Discriminative Spatial Saliency for Image Classification

Gaurav Sharma, Frédéric Jurie, Cordelia Schmid

Discriminative Spatial Saliency for Image Classification

Gaurav Sharma[1,2], Frédéric Jurie[1], Cordelia Schmid[2]
[1]GREYC, CNRS UMR 6072, Université de Caen
[2]LEAR, INRIA Grenoble Rhône-Alpes
http://lear.inrialpes.fr/

Abstract

In many visual classification tasks the spatial distribution of discriminative information is (i) non uniform e.g. person 'reading' can be distinguished from 'taking a photo' based on the area around the arms i.e. ignoring the legs and (ii) has intra class variations e.g. different readers may hold the books differently. Motivated by these observations, we propose to learn the discriminative spatial saliency of images while simultaneously learning a max margin classifier for a given visual classification task. Using the saliency maps to weight the corresponding visual features improves the discriminative power of the image representation. We treat the saliency maps as latent variables and allow them to adapt to the image content to maximize the classification score, while regularizing the change in the saliency maps. Our experimental results on three challenging datasets, for (i) human action classification, (ii) fine grained classification and (iii) scene classification, demonstrate the effectiveness and wide applicability of the method.

1. Introduction

The human visual system is capable of analyzing images quickly by rapidly changing the points of visual fixation. Estimating the distribution of such points i.e. the *visual saliency* is an important problem in computer vision [13, 15, 21, 29]. Initial works on visual saliency detection addressed generic saliency, highlighting (generally interesting) properties such as edges, contours, color, texture *etc*., building on the feature integration theory [15, 25]. For visual discrimination, generic visual saliency should be adapted to include task specific information. Many works [9, 10, 20], thus, define and compute saliency based on the *discriminative* power of local features *i.e*. how much does a feature contribute towards separating the classes. Such feature based discriminative saliency has been shown to be important in automatic visual analysis.

Furthermore, in many visual classification tasks there is a spatial bias which complements global feature saliency

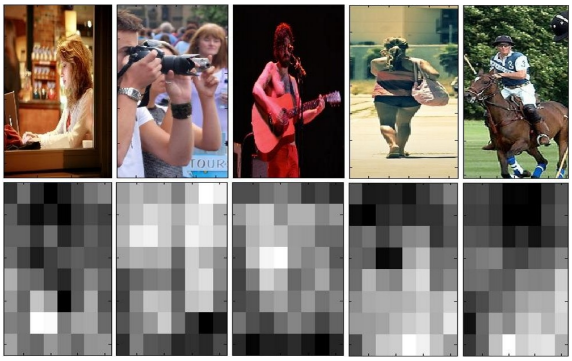


Figure 1. Example images and their spatial saliency maps obtained with our algorithm for 'interacting with computer', 'taking photo', 'playing music', 'walking' and 'ridinghorse' action classes (higher values are brighter).

e.g. for the 'coast' class in scene classification, sky-like regions are salient, not everywhere but in the *upper part* of an image. Thus, we argue that given a class, visual saliency is attributed to different local regions based on their appearance *and* their spatial location in an image *i.e*. a task specific *spatial saliency* is associated with each image.

In the present paper, we (i) extend the notion of discriminative visual saliency by including discriminative *spatial* information and (ii) learn it, together with the classifier, to obtain a more discriminative image representation for visual classification. Contrary to previous works [9, 10, 13, 14, 15, 20] that use saliency of features, irrespective of their positions, we work with saliency of regions in space *i.e*. for the 'ridinghorse' class instead of saying 'look for horse like features' we say 'look for horse like features *in the lower part of the image*'. Fig. 2 illustrates this point and Fig. 1 shows saliency maps obtained by our method.

Our definition of saliency is closely coupled with learning the classifier, unlike previous work which learn the saliency map and the classifier separately [9, 10, 23]. We learn the classifier while simultaneously modeling saliency in an integrated max margin learning framework. We formulate saliency in terms of local regions, and the learning based on a latent SVM framework adapted to incorporate

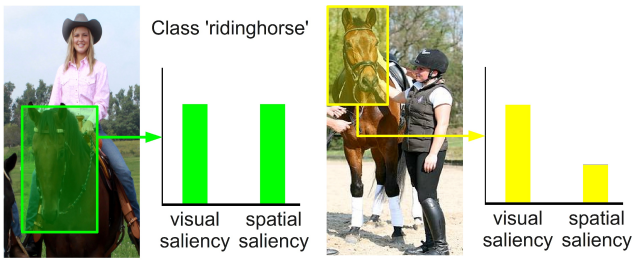


Figure 2. Illustrating the importance of spatial saliency. A horse is salient for the 'ridinghorse' class. However, it is salient if it appears in the lower part of the image (*e.g.* left image), but not if it appears in some other part of the image (*e.g.* right image).

the saliency model. We show that our saliency model improves results on three challenging datasets for (i) human action classification in images [5], (ii) fined grained classification i.e. persons playing vs. holding musical instruments [32] and (iii) scene classification [17].

1.1. Related work

Visual saliency has been investigated in the computer vision literature in many different ways. Salient local regions have been detected using interest points (*e.g.* [18, 19]) which can be made invariant to image transformations (*e.g.* rotation, scale, affine) and, thus, can be detected reliably and repeatably. They have been very successful for matching images under different transformations [18, 19]. Such regions were also used to sample small sets of salient patches from images for classification with bag-of-features representations [3], but dense (regular or random) sampling has been shown to perform better [22] and is currently the state-of-the-art [6].

Biologically inspired saliency, based on the feature integration theory [25], motivated another line of work. Regions were marked as salient depending on the difference with their surrounding area [13, 15], measured using low level features *e.g.* edges, texture, contours. Such generic saliency was further adapted to discriminative saliency [9, 10, 14, 20], where, given a visual classification task, saliency was defined by the capability of the features to separate the classes.

Moosmann *et al.* [20] learn saliency maps for visual search to improve object categorization. Gao and Vasconscelos [9] formulate discriminative saliency and determine it based on feature selection in the context of object detection [10]. Parikh *et al.* [23] learn saliency in an unsupervised manner based on how well a patch can predict the locations of others. Khan *et al.* [14] model color based saliency to weight features. Harada *et al.* [11] learn weights on regions for classification. However, they learn the weights per class *i.e.* the weights are the same for all images. Yao *et al.* [32] learn a classifier with random forests. They mine salient patches, for the decision trees, by randomly sampling patches and selecting the most discriminative ones.

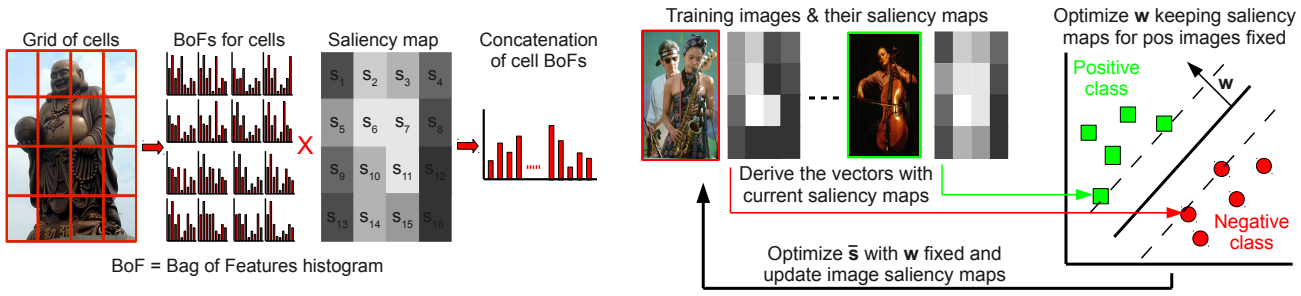
We model saliency based on the contribution of regions to classification i.e. our saliency is discriminative. We do not discard features, but weight them using the saliency map, which differs from *e.g.* [10, 22, 23]. Our model incorporates saliency modeling into the learning of separating hyperplane in a max margin framework. Hence, our saliency is more tightly coupled with the visual discrimination task unlike many previous works where learning saliency and classifiers are separate steps *e.g.* [9, 10, 14, 23].

Recently, latent support vector machine (LSVM) classifiers have shown promise in many visual tasks. Felzenszwalb *et al.* [8] use LSVM for part based object detection which has become a standard component in state-of-the-art systems [6]. Bilen *et al.* [1] model the position and size of the objects using LSVM for image classification. We adapt the LSVM formulation to incorporate saliency modeling. In our model the image saliency maps are latent variables and are thus integrated with learning the classifier.

2. Approach

We define image saliency as a mapping $s : \mathcal{G} \rightarrow \mathbb{R}$, where $\mathcal{G}$ is a spatial partition of the image, $c \in \mathcal{G}$ is a region of the image and $s(c)$ gives the saliency of the region. Our method is general and can work with any spatial partition of the images *e.g.* $\mathcal{G}$ can be the set of all image pixels, as in traditional saliency, or a set of user specified regions. We choose $\mathcal{G}$ to be the set of cells obtained with a spatial pyramid like uniform grid [17]. This is motivated by two reasons. First, we have a variable corresponding to every element of $\mathcal{G}$ for every image and, since contemporary visual discrimination datasets [4, 17, 31] have limited number of training images, using very fine regions *e.g.* pixels would make the number of variables very large compared to the training data. Second, the spatial pyramid, despite of its simplicity, is competitive with methods using more complex spatial models [6]. Given our choice of $\mathcal{G}$, we can equivalently write a saliency map as an ordered list of real values *i.e.* $\mathbf{s} = \{s_c | c \in \mathcal{G}\}$ where we use the row major order of the grid cells (Fig. 3a).

We work in a supervised binary classification scenario with given training images $I^i \in \mathcal{I}$ and corresponding class labels $y^i \in \{-1, 1\}$. Our model consists of three components, (i) the separating hyperplane $\mathbf{w}$, (ii) the image saliency maps $\mathbf{s}^i$ for images $I^i \in \mathcal{I}$ and (iii) a generic saliency map $\bar{\mathbf{s}}$ for regularizing the image saliency maps. The saliency map of an image maximizes the classification score while penalizing its deviation from the generic saliency map. Our full model is obtained by solving a max-margin optimization problem with the image saliency maps as latent variables. We present our model in the following



(a) (b)

Figure 3. (a) The images are represented by concatenation of cell bag-of-features weighted by the image saliency maps. (b) We propose to use a block coordinate descent algorithm for learning our model (Sec. 2.4). As in a latent SVM, we optimize in one step the hyperplane vector $\mathbf{w}$ keeping the saliency maps of the positive images fixed and in the other step we optimize the saliency keeping $\mathbf{w}$ fixed.

sections.

2.1. Maximum margin formulation

Given a saliency map $\mathbf{s}^i = \{s^i_c | c \in \mathcal{G}\}$ for the i^{th} image, we represent the image with the saliency map weighted concatenation of bag-of-features (BoF) histograms for the grid cells (Fig. 3a), *i.e.*

$$\mathbf{x}^i = [s^i_1 \mathbf{h}^i_1 \dots s^i_c \mathbf{h}^i_c \dots], \tag{1}$$

where $\mathbf{h}_c$ is the BoF histogram for cell $c \in \mathcal{G}$ with appropriate normalization. As noted in [27], normalization plays an important role, and we discuss this in more detail later.

We cast the problem in a maximum margin latent SVM framework with the image saliency maps $\{\mathbf{s}^i | I^i \in \mathcal{I}\}$ as latent variables. The optimization with hinge loss becomes

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C \sum_i \max(0, 1 - y^i f(\mathbf{x}^i, \mathbf{w})), \tag{2}$$

where f is the scoring function (Sec. 2.2, Eq. 3).

Latent SVMs have been very popular recently in the computer vision community [1, 8]. They lead to a semi-convex optimization i.e. the objective function is convex if the latent variables for the positive examples are fixed.

2.2. Image score

We score a given image as $f(\mathbf{x}, \mathbf{w}) = \max_{\mathbf{s}} \mathbf{w}^T \mathbf{x}$ (omitting superscript i for brevity) *i.e.* we allow the saliency map of the image to change to maximize its score w.r.t. the separating hyperplane. However, this leads to the trivial solution of selecting the highest scoring cell. To avoid this, we introduce a new variable, a generic saliency map, $\bar{\mathbf{s}}$. We penalize the score proportional to the deviation of the image saliency map from $\bar{\mathbf{s}}$. This regularizes the image saliency maps and gives smoother maps. The final score is thus obtained as

$$f(\mathbf{x}, \mathbf{w}) = \max_{\mathbf{s}} \mathbf{w}^T \mathbf{x} - \lambda (\mathbf{s} - \bar{\mathbf{s}})^T (\mathbf{s} - \bar{\mathbf{s}}), \tag{3}$$

where λ is the parameter controlling the trade off between maximizing the score by varying the saliency map and deviation of the image saliency map from $\bar{\mathbf{s}}$. We rewrite the first term of the score as

$$\mathbf{w}^T \mathbf{x} = \sum_{c=1}^{|\mathcal{G}|} s_c \sum_{k=1}^{K} w_{(c-1)\cdot K + k}\ h_{ck} = \mathbf{s}^T \mathbf{D}_w \mathbf{H}^T, \tag{4}$$

where K is the size of BoF codebook, $\mathbf{H}^T = [\mathbf{h}^T_1 \dots \mathbf{h}^T_{|\mathcal{G}|}]$ (concatenation of cell BoF histograms with appropriate normalization) and

$$\mathbf{D}_w = \begin{bmatrix} w_1 \dots w_K & & 0 \\ & \ddots & \\ 0 & & w_{(|\mathcal{G}|-1)\cdot K+1} \dots w_{|\mathcal{G}|K} \end{bmatrix}.$$

Normalization of the BoFs. As noted by Vedaldi et al. [27], in the context of linear classifiers, unnormalized histograms favor (assign relatively larger scores to) larger regions, L1 normalization favors smaller regions while L2 normalization is neutral and thus ideal. In our experiments, the images are of different size and the grids, specified in terms of fractional multiples of image width and height, results in different sized regions which makes normalization important. Harzallah et al. [12] had also previously noted that normalizing each cell separately instead of globally normalizing the whole descriptor gives slightly better results. Our preliminary experiments resulted in similar conclusions and in our final implementation we work with per-cell L2 normalized vectors *i.e.* each of the $\mathbf{h}_c$ are L2 normalized independently. The optimization problem in Eq. 3, after rewriting the first term using Eq. 4, takes a closed form solution (for $\mathbf{s}$) involving matrix operations and is very fast to compute.

2.3. Regularized formulation

By introducing $\bar{\mathbf{s}}$ into the formulation we have introduced another source of scaling. Everything else fixed, by scaling

Algorithm 1 Stochastic gradient descent for w ($\bar{\mathbf{s}}$ fixed)

while $t = 1 \dots T$ **do**
  Specify learning rate l_t^w for iteration t
  Choose a random training image I^i
  Calculate the saliency map $\mathbf{s}^i$ iff $y^i = -1$
  if $y^i f(\mathbf{x}_i, \mathbf{w}) \geq 1$ **then**
    $\mathbf{w} \leftarrow \mathbf{w} - l_t^{\mathbf{w}} \mathbf{w}$
  else
    $\mathbf{w} \leftarrow \mathbf{w} - l_t^{\mathbf{w}}(\mathbf{w} - CNy^i\mathbf{x}^i)$
  end if
end while

Algorithm 2 Stochastic gradient descent for $\bar{\mathbf{s}}$ (w fixed)

while $t = 1 \dots T$ **do**
  Specify learning rate $l_t^{\bar{\mathbf{s}}}$ for iteration t
  Choose a random training image I^i
  Calculate the saliency map $\mathbf{s}^i$
  if $y^i f(\mathbf{x}_i, \mathbf{w}) \geq 1$ **then**
    $\bar{\mathbf{s}} \leftarrow \bar{\mathbf{s}} - l_t^{\bar{\mathbf{s}}} \gamma(\bar{\mathbf{s}} - \mathbf{1})$
  else
    $\bar{\mathbf{s}} \leftarrow \bar{\mathbf{s}} - l_t^{\bar{\mathbf{s}}}(\gamma(\bar{\mathbf{s}} - \mathbf{1}) + 2CNy^i\lambda(\bar{\mathbf{s}} - \mathbf{s}^i))$
  end if
end while

the magnitude of $\bar{\mathbf{s}}$ we can change the image scores (as the saliency maps are multipliers in the score function). Thus, we can decrease the objective value without making any generalizable progress. To control such scaling we augment the objective function with a regularization term for $\bar{\mathbf{s}}$, which penalizes deviation from a uniform map which assigns unit weight to each cell similar to the (individual levels of) standard spatial pyramid, as

$$L(\mathbf{w}, \bar{\mathbf{s}}) = \frac{1}{2}||\mathbf{w}||^2 + \frac{\gamma}{2}||\bar{\mathbf{s}} - \mathbf{1}||^2 + C\sum_i \max(0, 1 - y^i f(\mathbf{x}^i)). \tag{5}$$

We now have one more parameter, $\gamma > 0$, to control the regularization of $\bar{\mathbf{s}}$. As the scales of $\bar{\mathbf{s}}$ and w are different we can not expect similar regularization w.r.t. loss, *i.e.* parameter C to work for both. Thus the model has three parameters for controlling different regularizations γ, C, λ.

The parameter C (cf. the standard SVM parameter) and γ control the relative trade-offs between constraint violation, margin maximization and regularization of $\bar{\mathbf{s}}$. The parameter λ controls the regularization of the saliency map for each image. To gain some more intuition about the parameter λ, consider the two limiting cases. In the first limiting case, when $\lambda \to \infty$, we have a highly smoothed model which forces all saliency maps to be the same as the generic saliency. In the other limiting case, when λ is zero, we have no smoothing and the saliency maps put all the weight on the best scoring cell per image.

2.4. Solving the optimization problem

We solve the problem with a block coordinate descent algorithm. We treat $\mathbf{w}$ and $\bar{\mathbf{s}}$ as two blocks of variables and alternately optimize on one while keeping the other fixed. Fig. 3b illustrates the learning process. In each of the inner iterations we optimize using stochastic gradient descent as detailed in Algorithms 1 and 2, where we use (the stochastic approximations of) the sub-gradient w.r.t. $\mathbf{w}$,

$$\nabla_{\mathbf{w}} L = \mathbf{w} + C\sum_i g_{\mathbf{w}}(\mathbf{x}^i) \tag{6}$$

$$g_{\mathbf{w}}(\mathbf{x}^i) = \begin{cases} 0 & \text{if } \; y^i F(\mathbf{x}^i) \geq 1 \\ -y^i\mathbf{x}^i & \text{otherwise,} \end{cases} \tag{7}$$

and sub-gradient w.r.t. $\bar{\mathbf{s}}$,

$$\nabla_{\bar{\mathbf{s}}} L = \gamma(\bar{\mathbf{s}} - \mathbf{1}) + C\sum_i g_{\bar{\mathbf{s}}}(\mathbf{x}^i) \tag{8}$$

$$g_{\bar{\mathbf{s}}}(\mathbf{x}^i) = \begin{cases} 0 & \text{if } \; y^i F(\mathbf{x}^i) \geq 1 \\ 2y^i\lambda(\bar{\mathbf{s}} - \mathbf{s}^i) & \text{otherwise.} \end{cases} \tag{9}$$

While keeping $\bar{\mathbf{s}}$ fixed we get a semi convex LSVM-like optimization [8] for $\mathbf{w}$. Unfortunately, that is not the case for the optimization of $\bar{\mathbf{s}}$ as, with $\mathbf{w}$ fixed, the hinge loss for each example is concave w.r.t. $\bar{\mathbf{s}}$ (the coefficient of $\bar{\mathbf{s}}^T\bar{\mathbf{s}}$ is $-\lambda < 0$). Thus, the total hinge loss (being the maximum over one convex *i.e.* zero function, and multiple concave functions *i.e.* per example hinge losses) is, in general, non convex and the algorithm will converge to a local minimum for $\bar{\mathbf{s}}$. To make sure that it does not end up in a very bad local minimum, we initialize $\mathbf{w}$ with a perturbed version of that learned using the baseline SVM (same optimization with all components of $\bar{\mathbf{s}}$ and $\{\mathbf{s}^i | I^i \in \mathcal{I}\}$ fixed to 1). Since we are directly minimizing the primal we can expect approximations to generalize reasonably [2]. In practice, we find that the models computed by our implementation perform well.

Parameters. We find initial learning rates $l_0^{\mathbf{w}}$ and $l_0^{\bar{\mathbf{s}}}$ by performing preliminary experiments on a subset of the full data and then we decrease the learning rates every iteration by dividing by the iteration number i.e. $l_t = l_0/t$ (as is common with stochastic gradient methods). We fix $C = 1$ for all experiments (this gives similar results on average as with C obtained by cross validation) and select λ and γ by cross validation on the training data.

Nonlinearizing using a feature map. Recent progress in explicitly computing the feature maps [28] induced by different non linear kernels allows us to address non linearity. The approach is to apply the non linear map to compute the feature vectors explicitly, and work with linear algorithms in the feature space.

We transform the histograms by taking their element-wise square roots *i.e.* $\phi(h) = \sqrt{h}$. It is known [28] that the product of the resulting vectors is equal to the Bhattacharyya kernel between the original histograms. Hence, using the feature map is equivalent to working with the non linear Bhattacharyya kernel, which has been shown to give better results than the linear kernel. We L1 normalize the original histograms so that the feature mapped vectors are L2 normalized.

3. Experimental results

We evaluate our method on three challenging datasets for (i) human action classification in still images [5], (ii) fine grained classification of humans playing musical instruments vs. holding them [32] and (iii) scene classification [17]. We first give the details of our implementation and baselines and then proceed to present and discuss the results on the three datasets.

Bag-of-features. Like previous works [4, 32] we densely sample grayscale SIFT features at multiple scales. We use a fixed step size of 4 pixels and use square patch sizes at 7 scales ranging from 8 to 40 pixels. We learn a vocabulary of size 1000 using k-means and assign the SIFT features to the nearest codebook vector (hard assignment). We use the VLFeat library [26] for SIFT and k-means computation.

Spatial pyramid (SP and overlapping SP). We use a four level spatial pyramid but instead of the usual non overlapping cells with uniform grids we expand the cells by 50% and let them overlap *i.e.* 2×2 cells are $3/4$ of the height (width) instead of $1/2$. We found that doing so provides better statistics (less sparse histograms) for finer cells and improves performance. This is inspired by the idea of 'non sparsification' of vectors [24]. We discuss this more in Sec. 3.4. Our initial experiments gave similar results with classifiers trained on the full pyramid descriptor and the weighted sum of descriptors from each level. We train classifiers for each level separately and combine levels, for the baselines as well as our method, by the weighted sum of classifier scores. The weights sum to one over all levels and are higher for finer resolution levels similar to previous work [17].

Baselines. We use SP and overlapping SP, as baselines, with linear SVM trained without our saliency model *i.e.* we fix all the saliency maps to be uniform in the optimization reducing it to standard linear SVM with spatial BoF. The baseline results are obtained with the liblinear [7] library.

Performance measure. The performance is evaluated based on average precision (AP) for each class and the mean average precision (mAP) over all classes.

Table 1. Results (AP) on actions dataset (Sec. 3.1)

	Per-obj	Baselines		
	inter. [5]	SP [17]	ov. SP	Ours
inter. w/ comp.	56.6	49.4	57.8	**59.7**
photographing	37.5	41.3	39.3	**42.6**
playingmusic	72.0	74.3	73.8	**74.6**
ridingbike	**90.4**	87.8	88.4	87.8
ridinghorse	75.0	73.6	80.8	**84.2**
running	**59.7**	53.3	55.8	56.1
walking	57.6	**58.3**	56.3	56.5
mAP	64.1	62.6	64.6	**65.9**

3.1. Willow actions

Willow actions[1] [4] is a challenging database for action classification on unconstrained consumer images downloaded from the internet. It has 7 classes of common human actions *e.g.* 'ridingbike', 'running'. It has at least 108 images per class of which 70 images are used for training and validation and rest are used for testing. The task is to predict the action being performed given the human bounding box. Like previous work [5], we expand the given person bounding boxes by 50% to include some contextual information.

Fig. 5a shows example images and their saliency maps obtained with our model and Tab. 1 gives quantitative results on the Willow actions dataset. Our implementation of the baseline spatial pyramid [17] achieves an mAP of 62.6% while that of a spatial pyramid with overlapping cells improves by 2%. Our model obtains 65.9% which is the state-of-the art result on this dataset. To compare with previous works, Delaitre et al. [5] obtain an mAP of 64.1% with a method modeling person-object interactions. Note that they model complex interactions between objects and body parts while using external data to train the several object and body part detectors.

Our method gives best results for four out of seven categories. The most significant improvement is obtained on the 'ridinghorse' class which has a strong spatial bias for horse and grass in the bottom part of the image. The saliency map modeling effectively exploits this (Fig. 5a). The drop on 'ridingbike' class can be explained by the limitation of the method to improve performance if the classifier is able to separate the training data almost perfectly and if there is not enough training data (Sec. 3.4).

3.2. People playing musical instruments

People playing musical instruments (PPMI)[2] [32] is a dataset emphasizing subtle difference in interactions between humans and objects (fine grained classification). It contains classes with humans interacting with *i.e.* either

[1] http://www.di.ens.fr/willow/research/stillactions/
[2] http://ai.stanford.edu/∼bangpeng/ppmi.html

Table 2. Results (mAP) on PPMI dataset (Sec. 3.2)

(a) Task 1: 24 class multi-class classification task

Grouplet [31]	Rn. forest [32]	Baselines SP [17]	ov. SP	Ours
36.7	47.0	45.3	46.6	**49.4**

(b) Task 2: 12 binary classification tasks

Grouplet [31]	Rn. forest [32]	Baselines SP [17]	ov. SP	Ours
85.1	**92.1**	89.2	90.3	91.2

Table 3. Results (mAP) on Scene 15 dataset (Sec. 3.3)

Pyramid level comb.	Baselines SP [17]	ov. SP	Ours
1	74.9 ± 0.5	74.9 ± 0.5	-
1+2	77.9 ± 0.4	78.8 ± 0.5	$\mathbf{85.1 \pm 1.2}$
1+2+3	81.8 ± 0.6	82.6 ± 0.4	$\mathbf{85.5 \pm 0.6}$
1+2+3+4	81.9 ± 0.5	81.9 ± 0.3	$\mathbf{84.6 \pm 0.7}$

playing or just holding, 12 different musical instruments. There are two tasks for this dataset (i) 24 class classification with each class being the human playing and holding the 12 instruments and (ii) 12 binary classifications for human playing vs. holding the instruments.

Fig. 5b shows some example images and their saliency maps and Tab. 2 shows our results on the PPMI datasets for 24 class multi-class classification (*Task 1*) and 12 binary classification problems (*Task 2*) respectively. For *Task 1* the spatial pyramid baseline achieves 45.3% and the overlapping spatial pyramid achieves 46.6% improving by 1.3%. Our method achieves a mAP of 49.4% which is state of the art for the dataset. In comparison to previous methods, we improve by 12.7% compared to Yao et al.'s Grouplet [31] and by 2.4% compared to their Random Forest classifier [32]. For *Task 2* the baselines are at 89.2% and 90.3% while our method achieves 91.2% compared to 85.1% of Grouplet [31] and 92.1% of Random Forest classifier [32]. The Grouplet method uses patches at only one scale which can perhaps explain its lower performance. Note that the Random Forest classifier has a much higher complexity than our approach, as it uses 100 decision trees. At each node of the tree they evaluate a linear SVM decision thus effectively performing 100s of vector dot products, whereas our approach only has one such computation. We perform slightly worse that the state of the art in *Task 2* due to performance saturation, see Sec. 3.4 for a discussion.

3.3. Scene 15

Scene 15[3] [17] is a dataset containing 15 scene categories, *e.g.* 'beach', 'office', with 4485 images. The task is multi-class classification with the dataset split into 100 random images per class for training and the rest for testing. Like previous works, we repeat the experiment 10 times and report the mean and standard deviation.

Fig. 5c shows some example images and their saliency maps and Tab. 3 show our results on the scene 15 dataset for 15 binary one-vs-rest classification problems. Our traditional and overlapping spatial pyramid baselines achieve a performance of 81.8% and 82.6% resp. for 3 levels. Our method achieves 85.5% improving the better baseline by 2.9%. It is interesting to note that our method at a low pyramid level of 2 already beats the best baseline obtained at a higher pyramid level of 3, which indicates a coarse spatial bias in the dataset. The state-of-the-art method on this dataset [30] achieves 88.1% (mean class accuracy). However, they combine 14 different low level features. Our best result is comparable to Krapac et al. [16], who used a similar setup as ours and achieved mAP of 85.6%. Note that they quantized features using discriminatively trained decision trees outperforming k-means based quantization. In the current paper, we have used k-means and arguably our results would improve further using similar stronger quantization instead.

3.4. Overlapping cells and training saturation

We use overlapping cells for the spatial pyramid decomposition. As noted by Perronnin et al. [24], when sparseness of the vectors increases, the performance of linear SVM decreases. This is because the more robust distance with sparse vectors is L1 while linear SVM corresponds to the L2 distance. To decrease the effect of sparsity we take overlapping cells in the spatial pyramid partition by increasing the sizes of the cells by 50%. Fig. 4 (left) shows the performances for different codebook sizes on the Willow actions dataset. We notice that for larger codebook sizes of 500 and 1000 the overlapping SP performs better than the non overlapping one but the difference is not significant for a codebook size of 100. As the codebook size increases, but the number of features stays the same, the sparsity of the histogram increases. Thus, pooling more features by increasing the size of the cells performs better, as the sparsity of the histograms is decreased.

We can also observe that our approach does not gain much when the training data is well separated *i.e.* the baseline SVM is saturated. This can occur when there is not enough training data or the task is relatively easy. In saturated cases the number of vectors within the margin—which effectively contribute towards refining the hyperplane—is small (< 100) and the saliency model is not able to derive additional information from so few examples. Fig. 4 (right) shows the performance for the different pyramid combinations for the Willow actions dataset. We observe that as the pyramid level increases, the gap between the baselines and the proposed method decreases due to increase in train-

[3] http://www-cvr.ai.uiuc.edu/ponce_grp/data/

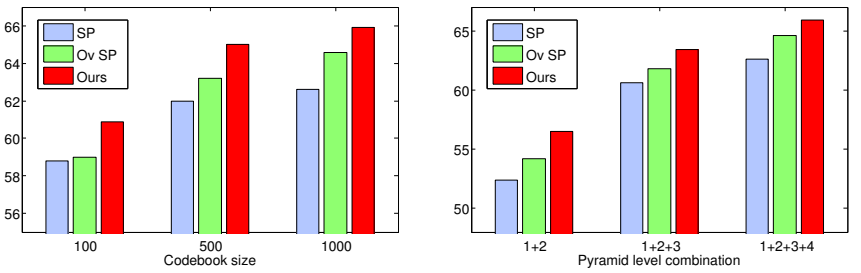


Figure 4. (Left) Evaluation (mAP) of the impact of the codebook size for a full pyramid representation. (Right) Evaluation (mAP) of the impact of the pyramid levels for a codebook size of 1000. The dataset is the Willow Actions [4].

Figure 5. Example images and their saliency maps (8×8 resolution) for images from two classes for each of the three databases (higher values are brighter). Notice how the maps adapt to the content of the image and highlight the spatially salient regions per image.

ing saturation. The trend is similar for increasing codebook size, Fig. 4 (left). This also explains why we get little or no improvement for the 'ridingbike' class (Tab. 1) and the *Task 2* of PPMI dataset (Tab. 2b).

3.5. Qualitative results

Fig. 5 shows example images from two classes for each of the three datasets together with their saliency maps. We can observe that the saliency maps focus on those parts of the images which we expect to be discriminative. For example, in the action class 'ridinghorse' the saliency maps give high weights to the lower regions which are expected to be salient as they contain the horse and grassy texture which are highly correlated with the class. The person (in the typical riding pose) is not weighted highly, because it might be confused with 'ridingbike', stressing the discriminative nature of the maps.

Furthermore, per image adaptation can be seen in all the examples. In the 'playingmusic' class the maps follow the hands and the musical instruments and differ for every image. A similar observation holds for 'tallbuilding' class where the middle part of the buildings seems to be more discriminative probably because of predominant sky in the upper part of many images.

The correlation between the locality of the task and the peaks in the maps is also clearly visible. A strong contrast is apparent between the 'playingmusic' class of the Willow actions dataset and the similar 'violin' and 'erhu' classes of PPMI dataset. In the actions dataset the discrimination is against more general actions ('running', 'photographing' *etc*.) and hence the maps capture the instrument, the pose of the hands *etc*. and have relatively spread out maxima. In contrast, for 'violin' and 'erhu' classes the maps have sharp peaks as the task is to differentiate between holding vs. playing instruments. The maps here quite accurately focus on the region of discriminative interaction between the person and the instrument.

4. Conclusion

We have presented a method for learning discriminative spatial saliency for images to improve the image representation and, thus, the classification performance. The method has wide applicability as was demonstrated with experiments on three challenging datasets. The method adapts saliency per image and focuses on regions which are salient for the given task. It improves over a baseline without spatial saliency and achieves better or comparable results w.r.t. the state of the art.

We plan to investigate fusing multiple discriminative spatial saliency maps obtained from various low level feature channels corresponding to different cues *e.g.* shape, color, texture or even high level concepts and attributes, appropriate for the classes.

Aknowledgement. This work was funded by the ANR, grant reference ANR-08-SECU-008-01/SCARFACE.

References

[1] H. Bilen, V. Namboodiri, and L. Van Gool. Object and action classification with latent variables. In *BMVC*, 2011. 2, 3

[2] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007. 4

[3] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Intl. Workshop on Stat. Learning in Comp. Vision*, 2004. 2

[4] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: A study of bag-of-features and part-based representations. In *BMVC*, 2010. 2, 5, 7

[5] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *NIPS*, 2011. 2, 5

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html. 2

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 5

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 2, 3, 4

[9] D. Gao and N. Vasconcelos. Discriminant saliency for visual recognition form cluttered scenes. In *NIPS*, 2004. 1, 2

[10] D. Gao and N. Vasconcelos. Integrated learning of saliency, complex features and object detectors from cluttered scenes. In *CVPR*, 2005. 1, 2

[11] T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative spatial pyramid. In *CVPR*, 2011. 2

[12] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009. 3

[13] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence*, 1998. 1, 2

[14] F. S. Khan, J. van de Weijer, and M. Vanrell. Top-down color attention for object recognition. In *ICCV*, 2009. 1, 2

[15] C. Koch and S. Ullman. Shifts in selective visual attention: Towards underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985. 1, 2

[16] J. Krapac, J. Verbeek, and F. Jurie. Learning tree-structured descriptor quantizers for image categorization. In *BMVC*, 2011. 6

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2, 5, 6

[18] D. Lowe. Distinctive image features form scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, 2004. 2

[19] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Intl. Journal of Computer Vision*, 60(1):63–86, 2004. 2

[20] F. Moosmann, D. Larlus, and F. Jurie. Learning saliency maps for object categorization. In *ECCV Workshops*, 2006. 1, 2

[21] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga. Saliency estimation using a non-parametric low level vision model. In *CVPR*, 2011. 1

[22] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006. 2

[23] D. Parikh, L. Zitnick, and T. Chen. Determining patch saliency using low-level context. In *ECCV*, 2008. 1, 2

[24] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010. 5, 6

[25] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980. 1, 2

[26] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 5

[27] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 3

[28] A. Vedaldi and A. Zisserman. Efficient additive kernels using explicit feature maps. In *CVPR*, 2010. 4, 5

[29] M. Wang, J. Konrad, P. Ishwar, K. Jing, and H. Rowley. Image saliency: From intrinsic to extrinsic context. In *CVPR*, 2011. 1

[30] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large scale scene recognition from abbey to zoo. In *CVPR*, 2010. 6

[31] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010. 2, 6

[32] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011. 2, 5, 6