Top-down Saliency with Locality-constrained Contextual Sparse Coding

Hisham Cholakkal
hisham002@ntu.edu.sg
Deepu Rajan
http://www3.ntu.edu.sg/home/ASDRajan
Jubin Johnson
jubin001@ntu.edu.sg

School of Computer Engineering
Nanyang Technological University
Singapore

Abstract

We propose a sparse coding based framework for top-down salient object detection in which three locality constraints are integrated. First is the spatial or contextual locality constraint in which features from adjacent regions have similar code, second is the feature-domain locality constraint in which similar features have similar code, and third is the category-domain locality constraint in which features are coded using similar atoms from each partition of the dictionary, where each partition corresponds to an object category. This faster coding strategy produces better saliency maps compared to conventional sparse coding. Proposed codes are max-pooled over a spatial neighborhood for saliency estimation. In spite of its simplicity, the proposed top-down saliency achieves state-of-the-art results at patch-level on two challenging datasets-Graz-02 and PASCAL VOC-07. A novel Gaussian-weighted interpolation further improves pixel-level saliency map derived from the patch-level map.

1 Introduction

Identifying the salient regions within an image has attracted a great deal of interest in the computer vision community in the past decade for a diverse range of applications ranging from image classification [16, 17] to image segmentation [6]. In general, saliency estimation methods can be grouped into either unsupervised bottom-up works or fully supervised top-down approaches. In bottom-up approaches, low-level visual cues like color, contrast and uniqueness of the features are used to estimate the salient regions. On the contrary, top-down approaches are goal-oriented, enabling them to utilize prior knowledge about the task for better saliency estimation. Following [11, 19], the specific goal of top-down saliency in this work is to identify image regions that belong to a pre-defined object category, indicated by a probabilistic saliency map that peaks at object locations.

Top-down saliency estimation models perform well even in the presence of cluttered background and partial occlusion [9, 11, 19]. Knowledge about salient objects learned

Pages 159.1-159.12
DOI: https://dx.doi.org/10.5244/C.29.159

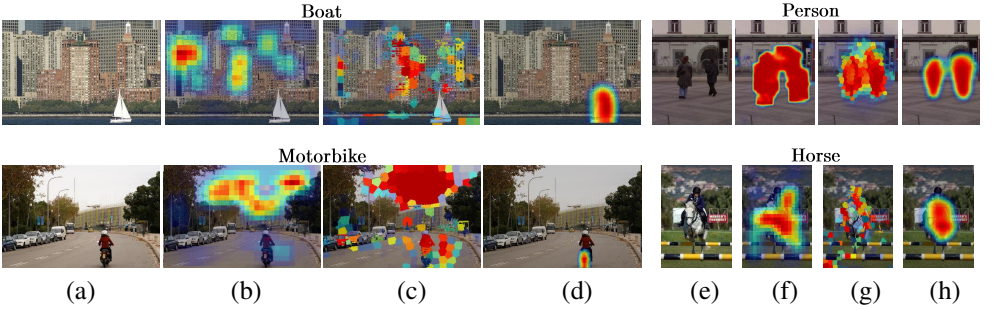


Figure 1: Illustration of proposed saliency model on challenging test images from PASCAL VOC-07 and Graz-02 datasets (best viewed in color). (a, e) input image, (b, f) Saliency maps of Yang and Yang [19], (c, g) Kocak *et al*. [11] and (d, h) the proposed method.

through supervised training enables these models to distinguish object patches from neighboring clutter. Recent works on top-down saliency use machine learning tools like conditional random field (CRF), dictionary learning and sparse coding (SC) [10, 11, 19]. Sparse codes of image features are also used in other applications like image classification [6, 20], object detection [15] and image matting [8]. Locality-constrained linear coding (LLC) [18], which replaces sparsity constraint with locality constraint in the feature domain, improves the speed of feature coding considerably. While sparse codes guarantee minimum reconstruction error, LLC ensures that similar features are coded similarly by limiting the coding of each feature to its nearest neighbor atoms in the dictionary. Sparse codes often fail to maintain the locality of features while LLC codes fail to ensure minimum reconstruction error for the feature. Both sparse coding and LLC coding do not consider contextual information in the neighborhood due to which features in spatially adjacent smooth regions are coded differently.

In the case of classification, it has been suggested [7] that using a discriminative dictionary in which features common to various classes are removed through supervised learning, results in increased accuracy with fewer number of dictionary atoms. However, removing common features limits performance in object localization, fine-grained image classification and in top-down saliency estimation [5]. In this paper, we propose a coding scheme that overcomes the above mentioned drawbacks of SC and LLC coding. The paper has two major contributions: (i) we propose locality-constrained contextual sparse coding (LCCSC) for a feature in which three forms of locality constraints are imposed on the code–category locality in which k-nearest neighbor atoms of a feature are chosen from sub-dictionaries of each category, spatial locality in which context is incorporated by selecting the nearest neighbor atoms for spatially nearby features and finally, feature locality as in [18]; (ii) we modify the contextual max-pooling of [22] by pooling the locality-constrained contextual sparse codes instead of category-specific LLC codes (note that context in LCCSC refers to the context arising from the spatial locality constraint and not to the contextual max-pooling operation). The advantages of the proposed method are (i) unlike [11, 19, 22], the sparse codes for different saliency models (classes) are not recomputed since the dictionary is common for all categories and (ii) faster computation time since there is no iterative dictionary learning as in [11, 19]. We also propose a Gaussian-weighted interpolation step to generate pixel-level saliency maps from patch-level maps. These contributions result in improved top-down saliency maps on Graz-02 [14] and Pascal VOC-07 datasets [3].

Fig. 1 shows three images from PASCAL VOC -07 dataset (boat, motorbike and horse) and one from person category of Graz-02 dataset along with their corresponding saliency maps. Our boat model assigned boat pixels with highest saliency values (fig. 1 (d)), while boat models of state-of-the-art top-down saliency approaches [11, 19] fail (fig. 1 (b, c)) due to cluttered background. Similarly, our person model could separate two persons (fig. 1 (h)) in the person image of Graz-02 dataset (fig. 1 (e)) while [11, 19] (fig. 1(f, g)) failed to do so. Motorbike was successfully assigned highest saliency values by proposed motorbike model (fig. 1 (d)) even though the image contains person, car and bus categories. Similarly, the proposed horse model produces a better saliency map (fig. 1 (h)) as compared to others.

2 Related Work

There are only a few top-down saliency approaches in the literature, which is dominated by bottom-up saliency methods. In general, top-down saliency models are learning-based approaches requiring a set of training images. In [4], a saliency model is learnt using discriminant features from a pre-defined filterbank (e.g DCT). Mutual information is used to compute the saliency of an image. Since salient features are selected based on image-level statistics, it fails to suppress background patches resulting in poor performance in the presence of cluttered background. Independent component analysis (ICA) along with spatial priors is used in [9] to replace the pre-defined filter bank of [4]. Its accuracy reduces considerably if the target appears at a location different from its location prior. Yang and Yang [19] proposed joint learning of CRF parameters and a category-specific dictionary. Sparse codes of SIFT features are used as latent parameters to learn the dictionary. Even after several iterations of joint dictionary learning, this method cannot suppress background patches that are visually similar to the object. To improve the pixel-level accuracy of this approach, Khan and Tappen [10] proposed a discriminative dictionary learning with spatial priors. Kocak *et al*. [11] further improved the pixel-level accuracy using superpixel-based features instead of image patches. Superpixel-based computations, use of color-based features and objectness [2], makes this approach computationally less efficient. The proposed framework is closely related to contextual pooling proposed in [22]. They use LLC coding of SIFT features on category specific dictionaries followed by contextual max-pooling and logistic regression. On a test image, the LLC codes need to be recomputed to estimate the saliency map for each object category. In the proposed framework, a novel LCCSC coding is used instead of LLC coding, wherein the sparse codes need not be computed since the dictionary does not change with object category. Moreover, when LCCSC is used in conjunction with the proposed Gaussian weighted interpolation from patch-level to pixel-level saliencies, we achieve more accurate saliency maps compared to [22].

Conventional object localization approaches aim to find a rectangular bounding box around the object while object segmentation approaches associate each pixel to object or background. On the other hand, approaches like [1, 13] generate an object estimate similar to that of a probabilistic top-down saliency saliency map. Compared to the proposed framework, [13] needs additional information about the training images indicating whether it is truncated or difficult and [1] needs a much larger code-book (500,000 atoms) compared to 512 atoms per category in our approach. Similiar to a recent top-down saliency paper [11], our results are compared with these approaches.

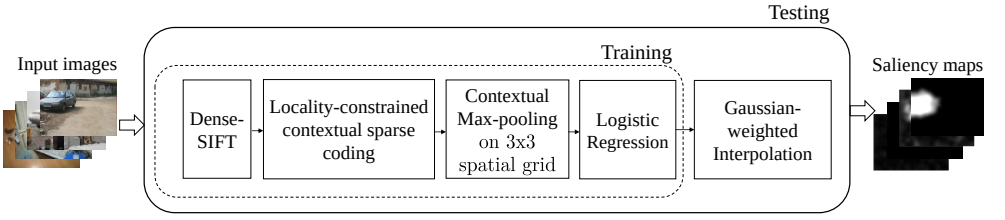


Figure 2: Training and testing of the proposed top-down saliency approach.

3 System overview

Fig. 2 shows the pipeline of the proposed method which is similar to the widely used sparse coded spatial pyramid matching (ScSPM) image classifier, i.e., feature extraction, feature coding, pooling and feature classifier. To reduce computations, dense SIFT features are extracted using only a single patch size. A novel locality-constrained contextual sparse coding strategy is proposed for feature coding. Spatial neighborhood of a patch is divided into a regular grid and the codes in each cell of the grid are max-pooled individually. These max-pooled vectors are vertically concatenated to form a context max-pooled vector representing the patch. Logistic regression-based feature classifier is learnt using these context max-pooled vectors. For saliency inference on a test image, the class-conditional probability of context max-pooled vectors are estimated from the learnt logistic regression model. This probability is the saliency value of a patch in the image. The pixel-level saliency map is obtained using a novel Gaussian-weighted interpolation of the patch-level saliency map.

4 Locality-constrained contextual sparse coding (LCCSC)

4.1 Formulation

Various coding schemes aim for specific objectives like sparsity [12], feature-domain locality [18] and spatial-domain locality [10, 21]. Here, all these desired properties are integrated into a single objective function. Also, feature coding using a *discriminative* dictionary is not the goal in top-down saliency; rather feature codes should be agnostic to object categories as long as the features contribute to locating the salient object (this will be elaborated soon). LCCSC ensures that features representative of salient regions are not ignored even if they are not discriminative.

Given a feature vector f and dictionary D with elements $D = [d_1, d_2, ., d_n]$, LCCSC coding searches for the codeword z that satisfies the following criteria:

$$\arg\min_{z} \|f - Dz\|_2 + \lambda_1 \|z\|_1 + \lambda_2 \|z \odot h_\pi\|_2 + \lambda_3 \sum_{j=1}^{c} \left(\left| \frac{\|z\|_0}{c} - \left\| z \odot col_j[\rho] \right\|_0 \right| \right); \quad (1)$$

where c is the number of object categories, $col_j[\rho]$ is the j^{th} column of a binary matrix ρ (to be described later) and $\odot$ is element-wise multiplication. The first two terms are the conventional sparse coding of feature f with l_1 constraint [12]. The third term imposes locality constraint in the feature domain as well as in the spatial domain. It is motivated from LLC [18] in which the feature domain locality constraint is denoted by $||z \odot h||_2$, where

$h = exp(\frac{dist(f,D)}{\sigma})$ and $dist(f,D)$ is a vector representation of the Euclidean distance between feature f and each atom (dictionary entry) of D. σ adjusts the rate of decay of the locality weight. In our formulation, the difference from [18] is in ensuring spatial-domain locality constraint also by considering not only a single feature f, but a set of features $f_{\mathcal{N}} : \mathcal{N} = 1, \ldots, N$ in the spatial neighbourhood of f. The *context* in LCCSC refers to the spatial neighborhood. Each vector f_i in $f_{\mathcal{N}}$ will have a corresponding vector h_i, which is a function of its Euclidean distance to the dictionary atoms. The minimum of this distance for all vectors in $f_{\mathcal{N}}$ constitutes h_π in eq. (1). i.e, n^{th} element of h_π is the minimum distance to the n^{th} atom in the dictionary among $f_{\mathcal{N}}$ which is computed by $h_\pi(n) = h_i(n),\ if\ h_i(n) \leq h_j(n)\ \ \forall j \in \mathcal{N}$ as illustrated in fig. 3. Thus, h_π ensures that the third term draws lower penalty when the non-zero terms in the code z corresponds to the dictionary atoms for which the distance from the feature or its neighbors is minimum.

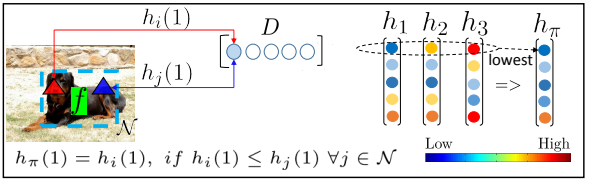


Figure 3: Illustration of computation of h_π.

As stated earlier, in top-down saliency, we are interested in how useful a feature is in identifying the salient object rather than in its discriminative ability. For example, a wheel which is common to the two classes of motorbike and car may not appear in a discriminative dictionary learnt for image classification [5]. If the dictionary is formed by unsupervised k-means clustering of features from all categories [21], atoms corresponding to wheel may be an averaged version of motorbike and car wheels due to the possibility of both types of wheels falling into the same cluster. Both these scenarios are not suitable for top-down saliency. In top-down saliency since the goal is to assign a probability to a feature f based on its representativeness in each category, its code z should be such that the number of atoms that contribute to the non-zero values of z are distributed among all object categories. Based on the association of the feature to each class their values can differ. This corresponds to $||z||_0/c$ in the fourth term of eq. (1). The underlying assumption is that the dictionary is partitioned so that there is the same number of atoms from each category in a partition (see fig. 4). In a practical situation an equal distribution of z values ($||z||_0/c$) is desired but not always possible and thus, the third term penalizes any deviation from the desired case. To this end, we define a binary matrix ρ of size $n \times c$ (n is the number of dictionary atoms) whose element ρ_{ij} is set to 1 if the i^{th} atom in the dictionary belongs to the j^{th} category. Now, consider two cases–one in which a code has two non-zero elements, both of which correspond to the same category (e.g., the first two elements are non-zero) and the other in which the two non-zero elements correspond to different categories. The penalty for deviation from desired as represented by the fourth term term in eq. (1) is higher in the former case, as desired. This illustrates the third aspect of locality, viz., category-specific locality, where atoms *local* to each object category participate in the sparse coding process as shown by the filled circles of O_1, O_2 and O_3 in fig. 4.

4.2 Approximate solution

A closed-form solution of eq. (1) is clearly not possible. Our approach is to ensure that the dictionary used in the sparse coding process satisfies the three locality constraints–feature domain, spatial domain (context) and category domain as formulated in the third and fourth terms of the objective function. The dictionary formation and subsequent sparse coding is

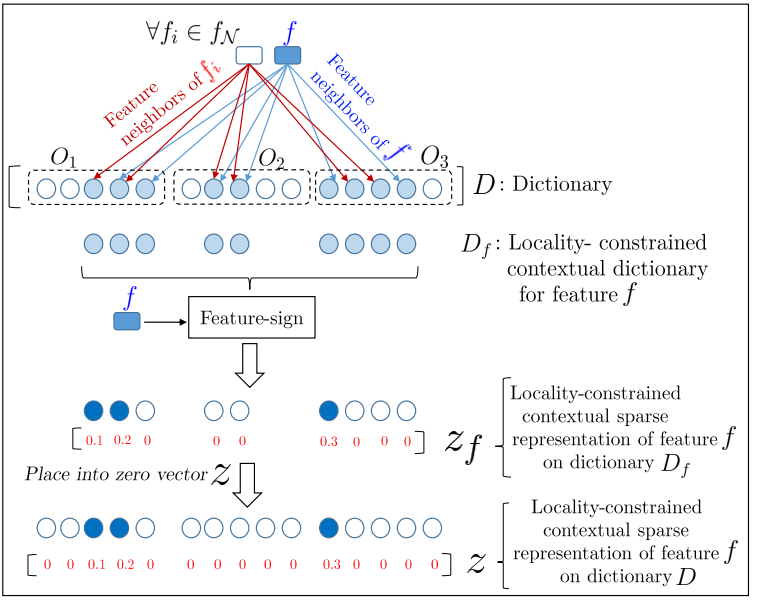


Figure 4: Illustration of approximate solution for locality-constrained contextual sparse coding for a feature f. Here f_i indicates a neighboring patch (best viewed in color).

shown in fig. 4. As noted in the previous section, an initial dictionary is formed by concatenating separate sub-dictionaries formed by k-means clustering of SIFT features from each category. In fig. 4, O_1, O_2 and O_3 are three such sub-dictionaries that make up the dictionary D. Category domain locality along with feature domain locality is enforced by picking k-nearest neighbors of a feature f from each sub-dictionary O_i using hierarchical k-nearest neighbor search. Next the spatial locality or contextual constraint involves consideration of the set of features $f_{\mathcal{N}}$ in the spatial neighbourhood of f and imposing category domain locality on each element of $f_{\mathcal{N}}$; this results in another set of atoms picked from D. $f_{\mathcal{N}}$ is made up of patches that overlap the patch containing f. These patches constitute the *context* for LCCSC.

Having picked atoms from D based on the three aspects of locality constraints, they form a smaller locality constrained contextual dictionary D_f, which is used to encode feature f (fig. 4) into z_f for minimizing the first two terms in eq. (1) through the feature-sign solver [12] with $\lambda_1 = 0.15$. The final sparse code z is formed by placing values in z_f in their respective positions in a vector z initialized to 0. This is to ensure that the vectors passed on for subsequent max-pooling and logistic regression are of the same size irrespective of the number of atoms that are picked to form D_f.

Computational complexity: Let n_o be the number of atoms per category and n be the number of atoms in dictionary D. Let n_f be the size of D_f and q_f be the number of non-zero terms in the final sparse code z. Traditional sparse coding on D using feature-sign solver has a computational complexity of $O(nr) + O(nq)$. Here q is the number of non-zero terms in the code and r is the length of a dictionary atom ($r = 128$). LLC reduces these computations to $O(n + k^2)$ for k-nearest neighbors (typically $k = 5$). Computational complexity of our approach is in between those of sparse coding and LLC. Hierarchical k-nearest neighbor computations on each category dictionary separately results in $O(n_o)$ per

category. These k-nearest neighbor computations for each category can be implemented in parallel to achieve multifold speed-up compared to nearest neighbor computations on D in LLC which has a complexity of $o(n)$. Feature-sign solver requires additional computations of $O(n_f r)+O(n_f q_f)$. So, in a parallel implementation, the computational complexity of proposed LCCSC can be reduced to $O(n_o)+O(n_f r)+O(n_f q_f)$. Due to the inherent parallelism in the framework, and much smaller size of feature-specific dictionary D_f as compared to the full dictionary D, a parallel implementation of the proposed LCCSC is faster than a parallel implementation of conventional feature-sign solver.

5 Contextual max-pooling for top-down saliency estimation

Context has an important role in deciding whether an image patch belongs to a particular object [22]. Contextual max-pooling refers to the representation of each patch by a max-pooled vector computed over its spatial neighborhood. The contextual max-pooling is done for LCCSC code vectors. The contextual neighborhood scale for max-pooling is empirically set to 6. i.e., 6 patches surrounding the current patch in each direction are considered for max-pooling. To preserve the spatial layout, feature codes of these 169 $((2\times 6+1)\times(2\times 6+1))$ patches in the context are equally divided into a 3×3 spatial grid [22]. Separate max-pooling on each of these 9 regions followed by vertical concatenation of these max-pooled vectors forms the contextual max-pooled vector which represents the patch containing feature f. Contextual max-pooled vectors from object patches and from an equal number of negative patches are collected from all training images and a logistic regression is learnt for each object. These logistic regression models are the top-down saliency models for the object, and are used to estimate the saliency map. The proposed saliency inference on a test image is simple and fast. SIFT feature extraction followed by locality-constrained contextual sparse coding, contextual max-pooling and prediction of class conditional probability by logistic regression gives the saliency for a patch. Pixel-level saliency maps are computed from patch-level saliency values through our Gaussian-weighted interpolation.

6 Gaussian-weighted interpolation for pixel-level saliency map generation

We propose a simple Gaussian-weighted approach to compute pixel-level saliency from patch-level saliency. Upsampling by bicubic interpolation used by Yang & Yang [19] reduces the pixel-level precision rates at EER by 10% compared to its patch-level accuracy [10]. In order to estimate the saliency value at a given pixel, we consider those patches that contain that pixel. Let $[p(1),\ p(2),\ldots.\ p(m)]$ be the saliency values for m image patches having centers at locations $(x_1,y_1),(x_2,y_2),\ ....,(x_m,y_m)$. Let g be the grid spacing i.e., the distance between adjacent patches. Since B is the width of the image patch, all image patches whose centers are located within $\frac{B}{2}$ radius from (x_l,y_l) contain the pixel at (x_l,y_l). Let $\Omega=[p_1,p_2,....p_j]$ be the patches containing (x_l,y_l) and $G(x_l,y_l,x_i,y_i)=exp^{-\frac{(x_i-x_l)^2+(y_i-y_l)^2}{2g^2}}$ be the Gaussian weight of patch i having center at (x_i,y_i) on this pixel. The saliency value at

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5: Illustration of Gaussian-weighted interpolation (best viewed in color). (a, e) Input image, (b, f) bicubic interpolation, (c, g) proposed Gaussian-weighted interpolation and (d, h) ground truth. White indicates true positive pixels, black true negatives, and red indicates false positves and false negatives.

(x_l, y_l) is computed as

$$p(x_l, y_l) = \frac{\sum_{i \,\varepsilon\, \Omega} p(i) G(x_l, y_l, x_i, y_i)}{\sum_{i \,\varepsilon\, \Omega} G(x_l, y_l, x_i, y_i)}. \tag{2}$$

Fig. 5 shows car and bike input images. From the same patch-level saliency values generated by proposed method, pixel level saliency maps are generated by bicubic interpolation and proposed Gaussian weighted interpolation. The saliency maps generated in both cases are binarized using a common threshold and misclassified pixels are shown in red (false positive+false negative). The proposed interpolation step results in better pixel-level saliency map with lesser false detections (fig. 5(c, g)) compared to bicubic interpolation used by [19] (fig. 5(b, f)) which spreads to the background pixels.

7 Experimental Results

We test our method on two challenging datasets - Graz-02 and PASCAL VOC-07. The same parameters are maintained in both datasets. For fair comparison, the same experimental set-up as [19] is followed, i.e., dense SIFT features are extracted from 64×64 image patches with a grid spacing of 16 pixels. For each object category, a sub-dictionary of 512 atoms is formed through k-means clustering of features extracted from positive training patches. A patch is considered as positive if at least 25% of the pixels belong to the object category.

7.1 Graz-02 dataset

Graz-02 dataset has 3 object categories (bicycle, car and person) and a background category. As in [11, 19], from each category, 150 odd numbered images are used for training and remaining 150 for testing. Saliency models for each object category are tested on 300 test images (150 test images of the object and 150 background test images) and precision rate at Equal Error Rate (EER) is determined.

Table 1(a) compares the patch-level results of our method with other top-down saliency models. The proposed method is called as LCCSC-pooled, where pooled indicates contextual max pooling of the code. LLC-pooled and SC-pooled refers to our framework except that LCCSC is replaced by LLC and SC respectively. DSD and SUN results are reported in [19]. In all the 3 classes we achieve state-of-the-art results. As illustrated in fig. 6, when LCCSC in the proposed framework is replaced by LLC (LLC-pooled) or SC (SC-pooled), the performance deteriorates, most notably in the car category because of background features that are similar to car features. Since LLC and SC do not consider context while coding, these

Table 1: Precision rates (%) at EER on Graz-02. (a) Patch level results and (b) pixel level results.

(a) Patch-level

	Bicycle	Car	Person	*Mean*
DSD [8]	62.5	37.6	48.2	49.43
SUN [8]	61.9	45.7	52.2	53.27
Yang and Yang [19]	80.1	68.6	72.4	73.7
LLC-pooled	81.91	71.3	70.9	74.71
SC-pooled	83.15	72.81	72.06	76.01
LCCSC-pooled	**83.46**	**75.97**	**73.13**	**77.52**

(b) Pixel-level

	Bicycle	Car	Person	*Mean*
Objectness [8]	53.5	48.3	43.5	48.43
Aldavert *et al.* [8]	71.9	64.9	58.6	65.13
Khan and Tappen [11]	72.1	-	-	-
Marszalek and Schmid [13]	61.8	53.8	44.1	53.23
Yang and Yang [19]	62.1	60.0	62.0	61.46
Kocak *et al.* [11]	73.9	68.4	**68.2**	70.16
LCCSC-pooled (*upsampling of [19]*)	73.41	68.6	61.25	67.75
LCCSC-pooled (*Gaussian-weighted interpolation*)	**76.19**	**71.2**	64.13	**70.49**

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6: Comparison of saliency maps generated using LCCSC against LLC and SC coding. (a, e) input images, (b, f) LLC-pooling, (c, g) SC-pooling and (d, h) LCCSC-pooling.

features resulted in false detection in this category (fig. 6(f, g)). The smooth regions within the person are not detected by LLC and SC (fig. 6(b, c)), but using context along with the other two locality constraints in LCCSC helped generate better saliency maps (fig. 6(d, h)). For fair comparison, LLC-pooled and SC-pooled are implemented on a dictionary of 2048 atoms formed by k-means clustering. Matlab simulations using parallel processing toolbox shows that LCCSC achieved 27% speed-up compared to conventional feature-sign based sparse coding on D (0.54 sec for LCCSC versus 0.74 sec for SC to encode 1000 features).

Table 1(b) compares pixel-level results of the proposed model with recent top-down saliency approaches, a bottom-up approach [2], and with two results in object segmentation [1, 13]. Even-though the saliency values are estimated at patch level, the Gaussian weighted interpolation yields the best reported results at pixel-level, which is better than models that estimate saliency directly at pixel-level [11]. The seventh row shows the EER of pixel-level saliency map using upsampling of [19] and the last row shows the results for Gaussian-weighted interpolation. The improvement by about 3% indicates the benefits of the proposed interpolation scheme. It is to be noted that we achieve this performance by using simpler feature coding and contextual max-pooling in comparison with computationally complex dictionary learning and graph based approaches of [11, 19]. Since conditional random field (CRF) used in these models are built on sparse codes, increasing the dictionary size will drastically increase the computational complexity by many times which is practically not feasible. Contextual max-pooling of LLC codes [18] are used by [22] to generate the saliency map. They use separate dictionaries of 1024 atoms for each category as opposed to a common dictionary for all categories. Instead of precision at EER, they report average precision at pixel-level. Our mean average precision is 75.5 (bike- 83.1, cars-75.7, person-68.3) which is much higher compared to their 62.1 (bike-69.1, cars-58.0, person-59.2).

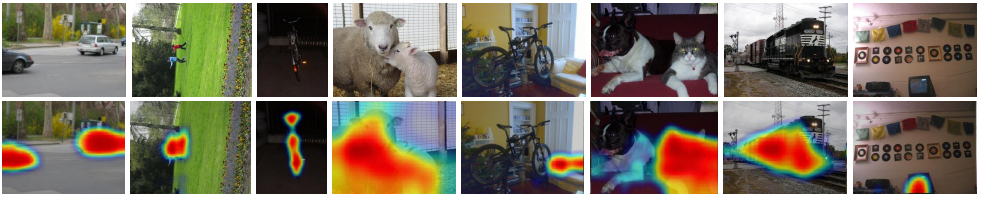
7.2 PASCAL VOC-07

PASCAL VOC-07 is a challenging dataset having 20 different object categories with multiple objects in some images. As in [19], all models are evaluated on the entire 210 segmentation

Table 2: Patch-level Classification rates at EER on PASCAL VOC-07

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
Yang and Yang [19]	**15.2**	**39**	9.4	5.7	3.4	22.0	30.5	**15.8**	**5.7**	8.0
Proposed	13.3	33.2	**22.1**	**11.2**	**8.6**	**33.5**	**37.2**	14.3	3.9	**22.3**

	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	T.V Monitor	Mean
Yang and Yang [19]	11.1	12.8	10.9	23.7	**42.0**	2	20.2	10.4	24.7	10.5	16.15
Proposed	**23.0**	**14.9**	**25.0**	**30.6**	38.9	**16.4**	**36.3**	**18.3**	**29.2**	**36.3**	23.4



Cars Person Bicycle Sheep Sofa Cat Train TV monitor

Figure 7: Top row: Input images from Graz-02 and PASCAL VOC-07 datasets. Bottom row: Our LCCSC-pooled results.

test images irrespective of the presence or absence of target. We outperform [19] in 15 out of 20 classes as shown in table 2. On averaging across all classes, in patch-level, we achieve a mean precision rate at EER of 23.4% which is better than 16.15% of [19]. With the help of proposed Gaussian weighted interpolation, we achieve a mean precision rate at EER of 17.65% in pixel-level. Khan and Tappen [10] report precision of 8.5% only for cow category for which our method gives 22.66%. We do not compare with [11] since they manually assign an all zero saliency map, if the object of interest is not present in the test image.

Fig. 7 shows qualitative results on challenging test images from Graz-02 (Cars, Person and Bike) and PASCAL VOC-07 (Sheep, Sofa, Cat, Train and TV monitor) datasets. Proposed method could perform well even on a rotated image (Person). The sofa was correctly detected even though the image is dominated by bicycle which is another category in the dataset. Similarly, cat is assigned with higher saliency, in spite of the presence of dog (another category) in the image. TV monitor is correctly identified in spite of the presence of visually similar structures within the image.

To compare with pixel classification accuracy of [1], our pixel-level saliency maps are threshold at 0.5, so that pixels having a saliency value above 0.5 is treated as belonging to that object category otherwise background category. A pixel is assigned to the class having highest saliency value in the cases where more than one category produces saliency value above threshold. This simple thesholding of saliency map gives an average pixel classification accuracy of 32.33% which is far superior as compared to 23% of dedicated class segmentation approach [1].

8 Conclusion

In this paper we propose a simple and highly efficient feature coding strategy specifically for top-down saliency estimation. The proposed Gaussian-weighted interpolation produces better pixel-level saliency map from patch-level saliency values. We plan to improve our top-down model by combining with bottom-up saliency approaches such as objectness [2].

References

[1] D. Aldavert, A. Ramisa, R. L. de Mantaras, and R. Toledo. Fast and robust object segmentation with the integral linear classifier. In *CVPR*, 2010.

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[4] D. Gao, S. Han, and N. Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *PAMI*, 2009.

[5] S. Gao, I.-H. Tsang, and Y. Ma. Learning category-specific dictionary and shared dictionary for fine-grained image categorization. *IEEE Transactions on Image Processing (TIP)*, 23(2):623–634, Feb 2014.

[6] K. R. Jerripothula, J. Cai, F. Meng, and J. Yuan. Automatic image co-segmentation using geometric mean saliency. In *ICIP*, 2014.

[7] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *CVPR*, 2011.

[8] J. Johnson, D. Rajan, and H. Cholakkal. Sparse codes as alpha matte. In *BMVC*, 2014.

[9] C. Kanan, M. H. Tong, L. Zhang, and G. W. Cottrell. Sun: Top-down saliency using natural statistics. *Visual Cognition*, 17(6-7):979–1003, 2009.

[10] N. Khan and M. F. Tappen. Discriminative dictionary learning with spatial priors. In *ICIP*, 2013.

[11] A. Kocak, K. Cizmeciler, A. Erdem, and E. E. Top down saliency estimation via superpixel-based discriminative dictionaries. In *BMVC*, 2014.

[12] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2007.

[13] M. Marszałek and C. Schmid. Accurate object recognition with shape masks. *International journal of computer vision*, 97(2):191–209, 2012.

[14] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *PAMI*, 28(3):416–431, 2006.

[15] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR*, 2013.

[16] Z. Ren, S. Gao, L.-T. Chia, and I.-H. Tsang. Region-based saliency detection and its application in object recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 24(5):769–779, May 2014.

[17] G. Sharma, F. Jurie, and C. Schmid. Discriminative spatial saliency for image classification. In *CVPR*, 2012.

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[19] J. Yang and M.-H. Yang. Top-down visual saliency via joint crf and dictionary learning. In *CVPR*, 2012.

[20] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[21] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Low-rank sparse coding for image classification. In *ICCV*, 2013.

[22] J. Zhu, Y. Qiu, R. Zhang, J. Huang, and W. Zhang. Top-down saliency detection via contextual pooling. *Journal of Signal Processing Systems*, 74(1):33–46, 2014.