IMAGE CATEGORIZATION THROUGH OPTIMUM PATH FOREST AND VISUAL WORDS

João Paulo Papa

UNESP – Univ. Estadual Paulista
Bauru, SP, Brazil
papa@fc.unesp.br

*Anderson Rocha**

University of Campinas (Unicamp)
Campinas, SP, Brazil
anderson.rocha@ic.unicamp.br

ABSTRACT

Different from the first attempts to solve the image categorization problem (often based on global features), recently, several researchers have been tackling this research branch through a new vantage point – using features around locally invariant interest points and visual dictionaries. Although several advances have been done in the visual dictionaries literature in the past few years, a problem we still need to cope with is calculation of the number of representative words in the dictionary. Therefore, in this paper we introduce a new solution for automatically finding the number of visual words in an N-Way image categorization problem by means of supervised pattern classification based on optimum-path forest.

***Index Terms*—** Image Categorization, Visual Dictionaries, Local Interest Points, Optimum Path Forest

1. INTRODUCTION

In many real-life applications, we have to make decisions based upon images and, in the hope of understanding them, data mining and machine learning approaches are paramount. In this scenario, one of the problems we often face is image categorization.

Image Categorization comprises the body of techniques that aim at distinguishing between image classes, pointing out the global semantic type of an image. Different from the first attempts to solve the categorization problem (often based on global features), recently, several researchers have been tackling this research branch through a new vantage point – using features around locally invariant interest points. Although originally developed for large baseline correspondence applications, there are some attempts for image retrieval and classification [1–4] with results often outperforming previous global-based descriptors.

These approaches stem on locally invariant interest points aiming at representing every image in a collection using a large number of points of interest (PoIs). Later on, it is possible to calculate a local descriptor around each PoI, and store it in an indexing data structure [5].

The hypothesis behind these approaches is that the PoIs convey more information than the other points in the image. Therefore, PoIs can be robustly estimated, even if the image suffers distortions – the major criterion of quality for a PoI algorithm is repeatability [5].

After locating them, each PoI is described by the analysis of a small patch around it. The literature has shown that local descriptors computed around points of interest are more robust to represent image's nuances than global descriptors [1–4]. However, this representative power comes with an advantage and a drawback. When searching for a specific target, this discriminative power is extremely important. Notwithstanding, when searching for complex categories, it is a problem since the ability to generalize becomes paramount. Therefore, as these solutions are often designed for exact matching, they do not translate directly into good results for image classification.

A possible solution to this problem is the technique of visual dictionaries which considers the high-dimensional descriptor space and split it into multiple regions. Usually, one uses a non-supervised learning technique (e.g., clustering) for this task in order to find the most discriminative points of interest. Each region of PoIs, becomes a visual "word" of a "dictionary".

After the creation of the vocabulary, one summarizes each image of the collection analyzing each of its PoIs and assigning them the closest word in the dictionary. In the end, each image is represented by a set of visual words [5, 6].

With this simple idea, the biggest challenge is to design a good dictionary. Although several advances have been done in the visual dictionaries literature in the past few years, a problem we still need to cope with is the number of representative words in the dictionary, or, in other words, the number of dimensions $\mathcal{K}$ of the Hilbert space $\mathcal{H}$ we need to map the PoIs to. Therefore, in this paper we introduce a new solution for automatically finding the number of visual words in an N-Way image categorization scenario. For that, we use a technique we call *Optimum Path Forest* (OPF) [7] which is itself a very fast classifier with small computational footprint when compared to traditional classifiers such as *Support Vector Machines* (SVMs).

Our contribution in this paper is twofold: (1) we present an automatic approach for computing the number of words; and (2) the approach itself is a fast classifier with competitive results with respect to classic classifiers such as *Support Vector Machines*.

2. VISUAL VOCABULARIES FOR IMAGE CATEGORIZATION

Visual dictionaries constitute a robust representation approach in which each image is treated as a collection of regions. For each region, the only information we care about is its appearance [8].

Our objective when creating a visual dictionary is to learn, from a training set of examples, the generative model [9] that selects the $\mathcal{K}$ more representative regions for a given problem. The number of selected regions, $\mathcal{K}$, must be large enough to distinguish relevant changes in the images, but not so large as to distinguish irrelevant variations such as noise [10]. These regions create a $\mathcal{K}$-dimensional Hilbert space $\mathcal{H}$, in which each region is now represented by a visual word [5].

*The authors thank the São Paulo Research Agency (FAPESP) for funding the research under the Awards 2010/05647-4 and 2009/16206-1.

Given a visual dictionary, we represent an image according to the visual words it contains. We map the original input image regions ϕ to a Hilbert space $\mathcal{H}$ represented by the calculated visual words. One of the main challenges we face in this new scenario is to create a representative dictionary that captures all the nuances of a given categorization problem.

The dictionary's creation requires the quantization of the description space, which can be done using clustering approaches, randomly, or sometimes, by calling in specialists to "select" the most important/representative words for a given problem.

2.1. Local Features

In order to represent the visual content of a given image, we find a set of points of interest in such images and characterize their surrounding regions. It is desired to choose scale-invariant interest points in order to achieve a representation robust to some possible image transformations. To do so, we can use several different approaches and two of the most commonly used are the *Speeded-Up Robust Features* (SURF) [11] and the *Scale-Invariant Features Transform* (SIFT) [1]. Both methods achieve high repeatability and distinctiveness. In this paper we focus on SIFT descriptors.

SIFT algorithm is one of the most robust methods under translations, scale and rotation transformations [1]. It has mainly four major steps:

1. **Scale-space extrema detection**: in which the algorithm searches for candidate points invariant to scale changes [1].
2. **Feature point localization**: Scale-space extrema detection produces some unstable keypoints and, in this stage, the algorithm strives for eliminating them. The remaining points are called points of interest (PoIs).
3. **Orientation assignment**: this stage assigns one or more orientations to each PoI based on local image gradient directions.
4. **PoI characterization**: in order to achieve invariance to rotation, this stage selects a patch around each PoI and rotates such patch toward the most frequent direction of the gradient.

2.2. Visual Vocabularies

As we mentioned earlier, SURF and SIFT are good low-level representative feature detectors. However, this distinctiveness power comes with a price: as these solutions are often designed for exact matching, they do not translate directly into good results for image classification in broad or even constrained domains.

In the broad-class categorization case, these approaches are not well suited for direct use. To preserve the distinctiveness power of such descriptors while increasing their generalization, we use the concept of visual vocabularies.

In the construction of a visual vocabulary, each region of PoIs becomes a visual "word" of a "dictionary". To solve an N-Way image categorization problem using visual dictionaries, we select and create a database of examples comprising training examples of each class of interest. In this training stage, we perform the localization of the interest points in all available images using either SIFT or SURF, and each image in the training ultimately generates a series of points of interest.

After finding the PoIs, we need to create the dictionary or codebook representing distinctive regions of each one of the classes of interest. For that, we need to choose the size (number of words) $\mathcal{K}$ of the dictionary. In this paper we tackle the problem of automatically finding $\mathcal{K}$ with no computational burden using a fast classifier named *Optimum Path Forest* (OPF) [7].

After selecting a meaningful number of words ($\mathcal{K}$) to create the dictionary, we can perform clustering such as K-Means for finding representative centers for the cloud of PoIs.

2.3. Training and Classification

From the training images, researchers in the literature usually create the visual dictionary using either random selection of words or clustering-based approaches.

After creating the dictionary, each of the training images' PoIs is assigned to the closest visual word of the dictionary. This step is known as *quantization*. In the end of the quantization process, we are left with a set of feature vectors representing the histogram of the selected visual words for each image.

To perform the final classification procedure, we select an N-way machine learning classifier. For training the classifier, we feed it with feature vectors calculated using the training images containing examples of each class of interest. Figure 1 depicts and summarizes the sequence of steps for categorizing images through the visual dictionaries point of view.

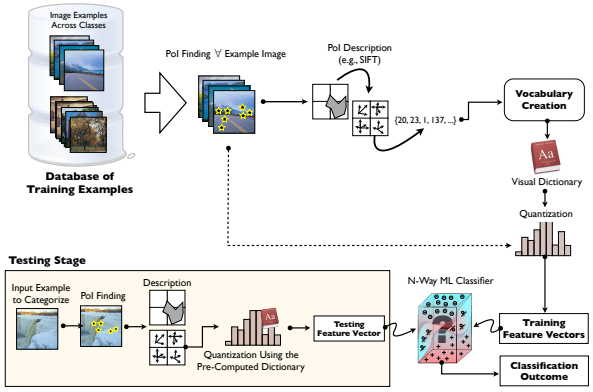

Fig. 1. Categorizing images using visual dictionaries.

3. AUTOMATICALLY FINDING THE NUMBER OF WORDS IN THE DICTIONARY

Our main contribution in this paper is the introduction of an approach for automatically finding the number of visual words that effectively represent a given number of image classes to categorize.

Unfortunately, several researchers still create the visual dictionary using a fixed number of words (e.g., 5,000) and need to find a classifier in the last stage robust to the curse of dimensionality. In this paper, we solve both of these problems. Using just a few training examples to probe the size of the dictionary, our classifier is robust to calculations using sparse feature spaces (result of the PoIs mapping to the dictionary).

Given a training set of images spanning N classes, we select less than 10% of the training images to probe for the size of the visual dictionary $\mathcal{K}$. Using 50% of the probe set, at iteration i we select $\mathcal{K}$ random visual words and train an OPF classifier using the same 50% of the probe set and test on the other 50%. If there is a performance gain with respect to the previous value of $\mathcal{K}$ at iterations $i-1$ or $i-2$, we increase the value of $\mathcal{K}$ for a small amount. The

increase on $\mathcal{K}$ can consider, for instance, the accuracy percentage gain times the number of classes. We repeat this process until there is no classification performance gain at a given iteration with respect to the the value of $\mathcal{K}$ at the two previous iterations. At the first iteration, we can define the minimum number of words $\mathcal{K}$ as a fraction of the the number of classes. The comparison of the performance gain with the last two calculated values of $\mathcal{K}$ is to diminish the impact of local minima on the classification results.

The advantage of using OPF classifier here is that:

- it is much faster for training than the competitor *Support Vector Machines* [7];
- its variance is not as high as the *K-Nearest Neighbors* classifier;
- it is not as sensitive to outliers when compared to other classifiers that require similar computational effort such as *K-Nearest Neighbors*;
- it can map the decision boundaries with just a few training examples; and
- it allows for the use of different (dis)similarity functions for comparing PoIs.

This automation using a low-cost approach is important for the success of dictionary-based categorization approaches since they must cope with millions of images in a common operational scenario.

The previous step is very fast and gives a very good candidate for $\mathcal{K}$. With $\mathcal{K}$ at hand, we must augment the visual words to the whole training set. Therefore, we now sample $\mathcal{K}$ words from the whole training set using either random selection or k-means clustering. Finally, each image in the training set is then mapped onto the new $\mathcal{K}$-sized visual dictionary generating N feature vectors which feed an OPF classifier in the last stage.

Given an input example to test, we map its PoIs onto the $\mathcal{K}$-sized visual dictionary and feed previous computed OPF classifier for categorization.

The Optimum Path Forest Classifier

OPF is a supervised pattern classification based on optimum-path forest [7] and, to our knowledge, this is the first time it is used for image categorization using visual words.

With OPF classifier, we model each sample of the data set as a graph node, which is connected to the other ones according to some predefined adjacency relation. The main idea concerns with choosing a small set of key samples (prototypes) in order to begin a competition process among them aiming at partitioning the graph into optimum-path trees (OPTs), each one of them rooted by its corresponding prototype.

An OPT may define one class or can be a part of a collection of OPTs with the same label to compose the more general class. Each prototype node tries to conquer the remaining samples of the graph by offering to them optimum-paths according to some path-cost function. When a sample is assigned to some prototype, it receives the class label of this prototype. Therefore, to design an OPF classifier, one only needs to specify three parameters: (i) adjacency relation, (ii) the methodology to estimate the prototype nodes, and (iii) the path-cost function. Thus, the OPF may not be considered a method, but a methodology to design graph-based pattern recognition algorithms.

In the OPF's training stage, the prototype nodes are chosen as the samples that belong to the region of the boundary of the classes, and are identified by calculating a *Minimum Spanning Tree* on the training set by marking the connected nodes with different classes. After that, each prototype sample tries to conquer the other ones by offering to them optimum paths computed as the maximum arc-weight along the path. The prototype that offers the optimum-path to a given node will be the one that conquers that sample. At the final of the process, one have an optimum-path forest, which is a collection of optimum-path trees rooted at each prototype.

To classify an input example, this example is added to the optimum-path forest generated in the previous step and connected to all nodes. OPF then evaluates which training node offers the optimum-path to the testing sample and assigns it the corresponding label.

4. EXPERIMENTS AND METHODOLOGY

4.1. Experimental Setup

In all experiments, we have performed a 5-fold cross-validation in order to assess how the results generalize to an independent data set [12]. All the experiments are carried out on an Intel i5 computer with 4Gb of RAM.

In addition, the two image data sets we use in the experiments are freely available through the Internet:

1. **Corel Relevants**. This data set comprises 1,624 images from Corel Photo Gallery. The collection contains 50 color image categories and is referred to as the Corel Relevant sets (RRSets – http://webdocs.cs.ualberta.ca/∼mn/BIC/);
2. **Darmstadt ETH**. This data set comprises 3,280 images of objects grouped into 80 equally-sized classes. The images portrait just one object each time and vary in pose and point of view. Some of the available classes are: apple, pear, cow, car, among others (http://tahiti.mis.informatik.tu-darmstadt.de/old-mis/Research/Projects/categorization/eth80-db.html);

4.2. Results

To validate our approach for finding the size $\mathcal{K}$ of a visual dictionary for an N-Way image categorization task, we need to analyze the behavior of random and k-means approaches for selecting the visual words varying $\mathcal{K}$ from 10 to 1,000 visual words with steps of size 10. For all the experiments we have used 5-fold cross validation in which, at each cross-validation stage, we use four folds for defining the visual dictionary and to train a classifier and one fold for testing the classifier. For the sake of comparison, the have used OPF and *K-Nearest Neighbor* classifier in this last stage. With a pre-defined $\mathcal{K}$ and its corresponding dictionary, OPF itself can be used as a classifier.

As we observe in Tables 1 and 2, there is a good cutting point in which is safe for selecting a near optimal value for $\mathcal{K}$ without the need for calculating the value of $\mathcal{K}$ using brute force.

Considering the ETH-80 data set, a good dictionary size would be $30 \leq k \leq 50$ leading to a classification accuracy of $\mu \cong 55\%$. However, we are only able to successfully point out this value of $\mathcal{K}$ if we have access to the complete curve of classification accuracies along with $\mathcal{K}$ different setups. Unfortunately, to compute such a curve, even in the simplified scenario of ten different values of $\mathcal{K}$ from 1 to a 1,000 words is computational intensive. If we reduce the step-size from 100 to 50 in order to be closer to a good value of $\mathcal{K}$, this scenario is even worse.

Table 2 also shows a good classification rate when the size of the dictionary is less than 250. However, this value has a small impact in the classification accuracy for the range of value $50 \leq k \leq 250$. These results are valid regardless the final classification method over

the calculated feature vectors (either OPF or the best performing nearest neighbor-based classifier, 3-NN).

Table 1. Performance curve for OPF and 3-NN classifiers using a visual dictionary calculated with K-Means and Random Selection for ETH data set. OPF and 3-NN here are used for each possible size of the dictionary. They are not being used to automatically probe such size.

# of visual words	OPF		3-NN	
	Random	K-Means	Random	K-Means
10	52.98%	52.29%	52.32%	51.59%
30	53.19%	55.22%	53.26%	54.25%
50	**53.84**%	**54.41**%	52.58%	53.67%
100	52.77%	53.95%	52.88%	52.93%
250	53.34%	52.45%	52.17%	51.81%
500	51.12%	50.98%	50.75%	50.57%
750	50.39%	50.33%	50.18%	50.34%
1000	50.18%	50.26%	50.14%	49.96%

Table 2. Performance curve for OPF and 3-NN classifiers using a visual dictionary calculated with K-Means and Random Selection for Relevants data set. OPF and 3-NN here are used for each possible size of the dictionary. They are not being used to automatically probe such size.

# of visual words	OPF		3-NN	
	Random	K-Means	Random	K-Means
10	52.69%	50.85%	51.43%	50.88%
30	52.98%	52.60%	51.48%	52.17%
50	53.78%	53.74%	53.41%	52.39%
100	53.32%	54.47%	52.12%	52.82%
250	**54.37**%	**54.41**%	53.17%	52.18%
500	53.40%	53.25%	52.80%	51.97%
750	52.52%	51.90%	51.52%	51.56%
1000	51.78%	51.58%	51.60%	50.92%

The proposed approach automatically selects the value of $\mathcal{K}$ without the need to compute the complete curve and achieves almost the same classification results much faster: less than two seconds to find a good value for the dictionary size $k \cong 42 \pm 19$. The proposed method requires less than a second for training with OPF and random selection of words after probing a good value for the dictionary size. If we use OPF and K-Means after probing a good value for the dictionary size, the method requires about 63 seconds for training while not losing accuracy (61% classification accuracy with OPF and K-Means against the best point $\cong 54\%$ using the brute-force method). Recall that for the curve, for each possible value of $\mathcal{K}$, we need to calculate a new visual dictionary and train a new classifier. Considering the brute-force, each point in the curve requires more than 10 seconds to find the dictionary while the complete procedure using our approach (probing the dictionary size and training the classifier) requires about one minute.

For the Relevants data set, we achieve similar results. Our approach automatically selects the value of $k \cong 78 \pm 11$ in less than three minutes. With this value of $\mathcal{K}$, the proposed method yields a classification accuracy of $\mu = 58.4\%$ which is in line with the best point in the calculated brute-force approach ($\mu = \cong 54.4\%$).

5. CONCLUSIONS

In this paper we presented an approach for automatically finding the size of a visual dictionary for categorization purposes. The approach probes the size of the dictionary using a few examples of the training set, trains a classifier with a dictionary using the probed size and yields a near-optimal dictionary in terms of feature discrimination, classification time and dictionary size.

The proposed solution uses a technique we call *Optimum Path Forest* (OPF) [7] which is itself a very fast classifier with small computational footprint when compared to traditional classifiers such as SVMs. In addition, the proposed solution favors the creation of smaller, yet discriminative, dictionaries with low classification variance. In addition, as the proposed solution is incorporated right in the process of probing the dictionary size, we are able to use select different (dis)similarity functions to compare the PoIs if needed.

The experiments results we presented show the proposed solution successfully find a good candidate for the dictionary size while preserving the classification accuracy and low-computational cost.

Finally, our future work consists of investigating the incorporation of the optimum-path forest information for pointing out the initial candidate words instead of random selection. We are also investigating other (dis)similarity functions to compare the PoIs since in this paper we have used L2 to compare them.

6. REFERENCES

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, February 2004.

[2] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *IJCV*, vol. 60, no. 1, pp. 63–86, 2004.

[3] M. Fatih Demirci, A. Shokoufandesh, Y. Keselman, L. Bretzner, and S. Dickinson, "Object recognition as many-to-many feature matching," *IJCV*, vol. 69, no. 2, pp. 203–222, 2006.

[4] J. Stoettinger, A. Hanbury, N. Sebe, and T. Gevers, "Do colour interest points improve image retrieval," in *ICIP*, 2007, pp. 169–172.

[5] E. Valle, *Local-descriptor matching for image identification systems*, Phd thesis, Universit de Cergy-Pontoise, Cergy-Pontoise, France, 2008.

[6] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE ICCV*, 2003, pp. 1470–1477.

[7] J. P. Papa, A.X. Falcão, and C.T.N. Suzuki, "Supervised pattern classification based on optimum-path forest," *IJIST*, vol. 19, no. 2, pp. 120–131, 2009.

[8] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *IEEE ICCV*, 2005, pp. 1800–1807.

[9] I. Ulusoy and C. M. Bishop, "Generative versus discriminative methods for object recognition," in *IEEE CVPR*, 2005, vol. 2.

[10] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and Cédric Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision*, 2004.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision*, 2006, pp. 1–14.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 1 edition, 2006.