyderabad, India
{abhinav.goel@research.,mayank.juneja@research.,jawahar@}iiit.ac.in

Abstract

We focus on the problem of mining object categories from large datasets like Google Street View images. Mining object categories in these unannotated datasets is an important and useful step to extract meaningful information. Often the location and spatial extent of an object in an image is unknown. Mining objects in such a setting is hard. Recent methods model this problem as learning a separate classifier for each category. This is computationally expensive since a large number of classifiers are required to be trained and evaluated, before one can mine a concise set of meaningful objects. On the other hand, fast and efficient solutions have been proposed for the retrieval of instances (same object) from large databases. We borrow, from instance retrieval pipeline, its strengths and adapt it to speed up category mining. For this, we explore objects which are "near-instances".

We mine several near-instance object categories from images. Using an instance retrieval based solution, we are able to mine certain categories of near-instance objects much faster than an Exemplar SVM based solution.

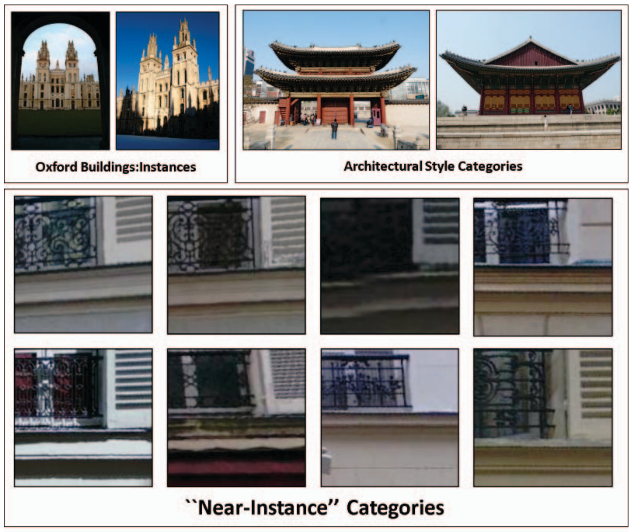

Oxford Buildings:Instances
Architectural Style Categories
``Near-Instance" Categories

Figure 1. On one hand, buildings have been explored as instances in the Oxford buildings dataset[19] (top-left), while as categories of architecture on the other [6] (top-right). Several object categories, such as those shown here (bottom) are much nearer to instances, despite being obtained from different buildings. We are interested in such near-instance categories.

1. Introduction

The retrieval of instances (same objects) [2, 3, 15, 24, 26] as well as categories (different objects, but same semantics) [7, 10, 17] are both prominent research directions. There are well established methods in both areas which solve the problem to a large extent. *Speed*, *accuracy* and *scalability* to a large database size is of prime importance. Results obtained in instance retrieval are typically more accurate, and obtained much faster due to the relatively easy nature of the problem. Category mining is much harder, as visual cues are not sufficient to solve the problem. It involves mining at the level of semantics. One way to solve this problem is to do clustering at image level which is of $O(n^2)$ complexity, n is number of images. Other class of methods are SVM-based methods [17, 23, 8]. Training and evaluation of SVM classifiers, however, is computationally expensive because it involves sliding window over a large number of windows. These methods are not scalable to big datasets. We bridge the gap between instance retrieval and category retrieval, and show how the category mining problem can be solved (to an extent) by efficiently adapting instance retrieval. While maintaining accuracy, we are able to greatly speed up the process of mining categories in unannotated big datasets.

Recently, Gordo *et al.*[13] touched on an important issue in instance retrieval - instance retrieval returns, among the top ranked images, results which are visually similar, but not always semantically consistent with the query. To overcome this issue, they incorporate information from category level labels while searching for instances. We pose a question in the reverse direction - Can instance retrieval be used to benefit retrieval of object categories?

There has been a recent surge of efforts in retrieving instances of the same object from large databases. The challenge is to retrieve accurately, albeit from occlusions, multiple viewpoints and scale variations. The Bag-of-Words

(BoW) method [24] has been the mainstay of instance retrieval techniques for many years. It was initially used to represent images as a histogram of visual words, obtained by vector quantizing SIFT descriptors [16] computed at interest points. Coupled with an inverted index, enabling fast indexing and searching, and tf-idf weighting to downscale the contribution of frequently occurring visual words, this method has become the popular instance retrieval pipeline.

To ensure spatial consistency between the query and the retrieved results obtained by the order-less Bag-of-Words method, geometric verification was introduced as a post processing step [24]. Spatial consistency can be enforced loosely by ensuring that matching visual words are from a small neighborhood in the retrieved image, or strictly by estimating a fundamental matrix from point correspondences using methods such as RANSAC [12]. The computational complexity of the geometric verification step motivated research in incorporating it into the retrieval pipeline [26, 15] itself, rather than being used as a post processing step.

While a plethora of work has been reported in recent past on retrieving similar instances of an object, the problem of image-category search, however, did not receive much attention as a retrieval problem. Rather, the latter has been well received as a learning-based classification problem. [7, 10]. Since the location and spatial extent of the object is unknown, such methods involve an exhaustive evaluation of all possible windows, at multiple spatial scales. This process is computationally expensive. Solutions proposed for solving an unsupervised variant of this problem employ methods borrowed from the data mining community [11, 20, 21, 22, 25]. Even though easily scalable to millions of transactions, adapting mining methods for solving computer vision tasks faces several challenges. The uncertainty in feature representations in images makes it hard for popular itemset mining schemes, such as Apriori [1] and FP-Growth [14] to be directly applied.

State-of-the-art methods [7, 10, 17] in object category retrieval/mining learn a SVM classifier for a set of labeled objects. This classifier (or filter) is applied to all possible patches, in a sliding window fashion, at multiple scales. This number is in the order of millions! The SVM score determines the presence or absence of the object in the image patch. Despite being accurate, classifier based methods are computationally expensive when retrieving object categories from a large data set, which is primarily due to two reasons. The training time of an SVM for a considerable set of positive and negative exemplars is high. Previous works [7, 10] have proposed mining of hard negatives from millions of negatives exemplars. This improves classification accuracy, but at the cost of computation time due to multiple rounds of SVM re-training. [17] proposed a method of learning a separate classifier for each positive exemplar in the given set. This compensates the problem of overly-generic detectors being learnt, and each positive exemplar is given the appropriate amount of attention. However, the number of classifiers increase manifold, leading to an increase in testing time.

We propose an instance-retrieval based solution for mining object categories in large, unannotated databases. Using our method, we perform unsupervised mining, and discover several near-instance object categories automatically. We first talk about the popular instance retrieval pipeline in the next section, which is the base of our approach.

2. Instance Retrieval Pipeline

Given a query image marked by the user, the goal is to achieve all instances (of the same object) which are present in the image database. The popular approach to solving this problem employs the Bag-of-Words method [24]. SIFT descriptors [16], are computed at interest points in the image. The visual codebook used to assign word ids is obtained by clustering a random subset of feature descriptors, using K-means clustering. A visual word identifier is assigned to each feature descriptor, similar descriptors being assigned the same word id. Using large vocabulary sizes (1 million visual words) has shown to give superior performance for retrieving instances [19].

A standard tf-idf weighting scheme is used to downweight the contribution of commonly occurring (and hence not very descriptive) visual words across the database. Each image I in the database is represented as a vector $V(I)$ of visual word occurrences. The similarity between the query image I_q and a database image I_d is defined as the cosine similarity of the two vectors

$$V(I_q).V(I_d)/(||V(I_q)||||V(I_d)|| \tag{1}$$

A rank list of the database images is obtained by comparing every database image against the query. Since the database is already indexed, this step can be done in the order of milliseconds.

Geometric Verification is used as a post processing step to incorporate spatial information into an otherwise orderless Bag-of-Words (BoW) method. The standard method is to use one of the several variants of the RANSAC algorithm [12] to generate multiple fundamental matrices from randomly sampled subsets of a given set of point correspondences. Based on the number of consistent “inliers“, one of the several generated hypotheses is selected. The geometric verification step is computationally expensive, thereby limiting its application to only the top few retrieved results.

The instance retrieval pipeline is both fast and robust, and is proven to be scalable with database size and visual vocabulary size. Instance of the same object can be retrieved in the order of milliseconds from huge data sets containing many thousand images.

3. Can Instance Retrieval work on Caltech categories?

We leverage the existing instance-retrieval pipeline to solve the problem of category retrieval. We propose a simple solution to re-rank the retrieved list of results using the HoG feature descriptor [7], and obtain better results on several object categories. Our method works well for the categories, which exhibit several common characteristics, such as relatively less intra-class variation in visual appearance. We term these categories as near-instance categories. While the instance retrieval pipeline accompanied by spatial verification is able to obtain impressive results for retrieval of same objects, it is not able to do the same for category retrieval. The typical vocabulary size used in category retrieval/classification is 4000. Thus, visual words are better able to capture the variation across categories. On the other hand, in instance retrieval we are interested in exact matches, hence using a much larger vocabulary size (1 million visual words). We choose a vocabulary size of 10,000 visual words, which is more suitable for representing "near instance" categories.

HoG Post-processing: We propose a HoG-based post processing method to improve the quality of retrieved results. HoG feature descriptors [7], with a block spacing stride of 8 pixels, are obtained for the query as well as retrieved images. The retrieved images are re-sized to the size of the query image, ensuring consistency in the dimensions of HoG vector representations. The descriptors for each image are concatenated into a 1-dimensional vector. The Euclidean distance between the HoG vector representations of the query sample and that of the retrieved images is computed. Two retrieved images are compared as

$$\begin{cases} Rank(I_1) < Rank(I_2) & \text{if } d(I_1, q) < d(I_2, q) \\ Rank(I_1) > Rank(I_2) & \text{if } d(I_1, q) > d(I_2, q) \end{cases}$$

HoG based reranking is employed as a post processing step to geometric verification. Accurate bounding boxes around the retrieved object, estimated from the geometric matches between SIFT key-points can easily be obtained. Comparison of vector representations of objects contained in the bounding box compensates for the translation invariant nature of the HoG descriptor. We compare against multiple baselines. In the first baseline, we use the simple instance retrieval pipeline (without spatial verification) to retrieve object categories. As a second baseline, we train a linear SVM classifier for each query exemplar. HoG descriptor of the query exemplar is used as a positive training instance. Thousand negative exemplars are obtained from 10 random samples from each class, except the class of positive exemplar. Each image I in the database is scored as

$$Score(I) = max(I_{m,n}) \tag{2}$$

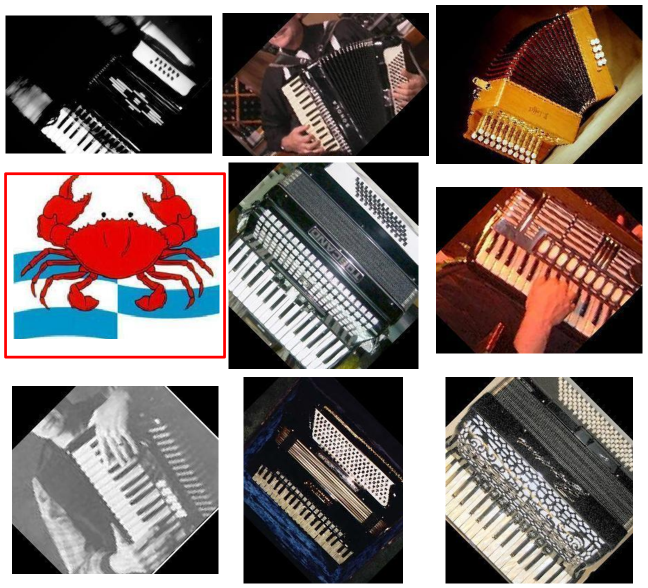


Figure 2. Top retrieved images (from top-left, in row-major form) for query image of class "Accordion". The relative similarity in viewpoint and appearance as compared to other classes makes this category "near-instances". The image outlined in red is a false positive.

where $I_{m,n}$ is the set of all possible HoG windows.

In a separate experiment, we also use the images retrieved by our approach for training a linear SVM. The top 20 retrieved images are used as positive samples, instead of using a single positive exemplar. The rest of the method is similar to the second baseline.

We test our method on the Caltech 101 dataset [9]. SIFT descriptors are computed for each image in the database at interest points obtained using a Difference of Gaussian (DoG) detector. A random subset of 100,000 SIFT descriptors obtained from the database images is used to create a vocabulary. The vocabulary size is chosen to be considerable large (10,000). This ensures that only significantly similar features are captured by each visual word. 5 images are randomly selected from each class to create the set of query images.

Table 1 compares the retrieval accuracy of our method with other baselines. Two measures have been used to report retrieval accuracy. (a) The precision obtained for the top 10 retrieved samples was computed. The average over 5 randomly chosen queries, known as *Mean Precision at 10* has been reported for each category. (b) The average precision was computed for the complete rank list. The average over 5 randomly chosen queries *Mean Average Precision* has been reported for each category.

Based on the results of our experiments, we are successfully able to divide the 101 Object Categories of the Caltech dataset into 3 sets. Images in each of the three sets exhibit different properties, based on which we infer how easy/tough it is to retrieve from them. The first set of cate-

	Mean Precision At 10				Mean Average Precision		
Class	**IR**	**IR + HoG**	**1E-SVM**	**IR+HoG+1E-SVM**	**IR**	**1E-SVM**	**IR+HoG+1E-SVM**
Accordion	0.60	0.82	0.51	0.15	0.17	0.02	0.06
Windsor Chair	0.62	0.68	0.48	0.31	0.36	0.16	0.17
Chair	0.11	0.10	0.12	0.15	0.02	0.06	0.09
Watch	0.11	0.12	0.11	0.00	0.02	0.01	0.02
Schooner	0.11	0.12	0.22	0.00	0.03	0.02	0.007
Electric Guitar	0.11	0.10	0.15	0.00	0.02	0.02	0.005

Table 1. Comparison of Various Methods for Object Category Retrieval. Cells marked in green are object categories which are near-instances. Cells marked in yellow are neutral categories. Cells marked in red are tough categories. IR = Instance Retrieval, IR + HoG = Instance Retrieval + HoG-based Re-ranking, 1E-SVM = One Exemplar SVM, 1C-SVM = One Class SVM

gories (highlighted in green in Table 1) are those which are *near-instances*. Using an Instance Retrieval based method accompanied by a HoG-based post processing step gives much better performance, both in terms of precision at 10 as well as average precision, than an exemplar SVM based method, as well as the standard instance retrieval pipeline. Figure 2 shows the top retrieved results for a query of the class "Accordion". Even though the images contain different models, there is similarity in appearance and viewpoint across all images, which makes it an ideal candidate for a "near-instances" object category.

The second set of categories are *neutral* categories (highlighted in yellow in Table 1). From observing the images belonging to these classes, we inferred that even though there is some similarity in visual appearance across images, it is not sufficient for instance retrieval to work. This observation is supplemented by the similar retrieval performance obtained by our method as well as 1-Exemplar SVM for these categories. The third set are *tough* categories (highlighted in red in Table 1). Images of objects in these classes exhibit high intra-class variance, due to which standard category retrieval methods such as SVM often outperform our method of retrieving near-instances.

4. Efficient Category Mining

There have been many attempts in the past at adapting well-established data-mining techniques for unsupervised mining in images. Quack et al [22] mine clusters containing similar image content from geo-tagged imagery obtained from Flickr. The mined clusters are then classified into objects/events, and text labels, obtained from Wikipedia articles, are associated with each cluster. Sets of discriminative patches, termed as Grouplets, were mined for modeling the interactions between humans and objects in [25].

Classifier based methods apply a SVM classifier to all possible windows, which is computationally expensive. We propose an instance-based solution for solving the problem of discovering near-instance object categories in a large dataset. We progressively evaluate patches at increasing spatial scales, all the while rejecting uninteresting patches at every step. As a result, we are able to quickly discover a set of semantically meaningful near-instance object categories.

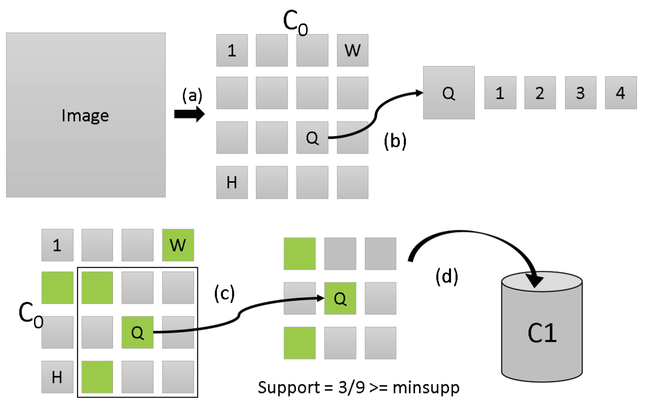


Figure 3. Our mining pipeline: (a) Image is divided into small, square patches. (b) Each patch is used to retrieve similar patches from the database. (c) Patch with high Goodness Score (marked in green), are grown by grouping with nearby patches. (d) Grouped patches which contain atleast *minsupp* good patches are used for further mining/retrieval.

The given image is first divided into square patches of fixed size of 25 $\times$ 25 pixels. This starting set of patches constitutes the initial candidate set L_0. Each candidate patch is evaluated and a score assigned to it which signifies the goodness of the patch as belonging to a larger, semantically meaningful and frequently occurring near-instance category. The top scoring patches from the candidate set L_k, based on the "Goodness" score, are used to create the candidate set L_{k+1}, where k denotes the level of grouping. The patches used to create the subsequent candidate sets are increased in size by incorporating a larger region around the current patch. Figure 3 gives a step by step overview of our method. We now talk about the algorithm in detail.

Patch Evaluation: Each patch in the candidate set is represented using a histogram of visual words. This patch is now used as a query to retrieve visually similar patches across the image database. The top few retrieved patches are re-ranked using a spatial verification step. We fit an affine transformation with 3 degrees of freedom (dof) between the query and retrieved patches. Similar to [2, 19], hypotheses are generated from single point correspondences, which speeds up matching. To accommodate matches on

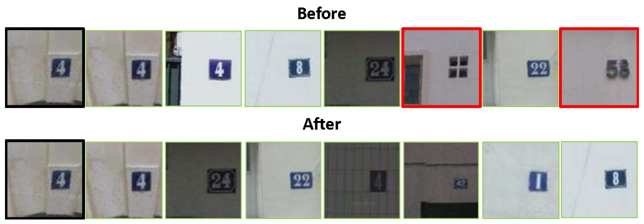


Figure 4. Figure showing improvement in cluster purity after HoG Re-ranking for a particular patch (House Numbers) at Level 2 of mining. The image outlined in black is the initial patch. Two false positives (outlined in red) appeared after mining (Row 1). These were rejected after re-ranking using HoG (Row 2).

near-instance categories, we allow large-re-projection errors. Due to the small size of the query patch, the number of matches to be evaluated is typically small. The retrieved list of patches is characterized by many false positives. This is mainly due to (a) using a reasonably large vocabulary size for matching object categories (not instances), and (b) small size of the query patch, which results in very small number of matches from spatial verification. We propose a goodness score to measure the quality of the retrieved patches obtained in the cluster. The Goodness Score of a patch is computed as

$$GScore(I_p) = \left(\frac{\sum_{i=1}^{N} d(HoG_p, HoG_i)}{N.A} \right)^{-1} \quad (3)$$

where $d(HoG_p, HoG_i)$ is the Euclidean distance between the HoG vector representations of the query patch I_p and the i^{th} retrieved patch, N is the number of patches retrieved, and A is the area of the query patch. The patches are resized to the size of the query before the HoG descriptors are computed, to ensure consistency in dimension across HoG vector representations. A larger value of N enforces a much stricter goodness constraint on each patch, at the cost of increased computation in calculating the distances between the HoG vector representations. Normalizing by the area enables fair comparison between the scores assigned to patches of varying sizes.

Patch Growing: The patches obtained from candidate set L_k with a high Goodness Score are grown to create a similar set L_{k+1} for next level of evaluation. A simple solution for growing the patch can be to select 8-connected neighboring patches of the current patch. A fundamental problem with this is that some parts of the new patch might be uninteresting, and may not have any semantic association with the object which, if present, we are trying to discover. Another method of selecting useful patches is non-maximum suppression (in terms of Goodness Score), where a larger image region around each local

Task	Time Taken (in sec.)
Similarity Search	0.797
Spatial Verification	0.324
HoG Score Computation	0.468

Table 2. Time taken to evaluate a single patch of size 50 $\times$ 50 pixels.

maximum is selected. At the k^{th} level of grouping, we select $(2k + 1) \times (2k + 1)$ square patches, where the size of each patch is 25 $\times$ 25 pixels. The support of this new patch $(Patch_k)$ of increased size is computed as

$$Supp(Patch_k) = \frac{M}{(2k+1).(2k+1)} \quad (4)$$

where M is the number of patches in the image region which are among the top ranked ones based on Goodness Score. Regions with support value greater than a minimum threshold *minsupp* are used to create the candidate set L_{k+1} for the next level of evaluation.

Mining Near-Instances: After multiple levels of refining, we are left with few, reasonably large sized patches. Similar to the method of patch evaluation, these reasonably large patches are used to query from the database. We retrieve the top 100 results for each query region. A HoG based post-processing step is employed to refine the clusters of near-instance object categories. The final rank of a retrieved patch is computed as

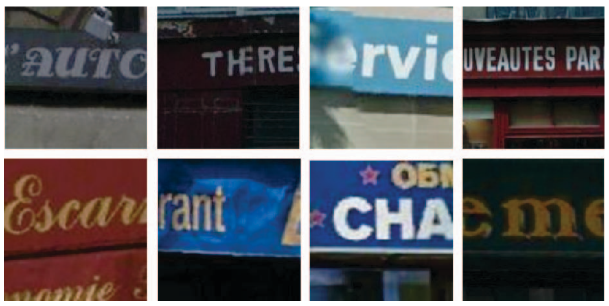
$$Rank = max(Rank_{SIFT}, Rank_{HoG}) \quad (5)$$

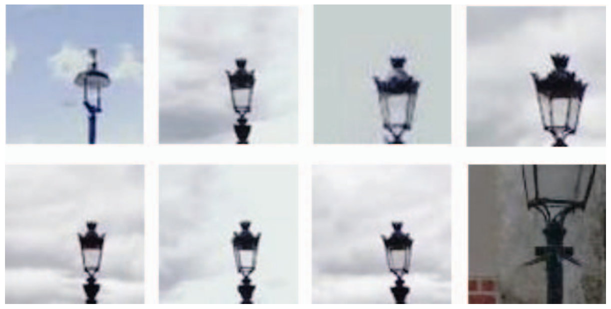
where $Rank_{SIFT}$ is the original rank in retrieved list, and $Rank_{HoG}$ is computed based on the Euclidean distance between the HoG templates of the query and retrieved image regions. A lower value of $Rank$ signifies that the result appears at the top of the rank list.

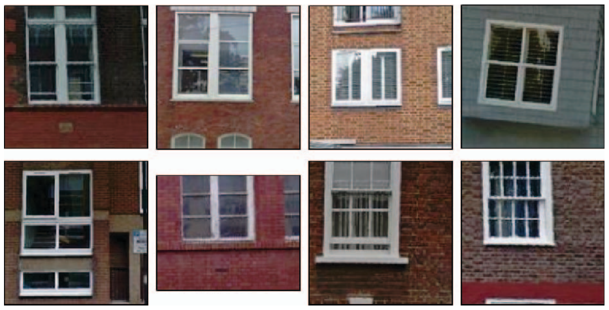
Our HoG-based re-ranking method suppresses false positives by pushing them away from the query, while retaining true positives in the top results. This method of pushing away false positives retains the ranking information obtained by SIFT matching, which is important for retrieval of near-instance object categories. Figure 4 visualizes improvement in purity of one particular mined cluster (House Numbers) after rejecting false positives using our method.

4.1. Computational Complexity

Our method is fast. Table 2 gives the time taken by each module of our approach for evaluating a single patch. The implementation has been done in MATLAB on a standard PC with 4 GB RAM. Evaluating multiple patches can easily be done in parallel since these are independent tasks. An exact time comparison with previous works for category retrieval and mining may not be possible due to the different settings (data sets, features) employed earlier. We provide


(a) Text Signs


(b) Lamps


(c) Windows

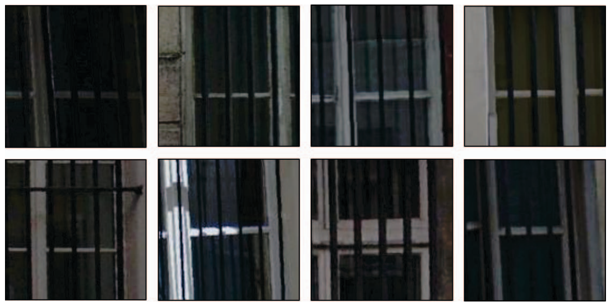

(d) Grills

Figure 5. Various near-instance Object Categories which were automatically discovered by our mining method

a theoretical evaluation of the time complexity of our approach with Exemplar SVM based approach for mining

Mining/Retrieval of Object Categories using Exemplar SVM comprises of two stages - training and testing. Training a linear SVM involves maximizing the margin between the single positive exemplar and millions of negative exemplars. For both primal and dual optimization, the complexity is $O(max(n, d)min(n, d)^2)$ [5], where n is the number of training instances (1 Positive + Many Negatives), and d is the dimension of representation of an exemplar. Both retrieval as well as mining are unsupervised methods, and require no time for training, as opposed to an SVM classifier. SVM testing scales linearly with the number of positive instances (but not the negatives), since a separate classifier is learnt for each positive exemplar. For an image (of dimension $H \times W$), applying a classifier (at say S spatial scales) takes $O(H \times W)$ computation (since $S << H \times W$). We can evaluate all E Exemplar-SVMs over $|D|$ database images in $O(E \times |D| \times H \times W)$.

4.2. Results

We perform both quantitative and qualitative evaluation of our method. For the discovery of near-instance object categories, we use the data set of Google Street View Images as provided by [8]. The images were obtained for 12 cities: : Paris, London, Prague, Barcelona, Milan, New York, Boston, Philadelphia, San Francisco, San Paulo, Mexico City, and Tokyo. We downloaded 25,278 images from Google Street View, the dimensions of each image being 936×537 pixels. This gives us $38 \times 22 = 836$ patches per image, when the starting patch size is 25×25 pixels. A starting set of 100 seed images was used for the purpose of discovery.

Image Representation: SIFT descriptors [16] are computed at affine-invariant Hessian regions [18]. Each SIFT descriptor is assigned a visual word id from a visual codebook. For image representation, we use a vocabulary trained on the Oxford Buildings data set [19] comprising of 100,000 visual words. An inverted index is built from the database images, which enables fast searching. A standard tf-idf weighting scheme is used, which suppresses the contribution of frequently occurring visual words. The Hellinger kernel is used for comparison of image vector representations, which has shown superiority over Euclidean distance in texture classification and image categorization tasks [4].

Quantitative Evaluation: For Quantitative evaluation, we compare with Exemplar-SVM method. A linear SVM is learnt for a single positive exemplar, and all possible windows from database images (other than query image) as the negative exemplars. We perform 10 iterations of retraining, and 5 iterations of mining hard negatives in each iteration of re-training. 100 hard negatives are mined in each iteration (of re-training) and added to the negative set. The PEGASOS SVM solver in the VLFeat library is used for training. The bias multiplier is set to 100, and the regularization parameter $\lambda = 100/N$, where N is the number of training samples. For training, the classifier is run on a subset of

Instance based Solution	**Time** (in s)	**Exemplar SVM**	**Time** (in min)
Similarity Search	0.012	Training	22.66
Geometric Verification	0.411	Testing	26.66
HoG Reranking	0.070		
Total (in mins)	**0.0082**		**49.32**

Table 3. Time taken to evaluate a single patch of size 50×50 pixels.

200 database images, at 4 spatial scales. We also perform retrieval on this subset using the same positive exemplar, followed by the HoG post-processing method. Table 3 summarizes the computation times for both methods.

Qualitative Evaluation: We perform mining upto 3 levels. The top 20 results were retrieved for each patch upto the penultimate mining level. Top 100 results were obtained for visualization at the last level. From the set of 100 seed images, we discovered several different concepts (at multiple levels), which fall into the category of near-instance objects. Figure 5 showcases examples of few near-instance object categories discovered by our method. The include balconies, street lamps, windows, and banners containing text. The type of object category discovered varies with the level at which we are mining. For example, "House Signs" and "Street Lamps" (Figure 5(b)) were discovered on the second level of mining, where the patch size is relatively low. As we move higher, the type of near-instance object categories discovered are those which cover a larger spatial extent, such as "Windows" (Figure 5(c)) and "Text Banners" (Figure 5(a)).

5. Conclusion

There are several object categories, which are near-instances. Objects in these categories exhibit less intra-class variance, which allows for instance retrieval techniques to be applied on them. We adapt the instance retrieval pipeline to solve tasks - category retrieval, and category mining, previous methods for which are computationally expensive. In both cases, we are successfully able to retrieve/mine near-instance object categories. Our method, however, is restricted to near-instance categories alone. For other categories, considerable work needs to be done to bridge the gap between instance and category retrieval.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994. 2

[2] R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *IEEE International Conference on Computer Vision*, 2011. 1, 4

[3] R. Arandjelović and A. Zisserman. Name that sculpture. In *ACM ICMR*, 2012. 1

[4] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 6

[5] O. Chapelle. Training a support vector machine in the primal. *Neural Comput.*, 2007. 6

[6] W.-T. Chu and M.-H. Tsai. Visual pattern discovery for architecture image classification and product image search. In *ACM ICMR*, 2012. 1

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2, 3

[8] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012. 1, 6

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 2006. 3

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 2010. 1, 2

[11] B. Fernando, . Fromont, and T. Tuytelaars. Effective use of frequent itemset mining for image classification. In *ECCV*, 2012. 2

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. 2

[13] A. Gordo, J. A. Rodríguez-Serrano, F. Perronnin, and E. Valveny. Leveraging category-level labels for instance-level image retrieval. In *CVPR*, 2012. 1

[14] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD*, 2000. 2

[15] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008. 1, 2

[16] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 2, 6

[17] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 1, 2

[18] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, 2002. 6

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 1, 2, 4, 6

[20] T. Quack, V. Ferrari, B. Leibe, and L. J. V. Gool. Efficient mining of frequent and distinctive feature configurations. In *ICCV*, 2007. 2

[21] T. Quack, V. Ferrari, and L. Van Gool. Video mining with frequent itemset configurations. CIVR, 2006. 2

[22] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *CIVR*, 2008. 2, 4

[23] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 1

[24] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *CVPR*, 2003. 1, 2

[25] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010. 2, 4

[26] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011. 1, 2