ENCODING STRUCTURAL SIMILARITY
BY CROSS-COVARIANCE TENSORS
FOR IMAGE CLASSIFICATION

MARCO SAN BIAGIO*,‡, SAMUELE MARTELLI*§,
MARCO CROCCO*,¶, MARCO CRISTANI*†,||
and VITTORIO MURINO*†,**

*Pattern Analysis & Computer Vision — Istituto Italiano di Tecnologia
Via Morego 30, 16163 Genova, Italy
‡marco.sanbiagio.iit.it
§samuele.martelli.iit.it
¶marco.crocco.iit.it
||marco.cristani.iit.it
**vittorio.murino.iit.it

Received 20 February 2014
Accepted 18 July 2014
Published

In computer vision, an object can be modeled in two main ways: by explicitly measuring its
c characteristics in terms of feature vectors, and by capturing the relations which link an object
with some exemplars, that is, in terms of similarities. In this paper, we propose a new similarity-
based descriptor, dubbed structural similarity cross-covariance tensor (SS-CCT), where self-
similarities come into play: Here the entity to be measured and the exemplar are regions of the
same object, and their similarities are encoded in terms of cross-covariance matrices. These
matrices are computed from a set of low-level feature vectors extracted from pairs of regions
that cover the entire image. SS-CCT shares some similarities with the widely used covariance
matrix descriptor, but extends its power focusing on structural similarities across multiple parts
of an image, instead of capturing local similarities in a single region. The effectiveness of SS-
CCT is tested on many diverse classification scenarios, considering objects and scenes on widely
known benchmarks (Caltech-101, Caltech-256, PASCAL VOC 2007 and SenseCam). In all the
cases, the results obtained demonstrate the superiority of our new descriptor against diverse
competitors. Furthermore, we also reported an analysis on the reduced computational burden
achieved by using and efficient implementation that takes advantage from the integral image
representation.

Keywords: Object recognition; scene classification; covariance; cross-covariance.

†Corresponding author.

1460008-1
1. Introduction

The modeling of an “object” in computer vision can be pursued by adopting two main paradigms: feature-based and similarity-based. The former aims at encoding an object by collecting and storing in a given descriptor visual cues such as color or more complex visual information (e.g. scale-invariant feature transform (SIFT),\textsuperscript{9} histogram of oriented gradients (HOG),\textsuperscript{3} local binary pattern (LBP),\textsuperscript{19} to quote some). In the latter case, the goal is to extract stable relations which characterize an object class with respect to a set of models or exemplars.\textsuperscript{1,8}

The similarity-based paradigm can be naturally extended to the concept of self-similarity, where the roles of “entity to be measured” and “exemplar” are shared among the parts of an image. In the simplest form, self-similarity can be evaluated among neighboring pixels,\textsuperscript{14} eventually estimating bag of self-similarities to compactly describe an entire image.

While the self-similarity relation is computed on top of feature descriptors (and not based on the raw pixel values), we propose an approach which fuses together the two paradigms, potentially joining their advantages. An explicative example of such idea, applied to the pedestrian detection task, can be found in the SST model\textsuperscript{10}: here the feature-based descriptors are the HOG features, which describe image regions whose similarities are estimated by Euclidean pairwise distances. This approach, even if it shows good performance and fast calculation, suffers of practical problems; the most critical issue is the alignment of the entities, which is a requirement that, especially in the case of structured objects, is hard to be satisfied. In fact, with misaligned parts, Euclidean distances are computed between diverse regions, failing to capture the visual structure of an object. In addition, the relations between parts are collapsed into scalar values (a vector), which become unstable in the case of high intra-class variations, or even worse, when in presence of noisy conditions. Our aim in this work is to overcome these limitations.

Our solution substitutes the analysis of linear distances between regions with their covariances, no more analyzing how similar the two regions are, but analyze how they correlate considering a particular low-level feature. This leads to a richer description of the local similarity between parts of an object.

Covariances of low-level features, in the form of covariance matrices, bear several advantages when used as single region descriptors, as pointed out in Refs. 12, 15 and 16; actually, they provide a natural way of fusing multiple features that might be correlated. Due to the analysis of statistics of pixels values, instead of considering a single gray value, the per-pixel noise management is more effective: Pixels values affected by clutter are filtered out with the average operation intrinsic to the covariance calculation. For the same reason, covariance matrices exhibit a certain robustness against scale change, since their calculation do not depend on the number of elements used to build it. Finally, when compared to other statistical descriptors, such as multi-dimensional histograms, covariances are intrinsically low-dimensional as their size is only $O(N^2)$, with $N$ being the number of features.
So far, covariances of low-level features have been used to describe single entities (images, regions, etc.). The original contribution here is to employ covariances to measure statistical similarities across different entities, in this case, different image regions. For this reason, covariance matrices have been properly generalized with the cross-covariance matrices, which capture the variations among two generally different feature vectors.

In particular, the structural similarity cross-covariance tensor (SS-CCT) is proposed here, encoding all the similarities among regions by means of Cross-Covariance matrices, each one capturing all the pairwise relationships between the single features extracted in a given couple of regions. The SS-CCT inherits the versatility of the covariance; furthermore, other than the advantages listed above, relations among regions can be encoded when these are modeled by any state-of-the-art descriptor: HOG, SIFT, LBP, etc. This represents one of the most important differences with the approach of Ref. 4, where similarities among image patches are considered by evaluating the raw pixels values, without resorting to more expressive appearance descriptors. In addition, SS-CCT embeds in the same model multiple features, while Ref. 4 is constrained to refer to a single one. Finally, the encoding provided by Ref. 4 is expensive in terms of memory, requiring thus more compact additional representations, like bag of words. This introduces other issues associated to vector quantization, dictionary learning, sparsity, etc. In our case, self-similarities are collected in a compact descriptor, without requiring higher-level descriptors.

As a proof of concept and for computational reasons, the proposed method is applied using the well-known HOG feature descriptor, and tested on two main classification scenarios: object and scene classification. The results witness significant performance improvements with respect to both the simple feature-based descriptors (HOG, LBP, SIFT) and the point-wise similarity-based SST approach in Ref. 10.

The present work considerably extends the study presented in Ref. 13, by detailing how the SS-CCT can be quickly computed through integral images, and showing numerical experiments. We also revise the experimental protocol for the scene recognition, obtaining results that are more generalizable. Finally, we consider the PASCAL VOC 2007 as further object recognition benchmark. All the new experiments confirm the effectiveness of our method.

The rest of the paper is organized as follows. In Sec. 2, the SS-CCT descriptor is introduced; in Sec. 3 some information on the object model is provided. In Sec. 4 the SS-CCT performances on Caltech-101, Caltech-256, PASCAL VOC 2007 and SenseCam datasets are reported and compared with other methods in the literature. Finally, in Sec. 4 conclusions and future work are envisaged.

2. Proposed Method

Let \( I \) be a gray scale or color image of size \( H \times V \), and let \( B \subset I \) a bounding box defining an area of interest in the image. We subdivide \( B \) into \( N \) generally overlapped rectangular regions \( R_i \), \( i = 1, \ldots, N \), with \( N = N_h \times N_v \) (respectively the number of
horizontal and vertical regions), each one of size $n = n_h \times n_v$ pixels (see Fig. 1). The stride of two adjacent regions is $S_h$ and $S_v$, along horizontal and vertical direction, respectively. The size of the bounding box $B$ is given by $\left\lceil \frac{(N_h - 1)S_h + n_h}{C_0} \right\rceil \times \left\lceil \frac{(N_v - 1)S_v + n_v}{C_1} \right\rceil$. By such relation it is clear that the union of the $N$ regions perfectly covers the bounding box and no region portion lies outside $B$. The degree of overlap between the regions depends both on the region size and on the strides. In general not every region pair share common pixels in $B$.

Let $z(x, y)$ be the $D$-dimensional vector of features extracted at a pixel with image coordinates $(x, y)$. The global Feature Level (FL) descriptor of the bounding box $B$ is obtained by stacking together the feature vectors whose coordinates belong to the bounding box itself:

$$FL = \{ z(x, y) : (x, y) \in B \}. \quad (1)$$

The proposed similarity level (SL) structural descriptor is built on top of FL, encoding the similarity among each couple of regions. In order to achieve a statistically robust and highly invariant description of this similarity, we calculate the covariance among each couple of features, using the feature values $z(x, y)$ as spatial samples.

In detail, given two regions $R_i$ and $R_j$, we calculate the $D \times D$ cross-covariance matrix $\text{Ccov}_{R_i, R_j}$ among the feature vectors $z(x, y)$ in the following way:

$$\text{Ccov}_{R_i, R_j} = \frac{1}{n-1} \sum_{(x,y) \in R_i} (z_{R_i}(x, y) - \bar{z}_{R_i})(z_{R_j}(x, y) - \bar{z}_{R_j})^\top, \quad (2)$$

with

$$z_{R_i}(x, y) = z(x + \Delta X_{R_i}, y + \Delta Y_{R_i}). \quad (3)$$

where the pixel differences $\Delta X_{R_i}$, $\Delta Y_{R_i}$ define the distance of the $i$th region from the first region at the upper left corner of the bounding box (see Fig. 2). They can assume
the following set of discrete values:

\[
\begin{align*}
\Delta X_{R_i} &= h_i \cdot S_h \quad h_i = 0, \ldots, N_h - 1 \\
\Delta Y_{R_i} &= v_i \cdot S_v \quad v_i = 0, \ldots, N_v - 1
\end{align*}
\]

(4)

\(\bar{z}_{R_i}\) in (2) is the mean of \(z(x, y)\) inside the region \(R_i\) defined as:

\[
\bar{z}_{R_i} = \frac{1}{n} \sum_{(x,y) \in R_i} z_{R_i}(x, y).
\]

(5)

In practice the \(a, b\)th element of \(\text{Ccov}_{R_i, R_j}\) is the spatial covariance of feature \(a\) in region \(R_i\) and feature \(b\) in region \(R_j\).

Notice that Cross-Covariance matrices do not share the same properties of covariance matrices. In particular, \(\text{Ccov}_{R_i, R_j}\) are not symmetric and, consequently, not semi-definite positive. Therefore, cross-covariance matrices do not lie on a Riemannian manifold defined by the set of semi-definite positive matrices,\(^{16}\) and the only known modality to use these descriptors in a machine learning framework is to vectorize them.

Calculating (2) across all the possible region pairs inside the bounding box \(B\), we obtain a block matrix \(\text{Ccov}\text{Block}\) of size \(DN \times DN\), defined as follows:

\[
\text{Ccov}\text{Block}(B) = \begin{bmatrix}
\text{Ccov}_{R_1, R_1} & \cdots & \text{Ccov}_{R_1, R_N} \\
\vdots & \ddots & \vdots \\
\text{Ccov}_{R_N, R_1} & \cdots & \text{Ccov}_{R_N, R_N}
\end{bmatrix}.
\]

(6)

It can be noticed from Eq. (6) that this matrix is block-symmetric, i.e. \(\text{Ccov}_{R_i, R_j} = \text{Ccov}_{R_j, R_i}\). Therefore the final structural descriptor, named SS-CCT, is built vectorizing \(\text{Ccov}\text{Block}(B)\) in the following manner:

\[
\text{SS-CCT} = \left[ \text{Vec}(\text{Ccov}_{R_1, R_1}) \, \text{Vec}(\text{Ccov}_{R_1, R_2}) \cdots \, \text{Vec}(\text{Ccov}_{R_1, R_N}) \right] \\
\left[ \text{Vec}(\text{Ccov}_{R_2, R_1}) \, \text{Vec}(\text{Ccov}_{R_2, R_2}) \cdots \, \text{Vec}(\text{Ccov}_{R_2, R_N}) \right] \\
\left[ \vdots \right] \\
\left[ \text{Vec}(\text{Ccov}_{R_N, R_1}) \, \text{Vec}(\text{Ccov}_{R_N, R_2}) \cdots \, \text{Vec}(\text{Ccov}_{R_N, R_N}) \right],
\]

(7)

where Vec is the standard vectorization operator.
The length of the SS-CCT descriptor is therefore \((N + 1)(N/2)D^2\). The final descriptor is obtained by joining together the FL of Eq. (1) and the SL of Eq. (7) descriptors, with a final length equal to \((N + 1)(N/2)D^2 + DM\) where \(M\) is the number of pixels in the bounding box (in general \(M\) is not equal to \(Nn\) because the regions can be overlapped, as shown in Fig. 1).

2.1. Efficient implementation

An efficient way to calculate the SS-CCT over multiple bounding boxes can be devised exploiting the concept of integral images. Each pixel of the integral image is the sum of all the pixels inside a rectangle bounded by the upper left corner of the image and the pixel of interest:

\[
\text{IntI}(x', y') = \sum_{x < x', y < y'} I(x, y). \tag{8}
\]

The cross-covariance, Eq. (2), among two regions \(R_i, R_j\) can be rewritten, expanding the means and rearranging the terms, as follows:

\[
\text{Ccov}_{R_i, R_j} = \frac{1}{n-1} \cdot \left[ \sum_{(x,y) \in R_i} \left( z_{R_i}(x, y) z_{R_j}(x, y)^T \right) - \frac{1}{n} \sum_{(x,y) \in R_i} z_{R_i}(x, y) \sum_{(x,y) \in R_i} z_{R_j}(x, y)^T \right]. \tag{9}
\]

To find the cross-covariance for a given couple of rectangular regions \((R_i, R_j)\), we have to compute the sum of each \(D \times D\) matrix \(z_{R_i}(x, y) z_{R_j}(x, y)^T\) and each \(D\) vector \(z_{R_i}(x, y)\). To do this, we build a set of integral tensors: Let \(P(x', y')\) be a 3D tensor of size \(H \times V \times D\) defined as follows:

\[
P(x', y') = \sum_{x < x', y < y'} z(x, y) \tag{10}
\]

and \(Q(x', y', \Delta X, \Delta Y)\) a 6D tensor of size \(H \times V \times (2N_h - 1) \times N_v \times D \times D\) defined as follows:

\[
Q(x', y', \Delta X, \Delta Y) = \sum_{x < x', y < y'} z(x, y) z^T(x + \Delta X, y + \Delta Y), \tag{11}
\]

where

\[
\Delta X = h_h S_h, \quad h_h = -N_h + 1, \ldots, N_h - 1
\]

\[
\Delta Y = h_v S_v, \quad h_v = 0, \ldots, N_v - 1. \tag{12}
\]
Now consider a bounding box whose first region $R_1$ in the upper left corner is bounded by pixels $(1,1)$ (upper left) and $(x', y')$ (lower right) of the whole image. In such a case, the formula of the Cross-Covariance, Eq. (9), can be expressed as follows:

$$ C_{\text{cov}}(1,1; x', y') $$

$$ = \frac{1}{n-1} \left[ \sum_{x < x'} \sum_{y < y'} (z_{R_1}(x,y)z_{R_1}(x,y))^\top - \frac{1}{n} \sum_{x < x'} \sum_{y < y'} z_{R_1}(x,y) z_{R_1}(x,y)^\top \right]. \quad (13) $$

Now, recalling the definition of $z_{R_1}(x,y)$ in Eq. (3), it is easy to see that the three sums in Eq. (13) can be expressed in terms of the integral tensors, Eqs. (10) and (11). In detail, defining the following quantities,

$$ P_{x'y'R_1} = P(x' + \Delta X_{R_1}, y' + \Delta Y_{R_1}) \quad (14) $$

$$ Q_{x'y'R_1} = Q(x' + \Delta X_{R_1}, y' + \Delta Y_{R_1}, \Delta X_{R_1} - \Delta X_{R_1}, \Delta Y_{R_1} - \Delta Y_{R_1}) \quad (15) $$

we can express Eq. (13) as:

$$ C_{\text{cov}}(1,1; x', y') = \frac{1}{n-1} \left[ Q_{x'y'R_1} - \frac{1}{n} P_{x'y'R_1} P_{x'y'R_1}^\top \right]. \quad (16) $$

In practice, the displacements $\Delta X_{R_1}, \Delta Y_{R_1}, \Delta X_{R_1}, \Delta Y_{R_1}$ defining the regions $R_1, R_1$ can be encoded into the relative displacements $\Delta X = \Delta X_{R_1} - \Delta X_{R_1}$ and $\Delta Y = \Delta Y_{R_1} - \Delta Y_{R_1}$ (see Fig. 2). In this way, the number of possible displacement combinations equal to $N_h^2 N_v^2$ is reduced to a much smaller number of relative displacements $(2N_h - 1)(2N_v - 1)$. Moreover considering that $C_{\text{cov}}(R_1, R_1) = C_{\text{cov}}^\top(R_1, R_1)$, just the quantities $C_{\text{cov}}(R_1, R_1)$ with $j \geq i$ need to be computed. Assuming to sort the regions in a row-wise manner, i.e. the first $N_h$ regions have $\Delta X_{R_1} = 0$, it holds that $\Delta Y_{R_1} \geq \Delta Y_{R_1}$ for each $j \geq i$ and consequently $\Delta Y \geq 0$, allowing to reduce the number of possible relative displacements to $(2N_h - 1)N_v$, as previously defined in (12). Overall the computation of Eq. (13) for each $(x', y')$ and each $(R_1, R_1)$ with the above-described integral tensors is $O(D^2 HV(2N_h - 1)N_v)$. Let us consider now a generic bounding box whose first region is bounded by the upper left and lower right corners, $(x', y')$ and $(x'', y''$, respectively. It is easy to see that the cross-covariance among two regions related to this bounding box, denoted with $C_{\text{cov}}(R_1, R_1)(x', y'; x'', y'')$, can be calculated as:

$$ C_{\text{cov}}(R_1, R_1)(x', y'; x'', y'') = \frac{1}{n-1} \left[ Q_{R_1, R_1} - \frac{1}{n} P_{R_1} P_{R_1}^\top \right], \quad (17) $$

where $Q_{R_1, R_1}$ and $R_1$ are linear combinations of the integral tensors defined as follows (see Fig. 3):

$$ Q_{R_1, R_1} = Q_{x'y'R_1, R_1} + Q_{x'y'R_1, R_1} - Q_{x'y'R_1, R_1} - Q_{x'y'R_1, R_1} \quad (18) $$

$$ P_{R_1} = P_{x'y'R_1} + P_{x'y'R_1} - P_{x'y'R_1} - P_{x'y'R_1} \quad (19) $$
Denoting with $N_B$ the total number of bounding boxes in the image, the computational cost of the global descriptor SS-CCT for all the bounding boxes is $O(N_B D^2 N (N + 1)/2 + M D^2 (2N_h - 1) N_v)$. The first term $O(N_B D^2 N (N + 1)/2)$ does not depend on the region size $n$ and accounts for the operations involved in Eqs. (17)–(19). The second term $M D^2 (2N_h - 1) N_v$ accounts for the operations involved in 10 and 11 which have to be computed once independently of the number of bounding boxes. If $B = 1$, i.e. the bounding box coincides with the whole image, then $M = [(N_h - 1) S_h + n_h][(N_v - 1) S_v + n_v]$. In general, $M = N_B (1 - O_h) (1 - O_v) \times [(N_h - 1) S_h + n_h][(N_v - 1) S_v + n_v]$, where $O_h$ and $O_v$ are the degree of overlap (in the range $[0, 1]$) between adjacent bounding boxes in horizontal and vertical directions. Differently, the computational cost associated to a naive procedure would be $O(N_B n D^2 N (N + 1)/2)$. The computational saving is significative and it is due to two factors: firstly sum over pixels are performed just once for the integral tensors and does not need to be repeated for each bounding box. Secondly the integral tensors are function of just the $(2N_h - 1) N_v$ relative displacements while in naive calculation each possible region couple, i.e. $N (N + 1)/2 = N_h N_v (N_h N_v + 1)/2$, must be taken into account.

In order to appreciate the computational advantage of the efficient implementation, its runtime is compared with the standard naive implementation considering different image resolutions, varying the number of regions in the image, their size and stride. In particular, in Fig. 4(a), runtimes are displayed for SS-CCT evaluated over a single image at QVGA resolution. It can be seen that increasing the number of regions, while decreasing the region size and keeping fixed the degree of overlap, results in an increase of the computational cost for both methods, but our efficient implementation guarantees a speed-up factor of about 3 to 13.5. Only in the limit case when just one region covers the entire image, the integral representation runtime is comparable with the standard one. Increasing the image resolution, the benefit in terms or runtime is even larger as
reported in Fig. 4(b). For a VGA image, the computational saving increases from 2.5 to 24.5 times.

More evident advantages of using the integral image implementation can be observed fixing the region size and increasing the number of regions, while decreasing the region stride, as can be seen in Fig. 5. Here the computational saving is even bigger with a significant speed-up of about 30 times.

Such comparisons are carried out considering just one bounding box. If a greater number of bounding boxes partially overlapping is considered, the computational advantage increases even more. Concerning the absolute runtimes, which in some cases are relevant, one has to consider that the implementation of both methods are in MATLAB®.
3. Object Model

The adopted object model depends on the size of the images considered and on the general characteristics of the dataset. In general, given an image, containing the object of interest, we calculate the low-level descriptor on a uniformly sampled set of patches, of size $w \times w$, whose overlap is $w/2$ in both horizontal and vertical dimensions. For every patch, we encoded the appearance of an object through the use of *Histograms of Oriented Gradients* descriptor, as defined in Ref. 3. We adopted this descriptor since it is relatively fast to compute and still considered one of the most expressive ones. Since each patch is mapped to a feature vector $z(x,y)$ related to a single pixel location, i.e. the patch center, the original image should be considered as decimated with a rate equal to $w/2$. In practice, the image $I$ of size $H \times V$, introduced at the beginning of Sec. 2, is a down-sampled version of the original image, of size $Hw/2 \times Vw/2$, on which HOGs are evaluated.

Since the experiments have been carried out on classification tasks, and not detection or localization ones, we considered a single bounding box coincident with the decimated image. After that, we defined a set of $N$ regions. The region size is defined considering the following criteria: (1) each region should contain a number of pixels sufficient to yield a significant statistic in the cross-covariance matrix calculus; (2) the patch size, determining the number of pixels over which $z(x,y)$ is evaluated, should be sufficiently large so as to retain the descriptor expressiveness; (3) finally, the region size should match the size of significant parts of the objects to be detected or classified.

\[\textit{a}\text{ It is important to distinguish between patch and region: The patch is the portion of the original image over which a single HOG descriptor is evaluated; the region is a portion of the decimated image over which covariance or cross-covariance of HOG descriptors is calculated.}\]
We calculate the SS-CCT descriptor evaluating the cross-covariance between all the couples of regions as formalized in Eqs. (6) and (7). The final descriptor, here dubbed $SS\text{-}CCT(\text{HOG})$, is given by the concatenation of SS-CCT and the HOG descriptors.

4. Experiments

In this section, we report experimental results obtained on two different tasks, using four datasets: Caltech-101, Caltech-256 and PASCAL VOC 2007 (object classification), and SenseCam Dataset (scene classification). In all the experiments, we employ a multi-class one-versus-all linear Support Vector Machine classifier, using LIBLINEAR, which is designed for linear classification of a large amount of data.

The proposed SS-CCT(\text{HOG}) is compared with a set of widespread descriptors including SIFT, LBP histograms, and the Self-Similarity Tensor described in Ref. 10. The latter, named SST(\text{HOG}), is built joining together the HOG descriptor and the pairwise Euclidean distances between all the patches, sharing the mixed feature-based and similarity-based philosophy of SS-CCT. In order to focus the comparison on the capabilities of the descriptors, the same baseline classifier and the same object model are adopted for all the tests.

4.1. Object classification

Caltech-101 and Caltech-256

In the object classification community, Caltech-101 dataset represents an important benchmark. It consists of 102 classes (101 object categories plus background) with a number of images per class ranging from 31 to 800. Despite its importance, Caltech-101 has some cues, notably the presence of strongly aligned object classes, which significantly ease the classification process. To overcome such limitation, the larger Caltech-256 dataset was subsequently introduced. It consists of 257 classes (256 + Clutter class) with a minimum of 80 images per class and a total number of images equal to 30,607. In Caltech-256, objects position inside the image is significantly varying for a lot of classes, as can be seen observing the average images for the 256 classes in Fig. 6, so making the classification task more challenging with respect to Caltech-101.

To test our descriptor, the object model introduced in Sec. 3 is adopted. The HOG, LBP and SIFT descriptors are calculated on dense patches of size $32 \times 32$ with an overlap of 16 pixels. The number of regions $N$ is set to $N = 9$, with $N_h = 3$ and $N_v = 3$; the region size is set to $n_h = n_v = 3$; finally the stride is set to $S_h = S_v = 3$. For Caltech-101, we considered 15 images per class for training and 15 images per class for testing, repeating the experiments with five different splits according to the standard procedure. The same was done for Caltech-256, we train our system on $\{5, 10, 15, 20, 25, 30\}$ images per class and test on 15 images, in 5 random splits each.
Experimental results on the Caltech-101 are displayed in Table 1. As can be seen, both SS-CCT(HOG) and SST(HOG) outperform HOG, LBP and SIFT with at least a 6% increment in the overall accuracy. On the other hand, SS-CCT(HOG) and SST(HOG) yield roughly the same performance: This is easily explainable considering that in Caltech-101 images are strongly aligned, reducing the need for robustness against position variation.

Results on the Caltech-256 in terms of accuracy versus the number of training images per class, are displayed in Fig. 7. As figure shows, our method outperforms HOG, LBP, SIFT and SST(HOG) in all the cases and the gap between our method and the others increase when the training set size is larger. Different from the Caltech-101 case, the higher complexity of the dataset highlights the superiority of our method with respect to SST(HOG).

In order to assess the statistical significance of the obtained results, a paired $t$-test has been run for both Caltech-101 and Caltech-256 taking the different splits as different realizations of the same process and considering as null hypothesis the statistical equivalence of SS-CCT(HOG) and the other descriptors. The obtained $p$-values for each couple [SS-CCT(HOG), other descriptor] are all lower than 0.006.

![Average of the images of the Caltech-256 dataset.](image)

Table 1. Classification results on the Caltech-101 dataset.

<table>
<thead>
<tr>
<th></th>
<th>SIFT</th>
<th>HOG</th>
<th>LBP</th>
<th>SST(HOG)</th>
<th>SS-CCT(HOG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>38.44%</td>
<td>41.32%</td>
<td>43.17%</td>
<td>47.67%</td>
<td><strong>47.77%</strong></td>
</tr>
</tbody>
</table>

Fig. 6. Average of the images of the Caltech-256 dataset.
except for \([\text{SS-CCT(HOG), SST(HOG)}]\) in Caltech-101 (\(p\)-value = 0.14) and \([\text{SS-CCT(HOG), SST(HOG)}]\) in Caltech-256 when 5 examples per class are used in training phase (\(p\)-value = 0.12). Overall the reliability of the improvement obtained with \text{SS-CCT(HOG)} is confirmed with a high degree of significance.

The parameters defining the patch size, the overlap and the region number, size and stride were tuned by trying a wide set of combinations of values and retaining the ones providing the best result. The tuning procedure was not extremely demanding from a computational point of view, as many parameters are mutually dependent (e.g. the region number and size). Interestingly, if the value of each parameter is chosen within a reasonable range, according to the criteria exposed in Sec. 3, the performance variation is not very large, and the method was found to be quite robust to parameter tuning. To support this statement in Fig. 8, the performance on Caltech-101 is displayed varying the patch size \(w \times w\), namely 16 \(\times\) 16 and 32 \(\times\) 32 pixels, and the number of regions, namely \(N = 2 \times 2\), \(N = 3 \times 3\) and \(N = 4 \times 4\). The patch overlap was fixed to half of the patch size while the region size was roughly inversely proportional to the number of regions, and the region stride was changed to keep roughly the same degree of region overlap. Results are visualized in terms of mean and

![Fig. 7. Results obtained on the Caltech-256 dataset.](image)

![Fig. 8. Results obtained on the Caltech-101 dataset, varying the patch size (16 \(\times\) 16 and 32 \(\times\) 32 pixels) and the number of regions \(N_h = N_v = 2, 3, 4\).](image)
The range of variability between the best and the worst result is about 5% in terms of accuracy, confirming the robustness of the method to the parameter tuning.

Pascal VOC 2007
The PASCAL VOC 2007 dataset consists of 9,963 images from 20 classes. These images range between indoor and outdoor scenes, close-ups and landscapes, and strange view-points. This dataset is extremely challenging, because all the images are daily photos obtained from Flicker with significant variations in the appearance of the objects (size, viewing angle, illumination, etc.), with frequent occlusions (see Fig. 9).

To test our descriptor, the object model introduced in Sec. 3 is adopted. The HOG descriptor is calculated on dense patches of size $32 \times 32$ with an overlap of 16 pixels. The number of regions $N$ is set to 36 (the images are bigger than the previous datasets), 6 along both the horizontal and vertical image direction, with a stride of $S_h = S_v = 2$. We considered the training/testing split available with the PASCAL VOC 2007 challenging.

The classification performance is evaluated using the average precision (AP) measure, a standard metric used by PASCAL challenge. It computes the area under the Precision/Recall curve, and the higher the score, the better the performance. In Table 2, we reported the average over all the 20 classes.

As table shows, our SS-CCT(HOG) outperforms both HOG and SST(HOG) with an overall average improvement that goes from 0.6% to 2%, respectively. Although the percentage increase is lower than in Caltech experiments, it demonstrates the goodness of our descriptor. As already demonstrated in the previous scenarios, our descriptor reaches the best performance when the intra-class variability is very high, i.e. cars and trains, whereas in other classes where objects are more aligned in the image, SST(HOG) may sometimes outperform our method. This behavior accounts for the complementarity of SS-CCT(HOG) and SST(HOG) and suggests that their combination could achieve even superior performance.
4.2. Scene classification

In the second experiment, the proposed framework is tested on the SenseCam Dataset. This dataset consists of images acquired with a SenseCam, a wearable camera which automatically shoots a photo every 20 s. It consists of 3962 images labeled according to 32 classes (e.g. bathroom home, car, garage home, biking, etc.). The images are divided into 30 random splits and in each round we extracted 480 images for training (15 images per class) and no more than 15 images for testing, for a total of 432. The dataset is challenging because most images present dramatic viewing angle, translational camera motions and large variations in illumination and scale. Figure 10 shows four images belonging to two classes extracted from the dataset.

As done in Ref. 13, the HOG descriptor has been calculated on dense patches of size $32 \times 32$ with an overlap of 16 pixels. The number of regions was set to 15:5 along the $x$-axis and 3 along the $y$-axis, with a stride of $S_h = S_v = 3$. Experimental results are displayed in Table 3, with standard deviations in brackets.

Our method outperforms both HOG and SST(HOG) with a difference in accuracy of about 6\% ($p$-value $< 10^{-4}$) and 2\% $p$-value $< 10^{-10}$ respectively, confirming its effectiveness in classifying images containing objects with a high degree of position variability.

Finally, as a complement to the runtime analysis carried out in Sec. 3, the three methods HOG, SST and SS-CCT are compared taking into account the image size and the object model considered in the four datasets previously described. In Table 4, runtimes for a single image are reported: For SST and SS-CCT runtimes do not include HOG computation which is common to the three approaches. The comparison is not completely fair because HOG has been implemented in C++ whereas the
other descriptors have been implemented in MATLAB. However, considering that a reasonable speed up with a C++ implementation would be around 5–10 it can be concluded that the additional computational complexity related to SS-CCT has a limited impact with respect to HOG and does not indent a practical application.

5. Conclusions and Future Work

This paper proposes a novel similarity-based descriptor for image classification. The idea is to encode similarities among different image regions by means of cross-covariance matrices calculated on low-level feature vectors, obtaining a robust and
compact representation of structural (dis)similarities of a given entity. The resulting descriptor SS-CCT can be efficiently calculated exploiting Integral Images, by means of an ad hoc procedure. The final descriptor, obtained joining together the low-level features (HOG in our case) and their structural similarities, has proven to out-perform all the other descriptors, on four challenging datasets. Despite the encouraging results obtained, further study will be devoted to find the best object model (number, shape and displacement of the parts) and the best features in a given context to improve the effectiveness of the proposed descriptor. This will allow the comparison with popular state-of-the-art approaches for detection and classification.

References


Marco San Biagio received M.Sc. degree cum laude in Informatics Engineering from the University of Palermo, Italy, in 2010, and Ph.D. in Computer Engineering from University of Genoa and Istituto Italiano di Tecnologia (IIT), Italy, in 2014, under the supervision of Professor Vittorio Murino and Professor Marco Cristani working on “Data Fusion in Video Surveillance”. Currently, he is a post-doc at the Pattern Analysis and Computer Vision department (PAVIS) in IIT, Genoa, Italy. His main research interests include statistical pattern recognition and data fusion techniques for object detection and classification.

Samuele Martelli received M.Sc. degree in Telecommunication Engineering in 2007 from University of Siena, Italy, and Ph.D. in Computer Science from the University of Verona, Italy, in 2012. Currently, he is a post-doc at the Pattern Analysis and Computer Vision department (PAVIS) in IIT, Genoa, Italy. His main research interests include statistical pattern recognition and data fusion techniques for object detection and classification, with a particular focus on the development of embedded computer vision systems.

Marco Crocco received Laurea degree in Electronic Engineering (2005) and Ph.D. in Electronic Engineering, Computer Science and Telecommunications (2009) from the University of Genoa. From 2005 to 2010, he worked at the Department of Biophysical and Electronic Engineering (DIBE) of the same university. In 2010 he got a post-doc position at the Istituto Italiano di Tecnologia (IIT), joining the Pattern Analysis and Computer Vision (PAVIS) group. He is the co-author of about 40 publications on international journals, proceedings of international conferences and book chapters.

Marco Cristani received M.Sc. degree in 2002 and Ph.D. degree in 2006, both in Computer Science from the University of Verona, Verona, Italy. He is now the Assistant Professor with the Department of Computer Science, University of Verona, working with the VIPS Lab. He is also a Team Leader with the Istituto Italiano di Tecnologia (IIT), Genova, working with the PAVIS Lab. His main research interests include statistical pattern recognition, generative modeling via graphical models, and nonparametric data fusion techniques, with applications on surveillance, segmentation and image and video retrieval.
Vittorio Murino is full professor and head of the Pattern Analysis and Computer Vision (PAVIS) department at the Istituto Italiano di Tecnologia (IIT), Genoa, Italy. He received Ph.D. in Electronic Engineering and Computer Science in 1993 at the University of Genoa, Italy. Then, he was first at the University of Udine and, since 1998, at the University of Verona, where he was Chairman of the Department of Computer Science from 2001 to 2007. He is also member of the editorial board of Pattern Recognition, Pattern Analysis and Applications, and Machine Vision & Applications journals, as well as of the IEEE Transactions on Systems, Man, and Cybernetics. Finally, he is a senior member of the IEEE and Fellow of the IAPR.