String distance for automatic image classification

Nguyen Hong Thinh*, Le Vu Ha* , Barat Cecile** and Ducottet Christophe**

*University of Engineering and Technology, Vietnam National University of HaNoi, Vietnam
e-mail : {hongthinh.nguyen, havl }@vnu.edu.vn
**University Jean Monnet, Saint Etienne, France
e-mail : {cecile.barat, ducottet}@univ-st-etienne.fr

Abstract

The Bag-of-visual Words (BOW) model has recently become the most popular representation to depict image content. It has proven to be quite effective for many multimedia and vision applications, especially for object recognition and scene classification or automatic image annotation. This model however ignores the spatial layout of features within images, which is yet discriminative for category classification. In this paper, we present a novel approach based on string matching to take into account geometric correspondences between images and facilitate category recognition. First, we propose to represent images as strings of histogram second, we introduce a new string distance in the context of image comparison. This distance automatically identifies local alignments between sub image regions and allows merging groups of similar sub-regions. Experiments on several dataset such as Scene-15, Caltech-101 and Pascal 2007 show that the proposed approach outperforms the classical BOW method and is competitive with state-of-the art techniques for image classification.

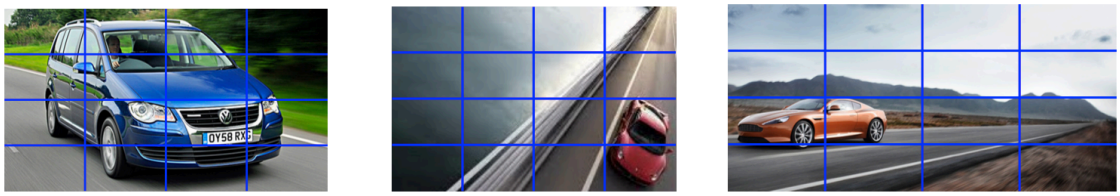
Keywords: Image classification, String distance, Region matching, spatial information

1. INTRODUCTION

The exponential increasing of the number of images requires efficient ways to classify them based on their visual content. The most successful and popular approach is the Bag of visual Word (BOW) [1-2] representation due to its simplicity and robustness. Unfortunately, this approach fails to capture the spatial image layout, which plays important roles in modeling image categories. Recently, Lazebnik et al [3] introduced the Spatial Pyramid Representation (SPR), which successfully incorporated spatial information into the BOW model. The idea of their approach is to split the image into a pyramidal grid and to represent each grid cell as a BOW. Assuming that images belonging to the same class have similar spatial distributions, it is possible to use a pairwise matching as similarity measurement. However, this rigid matching scheme prevents SPR to cope with image variations and transformations (as seen in Figure 1). In this paper, we intend to relax the exactness of rigid matching problem to improve object or scene classification by using string-matching principle.

For this purpose, we propose to model images as strings of ordered BOW and we present a new string distance specifically adapted to strings of histograms in the context of image comparison. The standard edit distance [5] between two strings is defined as the minimum number of edit operations, i.e. insertion, deletion, substitution, that are required to transform one string to the other.

Figure 1: Rigid matching problem where region-to-region matching is based on its position, then it introduce a lot of bad matching due to object variations and transformations.



(a) (b) (c)

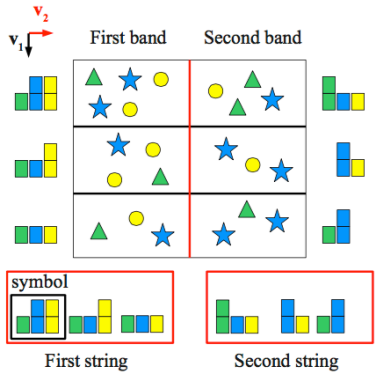
Our string distance variant supports a new edit operation, called merge operation, which combines two or more sub-regions from one image and compares this combination with one or several sub-regions of the second image. This distance automatically identifies local alignments between sub-regions or groups of similar sub-regions in images. With the proposed approach, the number of sub-regions for different images may vary and is adjusted according to the visual content, which brings flexibility to the matching process. We validate our approach on two well-known datasets: Scene-15, Caltech-101 and Pascal 2007.

The rest of the paper is organized as follows. Section 2 explains how to present an image as a string of histogram. In Section 3, we introduce the new string distance appropriate to strings of histograms. Section 4 describes the experiments and results obtained. Section 5 concludes the paper.

2. IMAGE REPRESENTATION AS STRING OF HISTOGRAM

In this section, we discuss how to represent an image as ordered strings of regions preserving spatial relationships of those regions. Our image representation model intends to encode spatial relationships between local features and we focus on the local histogram based representation. An image is first tiled into regions and each local region is described as a histogram of visual words. After partitioning an image into a regular grid of sub-regions, each one is described using the classical BOW model. The sub-regions are then organized as a set of strings of ordered BOW. In images, there exists a natural sequencing of objects or entities within objects. It is possible to find a principal direction along which the projection of local features may convey information about the image context or capture the essence of the form of an object. Intuitively, in natural scenes, vertical or horizontal directions can plausibly describe relationships among local features. For instance, the sky is above trees, and trees are above grass. For object images, as proposed in the major axis of an object can be obtained from the first principal component in a principal component analysis. Distribution of local features along this major axis is similar whatever the orientation or scale the object is. Our idea is to build strings from images in order to describe the intrinsic order of features in a given direction First, we choose an orthogonal basis {v1, v2} that may best describes the image content. We divide an image into x bands of same width along direction v2. Then, each band is subdivided into n subregions of same size, along direction v1. For each band, traveling along v1 provides an ordered string of subregions. An image is finally represented with a set of strings of subregions, the number of strings being equal to the number of bands. This idea is shown on Figure 2.

Figure 2: Image representation as two strings regions.



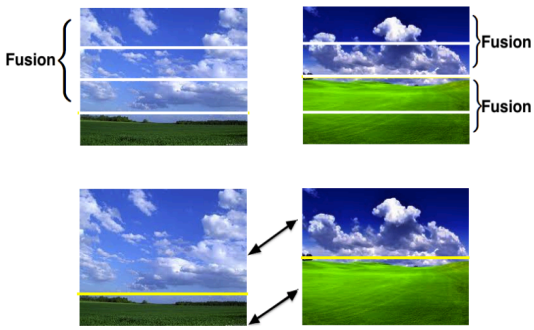
3. NEW STRING DISTANCE FOR IMAGE COMPARISON

The underlying idea of this section is to define a new distance to provide a better matching between two strings of histograms than a classical 2-by-2 comparison. This new distance supports merging operations between local histograms representing different sub-regions of the image. After formulating the new distance as an edit distance, we derive an efficient algorithm to compute it. Then, we define a specific intersection kernel well suited to image classification.

The classical approach to compare two images represented as a set of local histograms is to compare, two by two, corresponding histograms in the two images. The resulting distance is defined as the sum of distances between histograms of corresponding regions. In that case, the matching is rigid and depends on the position of the regions. If the visual elements of the two images are not located in the same regions, this could result in a high distance due to a bad matching between the regions.

The principle of our approach is to allow regions to be merged, separately on each image, depending on the visual content of the regions. When two regions are merged, the resulting histogram is the sum of the two initial ones. Then, depending on merge operations made within each string, one single region of one image may be matched to a set of merged regions in the second image resulting in a better matching between the two images. For example, in Figure 3, regions 1,2 and 3 of image 1 and regions 1 and 2, 3 and 4 of image 2 are merged. After merging, the combination regions are matching, and these matching are correct because they have same visual content.

Figure 3: Matching between regions with new merge operation



To compute the distance, we use the idea of edit distance. The distance also referred to the Levenshtein distance is designed to compare two strings of symbols representing for example two words of a given language. It considers three basic edit operations: the insertion of a new symbol, the deletion of an existing symbol and the substitution of one symbol into another. Compared to the standard edit distance, our distance does not use any deletion nor insertion. Nevertheless, let's first remark that in the standard edit distance, an insertion of a symbol in the first string followed by a substitution by an identical symbol in the second string is equivalent to a deletion of the symbol in the second string. Then, both delete and insert operations can be seen as the same delete operation made either in the first or in the second string. We can also remark that the merge operation of symbols x_{n} and x_{n +1} can be seen as the deletion of symbol x_{n} followed by a modification of symbol x_{n+1} which is assigned to the result of the merge operation.
In our approach, symbols are represented as local BOW. Thus, edit cost functions can be derived as standard histogram distance. Furthermore, compared to the standard edit distance, we must also take care that the symbols are modified during merge operations. More precisely, each merge operation generates a new symbol obtained by adding the histogram of the current symbol to that of the next one. This new symbol is then used for each subsequent operation.

4. EXPERIMENTS

In this experiment, we aim to compare the performance of our new distance with BOW baseline. Three datasets, 15 Scene, Caltech 101 and Pascal 2007 are used for evaluation. Here, 15 Scene is a scene dataset, which contains 4485 images of 15 classes. Caltech 101 is object dataset, which contains 9144 images. Each image has only one object but can be in different viewpoints and scales. Pascal 2007 is the most challenge dataset in three datasets. It contains more than 10000 images of 20 classes. Each image can contain one or more objects of different scales, colors and viewpoints.

To setup the experiment, local features of all the images are extracted by using dense SIFT descriptor on regular grid 16x16 pixels and with the step size 8 pixels. The k-means clustering approach is applied on a sub set of descriptors to create the visual codebook. The codebook size is set to K = 100. We use both hard coding (which means we assign each image feature to only one word) and sparse coding (we use sparse dictionary and sparse coding to assign each image feature to a few words in order to optimize image representation). We report the classification accuracy using both hard assignment coding and sparse coding for Caltech 101 and 15 Scene dataset. For Pascal 2007, only results of using hard coding are shown.

Table 1 shows classification result with 15 Scene dataset. To evaluate the classification performance, we use the same train/test setup as [2] which are: 100 images per class for training and the rest of image for testing. Our new proposed distance has shown outperform with the BOW baseline, for both hard coding and sparse coding, confirms the effectiveness of the combination of similar regions inside the images.

Table 1. Result for 15 Scene dataset

Coding	Baseline	Our proposed
Hard coding	75.48 +/- 0.55	83.16 +/- 0.77
Sparse coding	78.79 +/- 0.59	85.33 +/- 0.76

Table 2. Result for Caltech 101 dataset

Coding	Baseline	Our proposed
Hard coding	58.66 +/- 0.42	66.90 +/- 0.86
Sparse coding	57.10 +/- 0.86	73.92 +/- 0.91

Table 2 shown results on Caltech 101. We randomly select 30 images per class for training and maximum 50 images per class for testing. We do10-folds cross validation and report the average accuracy. Again, in both cases of coding method, the new distance outperforms the BOW baseline.

The Pascal 2007 dataset consist of 9963 images of 20 classes, which is an extremely challenging one due to variation of object scales and posed. The classification performance is evaluated using Average Precision (AP), a standard metric used by PASCAL challenge. We also keep the setup for training/testing set of the challenge: we train on 5011 images and test on 4952 images. The classification performance of the new distances and BOW baseline are shown on Table 3

Table 3. Result for Pascal 2007 dataset

Coding	Baseline	Our proposed
Hard coding	37.82 +/- 0.6	44.20 +/- 0.58

5. CONCLUSION

This paper has presented a new merge-based string matching edit-distance. Thanks to the new merge edit operation, the proposed distance has shown improvement over baseline method. However, the computational cost is still high. This problem has not been solved yet. Hopefully, using GPU in near future could solve it.

ACKNOWLEDEMENT

This work has been supported by VNU University of Engineering and Technology under Project CN.15.03

References

[1] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In Workshop on statistical learning in computer vision, ECCV, volume 1, pages 1–2

[2] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 1470–1477. IEEE.

[3] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spa- tial pyramid matching for recognizing natural scene categories. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, pages 2169–2178. IEEE

[4] Boureau, Y.-L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 2559–2566. IEEE

[5] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. In Soviet physics doklady, volume 10, page 707