WEIGHTED BAG OF VISUAL WORDS FOR OBJECT RECOGNITION

Marco San Biagio[1], Loris Bazzani[1,2*], Marco Cristani[1,2], Vittorio Murino[1,2]*

[1] Pattern Analysis & Computer Vision, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy
[2] Departimento di Informatica, Università di Verona, Strada le Grazie 15, 37134, Verona, Italy

ABSTRACT

Bag of Visual words (BoV) is one of the most successful strategy for object recognition, used to represent an image as a vector of counts using a learned vocabulary. This strategy assumes that the representation is built using patches that are either densely extracted or sampled from the images using feature detectors. However, the dense strategy captures also the noisy background information, whereas the feature detection strategy can lose important parts of the objects. In this paper we propose a solution in-between these two strategies, by densely extracting patches from the image, and weighting them accordingly to their salience. Intuitively, highly salient patches have an important role in describing an object, while those with low saliency are still taken with low emphasis, instead of discarding them. We embed this idea in the word encoding mechanism adopted in the BoV approaches. The technique is successfully applied to vector quantization and Fisher vector, on Caltech-101 and Caltech-256.

Index Terms— object recognition, dictionary learning, visual saliency, feature weighting

1. INTRODUCTION

Visual object recognition is one of the most studied problems in computer vision. It is challenging due to the high variability of the object appearance, the complex background, the illumination and viewpoint changes, the non-rigid deformations, the intraclass variability and other visual properties. A major effort have been spent in the last 20 years to engineer or learn an object *representation* that is invariant to these nuisances (e.g., [1, 2, 3]). A recent and comprehensive review can be found in [4].

The BoV approach [1] is widely accepted as standard technique to describe the object appearance from images. The idea is to build/learn a visual dictionary of patches (or their descriptor, such as SIFT [5]), and represent the image as a vector of counts of the number of elements associated to each word of the dictionary. The main advantages of this technique are the robustness to spatial translations of features, the efficiency to compute it, and its competitive performance in different image categorization tasks.

* Marco San Biagio and Loris Bazzani contributed equally to this work.

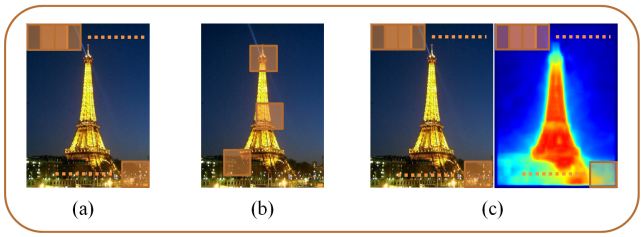

Fig. 1. (a) Dense features, (b) feature detector and (c) dense feature + salience.

The BoV representation considers either a dense grid of patches in the image [2] (Fig. 1 (a)) or a sparse set of patches selected by a feature point detector [5] (Fig. 1 (b)). However, the former approach can inject into the representation noisy information such as the background and other irrelevant objects or clutter. The latter approach may discard useful information that may be relevant for recognition. Furthermore, it often involves the tuning of a threshold parameter for keeping the most salient features.

In this paper, we propose a method that is a trade-off between the two approaches described above. The BoV representation is built using a grid of patches, where each patch generates a weight corresponding to its saliency, depending on the adopted saliency detection algorithm (Fig. 1 (c)). Intuitively, patches that are less salient are still considered but with low importance, instead of discarding them. In this way, the proposed method 1) considers all the patches of the image with different importance and 2) does not discard any information that may be relevant for the object recognition task. We show that the proposed technique can combine the salience map with different existing encoding methods: Vector Quantization (VQ) [2] and Fisher Vector (FV) [6]. Furthermore, our approach can be easily extended to other encoders, such as locality-constrained linear coding [7], and VLAD [8].

We tested our approach to two challenging public object recognition datasets, namely, Caltech-101 [9] and Caltech-256 [10]. In all the cases, we show that the proposed method is beneficial, leading to an improvement in terms of classification accuracy, with respect to classical encoding schemes.

The rest of the paper is organized as follows. Sec. 2 re-

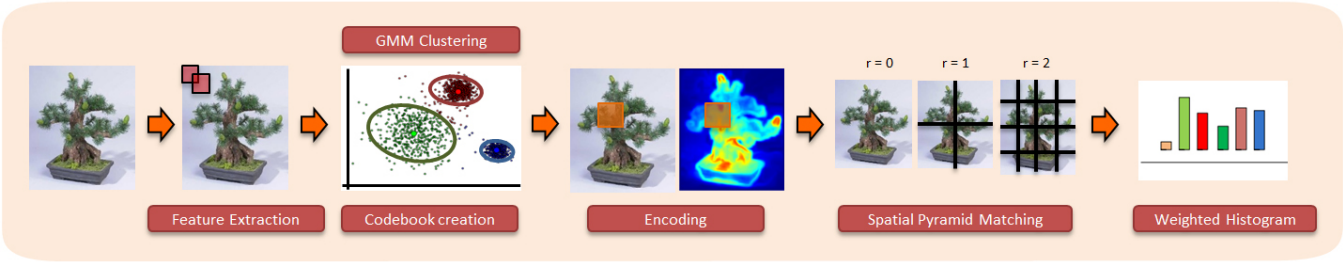

Fig. 2. Proposed object recognition pipeline.

ports the related work. Sec. 3 and Sec. 4 present our object recognition pipeline and the proposed technique, respectively. In Sec. 5, we report the classification results. In Sec. 6, we draw the conclusions and discuss the future work.

2. RELATED WORK

Many works are focused on proposing novel descriptors [11, 12] and/or encoders [7, 8, 13] with the goals of building a robust representation and obtaining state-of-the-art performance. Other papers investigate novel learning frameworks [14, 15, 16] that are engineered to perform the best on object recognition.

On the other hand, a lot of research has also been devoted to propose efficient and robust methods to extract object-generic, general-purpose salience maps extracted from the image. See [17] for a recent qualitative and quantitative analysis of techniques of saliency extraction. To the best of our knowledge, most of the existing BoV methods do not consider the importance of different patches in the image. We therefore propose to inject the salience of each patch in the encoder of the BoV method.

Our approach has some connections with [18], where the BoV vector is weighted by a context-independent salience score exploiting the segmentation between foreground and background. However, the foreground segmentation is itself a challenging problem in most of the object recognition datasets, where the background is usually cluttered. In [19], the salience is class-dependent and learned on training data. We instead use a bottom-up salience map [20] that is built from features and therefore is class-independent.

In [21], a fixed number of salient points are detected using a salience score, given by a principle of occurrence-based contextual salience: a code is salient if its presence cannot be inferred by other codes. Moreover, spatial information is embedded into salience. In contrast with the proposed approach, features that might be relevant for classification are discarded.

3. THE OBJECT RECOGNITION PIPELINE

The proposed pipeline is depicted in Fig. 2. In the first step, a grid of pixel locations with spacing of 4 pixels in both x,y directions is defined on the image. Around these pixel locations, patches of different sizes (12×12, 18×18, 24×24, 30×30 pixels) are extracted. On each patch, a SIFT descriptor [5] is calculated ("Feature Extraction" block in Fig. 2), generating a set of local descriptors for each image. In the "Codebook creation" step, the local descriptors are used to generate a codebook with K words, by Gaussian Mixture Model (GMM) clustering (usually using a subset of all the images' descriptors).

At this point, each descriptor of a given image is quantized into a *weighted* code, considering the most similar visual word of the codebook, and exploiting a saliency map [20], see the details in the next section (third step of Fig. 2). This results in a weighted histogram, where the height of each bin depends on the number of associated codes retrieved in the image, and on their related weights. This is called a Weighted Bag of Visual words (Weighted BoV). Note that previous works either consider all the patches without giving different importance to them (e.g., [2]) or selecting a subset of patches with a feature point detector (e.g., [5]).

In the next step, the image is partitioned into increasingly finer spatial sub-regions and Weighted BoV are computed from each sub-region, following a spatial pyramid scheme [2] ("Spatial Pyramid Matching" block in Fig. 2). Typically, $2^r \times 2^r$ sub-regions, with $r = \{0, 1, 2\}$ are used, as shown in Fig. 2. All the Weighted BoV extracted from each sub-region are pooled and concatenated together, generating the final Weighted BoV representation of the image. Finally, a one-vs-all linear SVM is used for classification.

4. WEIGHTED BAG OF VISUAL WORDS

The proposed approach acts on the encoding step by including additional information that will guide the exploration of the image, that is the salience of each patch. In this way, each patch is evaluated with more importance in the case that it is relevant or salient, and it has less weight in the opposite case.

In this section, we present the proposed method applied to two specific encoding schemes: VQ [2] and FV [6]. We named the two methods weighted VQ and weighted FV. Note that many other encoding methods fit the proposed idea due to its generality, *e.g.*, VLAD [8], LLC [7] and others.

Let us consider a vocabulary of K words, resulting from the clustering of some training feature vectors $\{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{x}_i \in \mathbb{R}^D$. In practice, the words are represented by K exemplars forming the set $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$, $\mu_k \in \mathbb{R}^D$ that is, the cluster centroids. In our experiments, we use the Expectation-Maximization algorithm for GMM on the SIFT feature vectors.

Let us assume that an image I can be decomposed in a grid of N_I patches, whose corresponding feature vectors are $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_I}\}$. Each patch is associated to a weight $\{\alpha_1, \alpha_2, \ldots, \alpha_{N_I}\}$. In the present work, we extracted a bottom-up salience map [20] from the image, and weights are defined as the sum of the pixel salience values inside each patch, normalized by its size. Given the generality of the proposed idea, any other method to compute the saliency of an image can be used, such as objectness [22].

Weighted VQ encodes a set of feature vectors extracted from an image by associating each element to the closest word in the vocabulary, where the association is weighted by the corresponding α_i. More formally, we define the bag of visual feature representation as a vector $\mathbf{v} = [v_1, v_2, \ldots, v_K]$ where:

$$v_k = \sum_{i=1}^{N_I} \alpha_i \, \delta(\mathbf{x}_i, \mu_k) \tag{1}$$

where δ is equal to 1 if the feature vector $\mathbf{x}_i$ is associated through a nearest neighborhood policy to the μ_k, 0 otherwise. Notice that if we set the α_i to be always 1, we obtain the standard VQ.

FV is an extension of VQ where first and second order statistics are also considered. Let us assume to have also a covariance matrix Σ_k associated to each word in the dictionary, that can be easily retrieved when using GMM for clustering. As in the original paper [6], we assume to have diagonal covariance matrices $\Sigma_k = \text{diag}(\sigma_k^d)$. The likelihood function defined in the weighted Fisher vector is the following

$$\mathcal{L}(\mathbf{X}|\lambda) = \sum_{i=1}^{N_I} \alpha_i \log \sum_{k=1}^{K} w_k p_k(\mathbf{x}_i|\mu_k, \Sigma_k) \tag{2}$$

where $\lambda = \{w_k, \mu_\mathbf{k}, \Sigma_k\}_{k=1}^{K}$ are the parameters of the mixture and $p_k(x|\mu_\mathbf{k}, \Sigma_k)$ is a Gaussian distribution.

In the same spirit of [6], we derived the following equations that build the weighted Fisher vector:

$$\frac{d\mathcal{L}(\mathbf{X}|\lambda)}{d\mu_k^d} = \sum_{i=1}^{N_I} \alpha_i \gamma_i(k) \left[\frac{x^d - \mu_k^d}{(\sigma_k^d)^2}\right] \tag{3}$$

$$\frac{d\mathcal{L}(\mathbf{X}|\lambda)}{d\sigma_k^d} = \sum_{i=1}^{N_I} \alpha_i \gamma_i(k) \left[\frac{(x^d - \mu_k^d)^2}{(\sigma_k^d)^3} - \frac{1}{(\sigma_k^d)^2}\right]. \tag{4}$$

The weighted Fisher vector is the concatenation of $\frac{d\mathcal{L}(\mathbf{X}|\lambda)}{d\mu_k^d}$ and $\frac{d\mathcal{L}(\mathbf{X}|\lambda)}{d\sigma_k^d}$ for each k and d. Note that when the αs are equal to 1 we have the original FV.

METHOD	VQ		FV
# OF WORDS	128	600	128
Weighted BoV	$\mathbf{65.16}\%$ (± 1.40)	$\mathbf{68.23}\%$ (± 1.05)	$\mathbf{66.57}\%$ (± 0.67)
BoV	63.93% (± 1.37)	67.15% (± 0.96)	64.63% (± 0.53)
Keypoint BoV	51.62% (± 1.31)	50.72% (± 1.91)	50.52% (± 0.61)

Table 1. Results obtained on the Caltech-101 using 15 training example for each class.

5. EXPERIMENTS

The proposed approach was tested on the Caltech-101 [9] and Caltech-256 [10] datasets. We followed the validation procedure proposed in [16], using their available code [23].

Caltech-101 [9] represents a key benchmark for the object recognition community. It consists of 102 classes (101 object categories plus background). The significant variations in color, pose and illumination inside each of the 101 classes make this dataset very challenging. The number of images per class ranges from 31 to 800 and most of them are at medium resolution, roughly 250×280 pixels.

Using the framework described in Sec. 3, we follow the common experimental setup, namely, we randomly chosen 30 per-class images and subsequently split into 15 for training and 15 for testing. Five different random partitions are considered and the average results with standard deviations are reported. For a fair comparison, while varying the number of training images, we keep constant the set of descriptors from which we extract the codebook and the codebook itself, for each method.

In Table 1, we report classification rates for different dimensions of the vocabulary and different methods (VQ and FV[1]). The first row shows the proposed weighted bag of feature approach. The second row and third row report the results using bag of feature with a dense grid of patches (called BoV) and a sparse set of patches given by the SIFT detector (called keypoint BoV), respectively. It is easy to notice in Table 1 that weighting the BoV representation is always beneficial for both VQ an FV. In average, the proposed method outperforms the standard BoV and the keypoint BoV of about 1.94% and 16.05% (for FV), respectively.

We extend our analysis also to Caltech-256 [10] that consists of 257 classes (clutter class included) with a minimum of 80 images per class and a total number of images equal to 30607. It represents much higher variability in object size, location, pose and lighting conditions than in Caltech-101. We followed a similar experimental setup as the Caltech-101. We train our system on $\{5, 10, 15, 20, 25, 30\}$ images per class and test on 15 images, in 5 random splits each.

[1] For the FV-based methods, we found that one level of the spatial pyramid is enough to obtain the best results.

METHOD	Encoder	5	10	15	20	25	30
Weighted BoV	FV	**22.28%** (±0.93)	**30.31%** (±0.86)	**35.08%** (±0.62)	**38.23%** (±0.75)	**40.25%** (±0.73)	**42.39%** (±0.25)
Weighted BoV	VQ	*21.14%* (±0.28)	*28.35%* (±0.67)	*32.19%* (±0.41)	*35.09%* (±0.59)	*37.09%* (±0.36)	*38.85%* (±0.59)
BoV	VQ	20.80% (±0.47)	27.92% (±0.72)	31.88% (±0.68)	34.40% (±0.27)	36.69% (±0.62)	38.32% (±0.5)
Keypoint BoV	VQ	12.07% (±0.17)	15.81% (±0.38)	18.41% (±0.54)	20.08% (±0.79)	21.32% (±0.49)	22.44% (±0.51)

Table 2. Results on Caltech-256 of the proposed method (first two rows) and the two baselines on different sizes of the training set (number of examples per class).

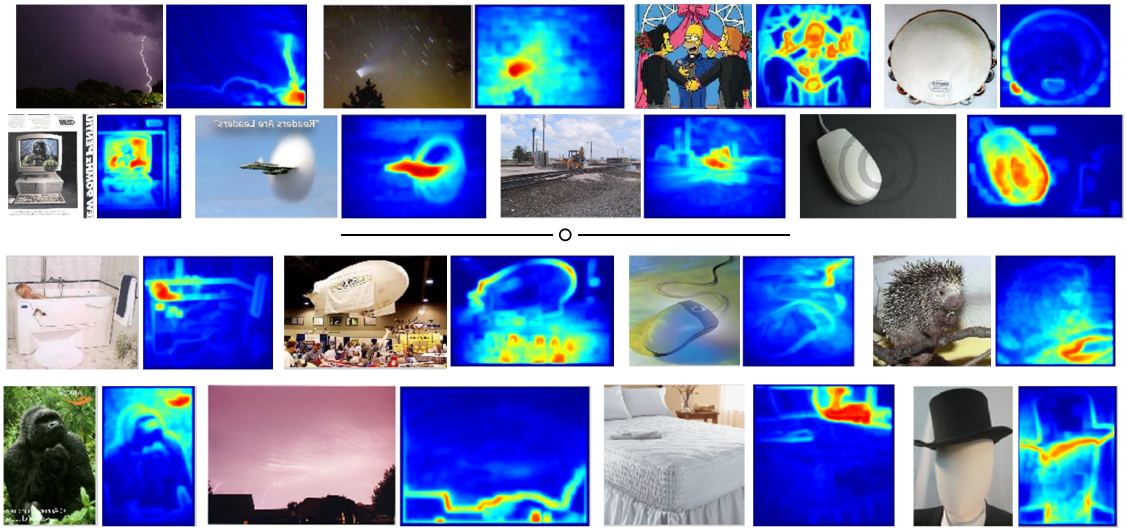

Fig. 3. Correctly-classified (top two rows) and miss-classified images (bottom two rows) evaluated by the salience pipeline.

Table 2 shows the results of the proposed method (first two rows) against the baselines (last two rows) increasing the number of training examples per category. We obtained the best results with the proposed method using the FV encoder with an improvement that goes from 1.48% to 4.07% with respect to the BoV. The improvement with respect to the keypoint BoV is even bigger (from 14.50% to 25.95%). Notice that also the weighted BoV based on the VQ encoder is always better than the two baselines.

To better understand the proposed method, we reported in Fig. 3 (top two rows), pairs of images of Caltech-256 (and the associated salience maps) correctly classified using the weighted BoV approach but miss-classified with the classic BoV. In Fig. 3 (bottom two rows), we also reported some examples miss-classified using the weighted BoV representation but classified correctly using the classic BoV. This qualitative analysis shows that the weighted BoV representation is better whether the salience map highlights regions of the object instance to be classified.

As further experiments, we considered also the Pascal VOC 2007 dataset [24]. We found out that the results for all the tested techniques were similarly low. This suggests that there is the need of more complex objects model and classifiers. We leave this as future work, instead here we focused on showing the net value of weighting visual words with saliency scores.

6. CONCLUSIONS

Modeling the object appearance for object recognition is a challenging task especially in cluttered images. In this paper, we proposed a novel idea to weight, in a different way, the elements that compose the descriptor used for the recognition task. This method is general because it can be applied to any approach based on the bag of visual words model. In practice, the proposed method is a trade-off between using a dense grid of patches and a sparse set of detected keypoints. We showed that the proposed approach outperforms the bag of visual words method with both dense and sparse patches on two challenging datasets, namely, Caltech-101 and Caltech-256. Future effort will be spent extending the idea to other existing encoders and investigating the role of other salience and objectness methods, and considering scene recognition scenarios.

7. REFERENCES

[1] R. Grzeszick, L. Rothacker, and G. A. Fink, "Bag-of-features representations using spatial visual vocabularies for object classification," in *ICIP*, 2013.

[2] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*. 2006, pp. 2169–2178, IEEE Computer Society.

[3] Y. Bengio and A. Courville, "Deep learning of representations," in *Handbook on Neural Information Processing*, pp. 1–28. Springer, 2013.

[4] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward," *CVIU*, vol. 117, no. 8, pp. 827 – 891, 2013.

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

[6] F. Perronnin and C. R. Dance, "Fisher kernels on visual vocabularies for image categorization.," in *CVPR*. 2007, IEEE Computer Society.

[7] J. Wang, J. Yang, K. Yu, and F. Lv, "Locality-constrained linear coding for image classification," *CVPR*, vol. 1, no. January, 2010.

[8] R. Arandjelović and A. Zisserman, "All about vlad," in *CVPR*, 2013.

[9] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *CVPR*, vol. 12, pp. 178, 2004.

[10] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep. 7694, California Institute of Technology, 2007.

[11] K.E.A. van de Sande, T. Gevers, and C.G.M. Snoek, "Evaluating color descriptors for object and scene recognition," *PAMI*, vol. 32, no. 9, pp. 1582 –1596, sept. 2010.

[12] M. San Biagio, M. Crocco, M. Cristani, S. Martelli, and V. Murino, "Heterogeneous auto-similarities of characteristics (hasc): Exploiting relational information for classification," in *ICCV*, 2013.

[13] A. Bergamo, L. Torresani, and A. W. Fitzgibbon, "Picodes: Learning a compact code for novel-category recognition," in *Advances in Neural Information Processing Systems*, 2011, pp. 2088–2096.

[14] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, "Group-sensitive multiple kernel learning for object categorization," in *ICCV*, 2009.

[15] M. H. Quang, L. Bazzani, and V. Murino, "A unifying framework for vector-valued manifold regularization and multi-view learning," in *ICML*, 2013, pp. 100–108.

[16] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV*, September 2009, pp. 606 – 613.

[17] A. Borji, H. R. Tavakoli, D. N. Sihite, and L. Itti, "Analysis of scores, datasets, and models in visual saliency prediction," in *ICCV*, Oct 2013.

[18] R. de Carvalho Soares, I.R. da Silva, and D. Guliato, "Spatial locality weighting of features using saliency map with a bag-of-visual-words approach," in *ICTAI*, 2012, vol. 1, pp. 1070–1075.

[19] M. Marszaek and C. Schmid, "Spatial weighting for bag-of-features," in *CVPR*. IEEE, 2006, vol. 2, pp. 2118–2125.

[20] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *PAMI*, vol. 34, no. 10, pp. 1915–1926, Oct. 2012.

[21] D. Parikh, C. L. Zitnick, and T. Chen, "Determining patch saliency using low-level context," in *ECCV*, pp. 446–459. Springer, 2008.

[22] A. Bogdan, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *PAMI*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[23] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," 2008.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge 2007 results," .