



**HAL**  
open science

# FEATURE SET CONSOLIDATION FOR OBJECT REPRESENTATION BY PARTS

Shamsuddin N Ladha, Piyush Yadav, Shailesh Deshpande

► **To cite this version:**

Shamsuddin N Ladha, Piyush Yadav, Shailesh Deshpande. FEATURE SET CONSOLIDATION FOR OBJECT REPRESENTATION BY PARTS. 2018. hal-01708549

**HAL Id: hal-01708549**

**<https://inria.hal.science/hal-01708549>**

Preprint submitted on 13 Feb 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# FEATURE SET CONSOLIDATION FOR OBJECT REPRESENTATION BY PARTS

*Shamsuddin N. Ladha, Piyush Yadav, Shailesh Deshpande*

TCS Innovation Labs, Pune, India

## ABSTRACT

Image data is growing by leaps and bounds. Machine learning based applications that run on image datasets increasingly use local image feature descriptors. In a sense, we can now visualize images as objects with features as parts. Typically there are thousands of local features per image, resulting in an explosion of feature set size for already humungous image datasets. In this paper we present a feature set consolidation strategy based on two aspects: pruning of non-discriminatory features across different object types and association of matching features for the same object type. We showcase the effectiveness of our consolidation strategy by performing classification on a building dataset. Our method reduces storage space footprint (~5%) and classification runtime (~4%), and increases classification accuracy (~2%).

**Index Terms**—Object representation by parts, part pruning, SIFT, classification, local image descriptor

## 1. INTRODUCTION

Traditionally, for vision based machine learning problems, each object instance in the dataset has been cohesively represented as a single—possibly very high dimensional—vector. This approach could be attributed to the observation that certain machine learning problems are better handled in high dimensional spaces (e.g. Support Vector Machine (SVM) [1]). However, with the availability of discriminating local features (such as SIFT [2]) it has been possible to conjure objects as a composition of parts (or components). For example, given an image of an animal (object), we refer as parts all the extracted local features (by a feature extraction algorithm). Here we have assumed that each image has only one object. This notion of representation by parts is different than the cognitive notion of physical parts of the animal such as head, body, etc.

While locally discriminative features may provide a richer description of objects, they pose several new challenges for object representation and learning,

- A fixed high dimensional vector representation of object remains no longer practical as the number and order of local features generally vary across objects.
- The number of parts that represent a single object are typically in the order of hundreds (e.g. SIFT features)

thus increasing the space and time complexity to store and process these features.

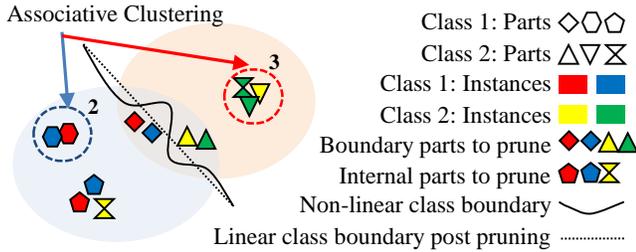
- Both intra-class and inter-class variability shoots up significantly due to richer object representation.
- The dimensionality of individual features is generally low. This could be problematic for learning algorithms where high dimensionality is desired (e.g. SVM).

To address these challenges several approaches are found in the literature. These approaches can be broadly classified into three types, i.e., approaches that reduce a) the number of features; b) the dimensions of feature; c) both the number and dimensions of feature.

Common approaches such as object representation by top  $N$  features or feature clustering aim at reducing the number of features. In [3], the authors reduced the number of SIFT features used for indoor scene representation based on the observation that a majority of the detected key points do not match between images that share common camera view point. Authors of [4] represented objects by clustering SIFT features of an object with  $k$ -means to solve content based image retrieval (CBIR) problem.

Methods such as PCA-SIFT [5] and feature quantization [7] reduce feature dimensionality. For example, [6] used PCA-SIFT and applied locality sensitive hashing (LSH) for fast object retrieval. [7] introduced a method to quantize sift features based on inverted intensity images and then performed clustering to create a codebook for image matching. This method reduces not only feature dimensionality but also the size of feature set.

A key aspect that has not been addressed, in these methods, is elimination of features that are non-discriminative across different objects. Although SIFT features are discriminative for a given image, there is a possibility that some of these features may lose their discriminatory powers in presence of similar features from altogether different objects. In other words, the distance between some of the SIFT features belonging to different objects could be small enough to cause incorrect feature matching across objects. In this paper we intend to bridge the gap in the existing methods by pruning non-discriminative SIFT features across objects. In addition, we improvise feature clustering by integrating concepts of associative memory from [8]. Our approach that is a combination of non-discriminatory feature elimination and associative feature clustering is termed as feature set “consolidation”. In the following section, we describe in detail both aspects of consolidation.



**Figure 1: Object representation by parts**

### 3. FEATURE SET CONSOLIDATION

Consider a simplified schematic (see Figure 1), that demonstrates object representation by parts. The figure shows two object instances of two different object classes. Each object instance is represented by corresponding three parts. Consolidation has two aspects, i.e., elimination and associative clustering. To explore the impact of consolidation in greater detail we will study it in the context of object classification problem.

For the schematic problem, accurate classification of objects would require learning a complex decision boundary (in this case non-linear). If we eliminate the non-discriminatory parts that lie near the decision boundary then the boundary simplifies (becomes linear in this case). In any case, elimination of non-discriminatory parts will not increase the complexity of the decision boundary because any non-discriminatory part could be labeled as internal (not impacting decision boundary) or boundary part.

In general, while building a classifier it is important to strike a good balance between the errors due to bias and variance in order to have good generalization accuracy. It is evident that pruning of non-discriminative parts helps reduce not only bias—by reducing model complexity—but also variance—by reducing unwarranted variance in the dataset. Pruning, as demonstrated in the schematic (Figure 1), helps reduce errors due to bias and variance.

While pruning of parts may sound similar to pruning of object instances [9], there are some key differences. In general, pruning of parts—unlike object pruning—does not lead to loss of representation of entire object except when all the parts that compose an object are pruned. Inter-class object parts are likely to be closer in hyper-space than inter-class objects due to low dimensionality of parts and large number of object parts.

Large number of object parts allows us to not only be aggressive while pruning but also statistically determine the class of an object based on the underlying class distribution over parts. Thus an object will be classified accurately as long as the majority of its parts are correctly classified.

Our approach can be used to complement existing approaches for feature reduction mentioned earlier.

Next, we look at the associative clustering aspect of feature set consolidation. Typical clustering methods substitute parts from same object class that belong to a single cluster with a single representative part. The strength of the cluster that is indicative of the association of cluster members is not utilized. In addition to typical clustering, we store the strength of each cluster. At the time of classification of an object, the cluster strength contributes to the class distribution that will be used to determine the class of the query object. Higher the cluster strength larger will be the contribution to the distribution and vice-versa. This concept is inspired from sparse distributed memory (SDM) [8]. However, it is significantly different from SDM. While SDM permits clustering of instances with same or different class labels, our approach only permits clustering of same label instances. Thus clusters generated by our method are pure, i.e., all the cluster members belong to the same object class.

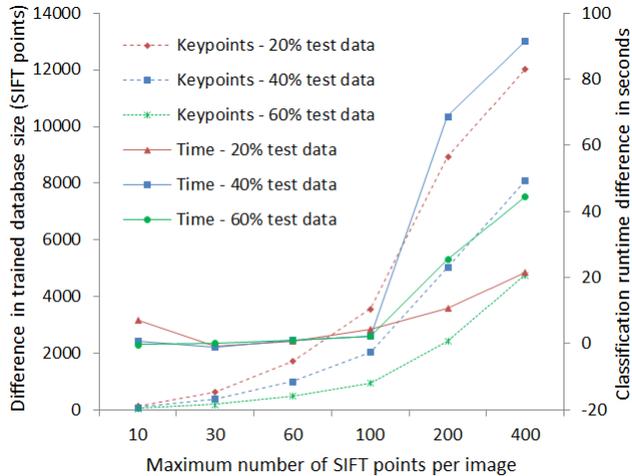
The two aspects of consolidation, i.e., pruning and associative clustering are complementary. Together they contribute to reduction in storage space, classification runtime and help strike a balance between model bias and dataset variance.

### 4. EXPERIMENTS AND RESULTS

To demonstrate the concept of pruning and its effectiveness we have implemented a parts based image classification system. Classification experiments consist of following three steps,

1. Store the training data set features in a database in an off-line mode.
2. At runtime match the test data set features to those stored in the database. This results in assignment of a class label to each feature in the test set.
3. Derive a class label for each query object from its feature set labels (from Step 2).

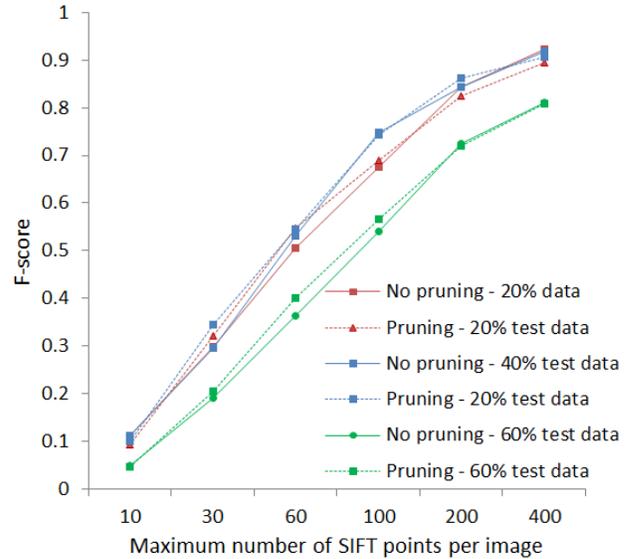
Our experiments were conducted on a standard Intel core i5 processor with Windows 7 Professional operating system and 4GB RAM. The system was developed in Python with OpenCV port [11]. We have used the Zurich building dataset [10] for this purpose. The dataset with a total of 1005 images consists of five views of 201 different buildings in Zurich. Each view image has dimensions 640 x 480. Local features (SIFT) were extracted for each image where each SIFT feature interest point, in the context of this paper, represented a part of the image. The number of SIFT points extracted per image ranged from about 1000 to 2000. The dataset was split into training and test sets. Two sets of classification experiments were conducted, one without and another with consolidation of features. For the purpose of consolidation, the SIFT points were matched using brute force matcher. Matched points with distance less than a specific threshold were only considered as appropriate matches. This step of storing training data, without or with consolidation, was done in an off-line mode.



**Figure 2: Experiments showcasing reduction in the number of SIFT points stored and classification runtime due to pruning.**

At runtime, during classification of a query image, its local features were extracted and matched to the stored features resulting in a distribution over class labels. The query image is labeled with the mode of the distribution. A range of experiments were conducted on the dataset to assess the impact of consolidation. Experiments with varying number of SIFT features, different size training datasets, and matching threshold were performed. Table 1, Table 2, and Table 3 respectively list the number of key points, classification time, and F-score values for these experiments. Figure 1 and Figure 2 respectively show the plot of difference, without and with consolidation, in the number of SIFT key points (left vertical axis) and classification runtime (right vertical axis), as a function of the maximum number of SIFT features extracted per image for three different sizes of test data. We see that as the number of extracted SIFT points per image increase, consolidation leads to increase in savings of storage space and reduction in classification runtime. On an average our method reduces storage space requirements by 5% and shortens classification runtime by 4%. As the size of training set reduces, reduction in storage space decreases (the dashed lines). These results should be assessed in conjunction with the results of Figure 3 that show the classification performance of the two approaches (without and with consolidation) as the number of SIFT points extracted per image varies.

For each experiment, assessment of the classification results was done by computing F-score. Table 3 lists the actual F-scores obtained for various experimental setups. Within a given experimental setup (e.g. with 20% dataset held out), we observe that the F-score steadily increases for both without and with pruning stored features.



**Figure 3: Graph with F-scores for different experiments with and without pruning.**

**Table 1: Number of features stored for different experimental setups.**

Dataset held out % $\rightarrow$	Experimental Setup 1 (20%)		Experimental Setup 2 (40%)		Experimental Setup 3 (60%)	
	Total features stored	Pruned features stored	Total features stored	Pruned features stored	Total features stored	Pruned features stored
SIFT features per image $\downarrow$						
10	8040	7892	6030	5946	4020	3974
30	24120	23488	18090	17722	12060	11875
60	48240	46515	36180	35183	24120	23632
100	80400	76837	60300	58255	40200	39249
200	160800	151861	120600	115565	80400	77981
400	321499	309461	241099	233006	160699	155940

**Table 2: Classification runtime (in seconds) for different experimental setups.**

Dataset held out % $\rightarrow$	Experimental Setup 1 (20%)		Experimental Setup 2 (40%)		Experimental Setup 3 (60%)	
	Time (No pruning)	Time (With pruning)	Time (No pruning)	Time (With pruning)	Time (No pruning)	Time (With pruning)
SIFT features per image $\downarrow$						
10	74.16	67.16	135	134.32	201.34	201.64
30	72.78	73.57	142.38	143.35	210.01	209.87
60	91.93	91.08	171.84	170.73	236.74	235.68
100	133.84	129.38	237.14	234.89	297.72	295.54
200	352.59	341.84	573.09	504.33	573.69	548.1
400	1007	985.38	1542.26	1450.8	1618.37	1574

**Table 3: Classification F-scores for different experimental setups.**

Dataset held out % $\rightarrow$	Experimental Setup 1 (20%)		Experimental Setup 2 (40%)		Experimental Setup 3 (60%)	
	F-score (No pruning)	F-score (With pruning)	F-score (No pruning)	F-score (With pruning)	F-score (No pruning)	F-score (With pruning)
SIFT features per image $\downarrow$						
10	0.11	0.10	0.11	0.10	0.04	0.05
<b>30</b>	<b>0.30</b>	<b>0.32</b>	<b>0.30</b>	<b>0.34</b>	<b>0.19</b>	<b>0.20</b>
<b>60</b>	<b>0.51</b>	<b>0.55</b>	<b>0.53</b>	<b>0.54</b>	<b>0.36</b>	<b>0.40</b>
<b>100</b>	<b>0.68</b>	<b>0.69</b>	<b>0.75</b>	<b>0.74</b>	<b>0.54</b>	<b>0.57</b>
200	0.84	0.83	0.84	0.86	0.73	0.72
400	0.92	0.90	0.92	0.90	0.81	0.81



**Figure 4: Sample matching points (in blue) from two different images (separated by the yellow marker) that were pruned.**

The F-scores for classification with consolidation is better on an average by about 2-3% particularly for experiments where the number of SIFT features are limited (highlighted entries in Table 3). For other experiments the F-scores with consolidation are similar to those without consolidation. This trend is observed for all the experimental setups (as seen in Figure 3) indicating no adverse effect of our pruning strategy on classification outcomes.

A sample result of unrelated matching key points, one point from building façade and another from zebra crossings, which were pruned from two different building images, is shown in Figure 4.

## 5. SUMMARY

In this paper, we have presented a feature set consolidation strategy suitable for scenarios where objects are represented by parts. During the process of consolidation not only the non-discriminatory parts (belonging to different objects), that often complicate machine learning models (e.g. the classifier decision boundary), were eliminated but also the matching parts (across different instances of the same object class) were strengthened. Our experiments have demonstrated that such a strategy is efficient as it leads to significant savings in terms of both storage space and classification runtime without compromising classification

performance. With ever exploding world of image data and greatly improved local descriptors, such consolidation strategies will be helpful in realizing ever increasing memory and time intensive applications on limited computing power devices.

## 6. REFERENCES

- [1] B. E. Boser, I. M. Guyon and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Computational Learning Theory*, New York, 1992.
- [2] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [3] L. Ledwich and S. Williams, "Reduced sift features for image retrieval and indoor localisation," in *Australian Conference on Robotics and Automation*, 2004.
- [4] G. A. Montazer and D. Giveki, "Content based image retrieval system using clustered scale invariant feature transforms," *International Journal for Light and Electron Optics*, vol. 126, no. 18, pp. 1695-1699, 2015.
- [5] K. Yan and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition*, 2004.
- [6] J. J. Foo and R. Sinha, "Pruning SIFT for Scalable Near-duplicate Image Matching," in *Conference on Australasian Database*, Ballarat, 2007.
- [7] J. S. Hare, S. Samangoeei and P. H. Lewis, "Efficient Clustering and Quantisation of SIFT Features: Exploiting Characteristics of the SIFT Descriptor and Interest Region Detectors Under Image Inversion," in *ACM International Conference on Multimedia Retrieval*, 2011.
- [8] P. Kanerva, *Sparse Distributed Memory*, Cambridge: MIT Press, 1988.
- [9] D. R. Wilson and T. R. Martinez, "Instance Pruning Techniques," in *International Conference on Machine Learning*, San Francisco, 1997.
- [10] C. V. L. E. Zurich, "Zurich Building Image Database," [Online]. Available: <http://www.vision.ee.ethz.ch/showroom/zubud/>. [Accessed 2 February 2016].
- [11] OpenCV Dev Team, "OpenCV-Python Tutorials," [Online]. Available: [http://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_tutorials.html](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html). [Accessed 2 February 2016].
- [12] J. Snaider, R. McCall and S. Franklin, "The LIDA Framework As a General Tool for AGI," in *International Conference on Artificial General Intelligence*, Mountain View, 2011.