Object of Interest Detection by Saliency Learning

Pattaraporn Khuwuthyakorn[1,3], Antonio Robles-Kelly[1,2], and Jun Zhou[1,2]

[1] RSISE, Australian National University, Canberra, ACT 0200, Australia
[2] National ICT Australia (NICTA*), Canberra, ACT 2601, Australia
[3] CRC for National Plant Biosecurity , Canberra, ACT, 2617, Australia

Abstract. In this paper, we present a method for object of interest detection. This method is statistical in nature and hinges in a model which combines salient features using a mixture of linear support vector machines. It exploits a divide-and-conquer strategy by partitioning the feature space into sub-regions of linearly separable data-points. This yields a structured learning approach where we learn a linear support vector machine for each region, the mixture weights, and the combination parameters for each of the salient features at hand. Thus, the method learns the combination of salient features such that a mixture of classifiers can be used to recover objects of interest in the image. We illustrate the utility of the method by applying our algorithm to the MSRA Salient Object Database.

1 Introduction

Saliency map is an important tool in vision research [1]. Each pixel in this map is assigned with a measure of "relevance" or "importance" so as to reflect the degree to which a region in the image is attractive to visual attention. The research on visual saliency has generated a vast literature in computer vision and found applications in many areas, such as region of interest extraction [2], segmentation [3], tracking [4], object detection [5], thumbnailing [6] and image retrieval and classification [7].

It has been widely accepted that visual saliency computation can be effected in a bottom-up manner [8–11]. Departing from this strategy, Itti et al. [9] proposed a computational framework for visual saliency which decomposes visual input into component feature maps. In [12], Alter and Basri used image edges to construct the saliency map. The work in [12] is in line with the common approach to model contour or curve saliency, where length and smoothness of the edge points are often used [13, 14].

The combination of individual features into saliency maps can be greatly influenced by the behavioral goal of human attention [15]. This can be considered as a top-down modulation mechanism [16]. Note that, when guided by

* NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

observer preferences, those parts that are less related to the visual targets of visual attention can be assigned smaller contributions on the saliency map or even completely ignored. To model this process, Navalpakkam and Itti [17] proposed a method to maximise the signal-to-noise ratio between the mean salience of the target and that of the distractor. Berengolts and Lindenbaum [14] also proposed a method to recover the distribution of the edge lengths and curvature on the region corresponding to the target of interest making use of labelled objects. In [18], saliency maps were computed as a linear combination of features whose weights were recovered through a linear regression model applied to manually labeled images. Liu et al. [2] formulated the saliency detection problem as a region of interest segmentation task where learning is performed via a conditional random field.

Note that, in some of the methods above, the same features at different scales are added together in a linear fashion [9, 2] or modelled in a scale-space setting [19]. This suggest that salient objects or regions with different sizes may generate the same contribution to the final saliency map. Moreover, the intrinsic relationships between the individual features is often overlooked. This is due to the fact that, in existing methods, the optimisation step treats the features as independent primitives, despite the fact that they may actually be interrelated or highly correlated. This is even more important since, in the case of saliency features, we often deal with a large sample size with moderate feature dimension. Thus, for purposes of saliency learning, the features may span a space which is nonlinear in nature. This is in contrast with other settings in computer vision where linear classifiers can be applied on high dimensional features.

Hence, in this paper, we present a method which aims at combining salient features through a structured learning characterisation of the problem so as to achieve two desirable properties. Firstly, recovering a classifier model with the efficiency of linear Support Vector Machines. Secondly, reaching the discrimination power of nonlinear classifiers. To do this, we adopt a divide-and-conquer strategy that exploits partitioning the feature space into regions that are linearly separable. This is effected through a mixture of Support Vector Machines (SVMs) where the mixture weights and the feature combination coefficients are optimised using an Expectation-Maximisation (EM) approach. The method presented here is quite general in nature and can accommodate a number of saliency features found in the literature. In our work, we make use of the multi-scale features in [9] and [2], and present their natural extensions to neighbourhood-based descriptors.

2 Structured Learning

As mentioned earlier, our object of interest detection method makes use of saliency features and structured learning. The structured learning approach hinges in the notion that non-linear classification can be effected in a piecewise-linear manner across the feature space. This provides a means to efficiency through the use of linear classifiers while preserving the flexibility of non-linear

methods. Our probabilistic formulation employs two ingredients. The first one is the prior probability of the mixture given a feature-set at a pixel-site on the image. The second ingredient is the posterior probability corresponding to the outputs for each of the linear SVMs.

2.1 Mixture of SVMs

In this section, we cast the recovery of the saliency map into a structured learning setting. The aim is to combine the saliency features so as to perform classification, i.e. separate salient objects from the background in the image, based upon objects of interest provided as training data. Here, we formulate the problem in terms of a generative model over the training data. This joint distribution model enables us to explicitly incorporate mixture coefficients into the likelihood function. Consequently, we can perform parameter learning and model selection simultaneously by imposing a proper prior on the mixture co-efficients based on the minimum message length (MML) criterion [20]. Parameter update is then achieved making use of the EM algorithm [21]. For model selection, we start with an overcomplete model and automatically prune vanishing SVM mixture co-efficients. Hence structured learning is implicitly incorporated into the optimisation process and performed in a top-down manner.

To commence, consider a set of M tuples $(X, Y) = \{(\mathbf{x}_{i,l}, y_i)|i = 1, \ldots, M, y_i \in \{-1, 1\}\}$, where $(\mathbf{x}_{i,l}, y_i)$ are the i^{th} data-label pair in the training data corresponding to the l^{th} saliency feature, where the total number of salient feature is N. In practice, Y accounts for the corresponding object of interest regions provided at input. The linear SVM classifier solves the following optimisation problem

$$\min_{\mathbf{w}} \quad \frac{||\mathbf{w}||^2}{2} + C \sum_i \epsilon(\mathbf{w}; \mathbf{x}_{i,l}, y_i) \tag{1}$$

where $\epsilon(\mathbf{w}; \mathbf{x}_{i,l}, y_i) = \max(1 - y_i \mathbf{w}^T \mathbf{x}_{i,l}, 0)$ is the Hinge loss function which specifies an upper bound on the classification error. The first term on the right hand side is regularisation term on classifier weights. Without loss of generality, we have subsumed the bias term b in the above formulation by appending each data instance with an additional dimension $\mathbf{x}_{i,l}^T = [\mathbf{x}_{i,l}^T, 1]$ and $\mathbf{w}^T = [\mathbf{w}^T, b]$.

We can extend the SVM model above to a two-layer mixture model formulated using the joint probability distribution over the salient regions provided by the user and the SVM binary classifier. The model, hence, consists of two parts. The hidden layer, which is composed of the gating network that produces a soft-partition of the input space by generating a data-dependent weight distribution. Each node in the hidden layer is connected to a linear SVM classifier in the input layer, which is responsible for the salient object recovery.

We establish the link between the proposed mixture model and the associated generative model using the joint probabilistic distribution over the data in X

and the labels in Y given by

$$P(Y|X,\Theta) = \prod_i P(y_i|\mathbf{x}_{i,l},\Theta) = \prod_i \sum_{z_i} P(y_i|z_i,\mathbf{x}_{i,l},\Theta)P(\mathbf{x}_{i,l}|z_i,\Theta)P(z_i \mid \Theta) \tag{2}$$

where i indexes data samples as before, $\Theta = \{\alpha, \beta, \tau, \gamma\}$ are the parameters of the underlying model and z_i is the hidden variable introduced for the ith sample for each of the N salient features under study. In the equation above, α and β are the multinomial parameters that generate the hidden variables z_i's whereas τ and γ are parameters for the gating nodes and classifiers, whose specific parametric forms will be explained later. The probability $P(\mathbf{x}_{i,l}|z_i,\tau)$ represents the posterior for the mixture component with hyperparameters τ, and $P(y_i|z_i,\mathbf{x}_{i,l},\gamma)$ is the posterior probability of corresponding linear SVM output for the ith sample.

It is worth noting that our mixture of SVMs model can also be viewed from the perspective of graphical model due to its generative nature. From this viewpoint, $\mathbf{x}_{i,l}$ and y_i are the target random variables whose joint distributions are to be modeled, and z_i is the hidden variable generated from a multinominal distribution with parameters $\alpha = \{\alpha_1, \ldots, \alpha_K\}$ and $\beta = \{\beta_1, \ldots, \beta_N\}$ for K-mixtures and N features. Thus, $\mathbf{x}_{i,l}$ is generated from an isotropic Gaussian distribution with parameter τ conditional on z_i, where $\tau = \{(\mu_{1,1}, \Sigma_{1,1}), \ldots, (\mu_{K,N}, \Sigma_{K,N})\}$ and $\mu_{j,l}$ and $\Sigma_{j,l}$ are the mean vector and the variance for the jth mixture component performing inference upon the saliency feature-set indexed l. The target random variable y_i is generated from a probabilistic classifier model with parameter γ conditional on $\mathbf{x}_{i,l}$ and z_i, where $\gamma = \{\mathbf{w}_{1,1}, \ldots, \mathbf{w}_{K,N}\}$, and $\mathbf{w}_{j,l}$ is the classifier weight-vector for the jth linear SVM corresponding to the l^{th} saliency feature-set. This yields

$$P(Y|X,\Theta) = \prod_i \sum_{z_i} P(y_i|\mathbf{x}_{i,l},\gamma)P(\mathbf{x}_{i,l}|z_i,\tau)P(z_i \mid \alpha,\beta) \tag{3}$$

The proposed model bears some resemblance with the mixture of experts (HME) model proposed by Jacobs and Jordan [22]. Nonetheless, they are inherently different in nature in the sense of the probabilistic distributions they capture. Our model captures the joint distribution of data and labels, whereas the HME model is associated with the conditional probability distribution of labels given the data. In the HME model, the hidden variable z_i is generated from a conditional probability distribution while in our method it arises from a multinominal distribution with parameter α. This enables us to control the complexity of the model implicitly by enforcing proper sparseness priors on α.

Equation 2 suggests parameter estimation can be effected via Maximum Likelihood Estimation (MLE) by maximising the following log-likelihood function

$$\begin{aligned} \mathcal{L}(\Theta) &= \sum_i \log P(y_i|\mathbf{x}_{i,l},\Theta) + \sum_j \Omega(\mathbf{w}_{j,l}) \\ &= \sum_i \log \Big\{ \sum_l \beta_l \sum_j \alpha_j P(y_i|\mathbf{x}_{i,l},\mathbf{w}_{j,l})P(\mathbf{x}_{i,l}|z_i,\tau) \Big\} + \sum_j \Omega(\mathbf{w}_{j,l}) \end{aligned} \tag{4}$$

where $\Omega(\mathbf{w}_{j,l}) = \log\{P(\mathbf{w}_{j,l})\}$ is a log-prior term for regularisation purposes. The last line follows from Equation 3, the definition of $\gamma = \{\mathbf{w}_{1,1}, \ldots, \mathbf{w}_{K,N}\}$ and the use of the shorthand $P(z_i \mid \alpha, \beta) = \alpha_j \beta_l$ for the j^{th} mixture and the l^{th} salient feature-set. This responds to the fact that here, we view $P(z_i \mid \alpha, \beta)$ as a data-independent term which specifies the prior probability of the mixture and salient feature pair at a given pixel-site on the image.

In order to incorporate the linear SVM into the log-likelihood above, we view the associated constrained quadratic optimisation problem corresponding to the negative log-likelihood from a probabilistic veiwpoint. Note that the second term on the right hand side is related to the prior $\Omega(\mathbf{w})$, whereas the first term corresponds to the conditional probability $P(y|\mathbf{x}, \mathbf{w})$ related to classification errors. These are given by

$$\Omega(\mathbf{w}_{j,l}) = -\zeta ||\mathbf{w}_{j,l}||^2 \tag{5}$$

$$P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l}) = e^{-\epsilon(\mathbf{w}_{j,l}; \mathbf{x}_{i,l}, y_i)} \tag{6}$$

Here we have omitted the normalisation factor for the conditional probability $P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l})$, which leads to an approximation of the probability measure. This is mainly due to the consideration regarding the use of numerical optimisation which enables us to employ existing fast linear SVM solvers [23] for parameter estimation. This simplification is still valid in the large margin case where the probability of the negative class is usually very small. More importantly, the likelihood function in Equation 4 is guaranteed to increase using the EM algorithm, as we discuss in the next section, regardless of whether or not $P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l})$ is a proper probability measure over y_i.

2.2 The EM Algorithm

In this section, we describe an EM algorithm for solving the mixture of linear SVMs presented in the previous section. The E-step updates the posterior probability of assigning each sample to the component classifiers. Let $\Theta^{(t)} = \{\alpha_j^{(t)}, \beta_l^{(t)}, \mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)}, \mathbf{w}_{j,l}^{(t)} | j = 1, \ldots, K; l = 1, \ldots, N\}$ be the parameters at the current iteration, the probability of the ith sample given the j^{th} classifier and the l^{th} saliency feature is given by

$$q_{i,j,l}^{(t+1)} = \frac{\alpha_j^{(t)} \beta_l^{(t)} P(\mathbf{x}_{i,l}|\mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)}) P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})}{\sum_s \sum_u \sum_v \alpha_u^{(t)} \beta_v^{(t)} P(\mathbf{x}_{s,v}|\mu_{u,v}^{(t)}, \Sigma_{u,v}^{(t)}) P(y_s|\mathbf{x}_{s,v}, \mathbf{w}_u^{(t)})} \tag{7}$$

where $s \in \{1, \ldots, M\}$, $u \in \{1, \ldots, K\}$, $v \in \{1, \ldots, N\}$. $P(y_i|\mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})$ is given by Equation 6, and $P(\mathbf{x}_{i,l}|\mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)})$ is given by the following multivariate, d-dimensional Gaussian distribution,

$$P(\mathbf{x}_{i,l}|\mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)}) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma_{j,l}^{(t)} \mid}} \exp\left(-\frac{1}{2}(\mathbf{x}_{i,l} - \mu_{j,l}^{(t)})^T \left(\Sigma_{j,l}^{(t)}\right)^{-1} (\mathbf{x}_{i,l} - \mu_{j,l}^{(t)})\right) \tag{8}$$

The M-step involves simultaneously updating the parameters for the gating nodes and SVM classifiers so as to solve two independent optimisation problems. Parameter estimation for the gating nodes is similar to the estimation of parameters for the Gaussian mixture model. Specifically, for the jth mixture component and lth saliency feature we have

$$\alpha_j^{(t+1)} = \frac{\sum_s \sum_v q_{s,j,v}^{(t+1)}}{\sum_s \sum_u \sum_v q_{s,u,v}^{(t+1)}} \tag{9}$$

$$\beta_l^{(t+1)} = \frac{\sum_s \sum_u q_{s,u,l}^{(t+1)}}{\sum_s \sum_u \sum_v q_{s,u,v}^{(t+1)}} \tag{10}$$

$$\mu_{j,l}^{(t+1)} = \frac{\sum_s q_{s,j,l}^{(t+1)} \mathbf{x}_{s,l}}{\sum_s q_{s,j,l}^{(t+1)}} \tag{11}$$

$$\Sigma_{j,l}^{(t+1)} = \frac{\sum_s q_{s,j,l}^{(t+1)} (\mathbf{x}_{s,l} - \mu_{j,l}^{(t+1)})^T (\mathbf{x}_{s,l} - \mu_{j,l}^{(t+1)})}{\sum_s q_{s,j,l}^{(t+1)}} \tag{12}$$

As a result, parameter estimation for the linear SVMs reduces itself to updating the classifiers for reweighted samples where the weights are specified by the posterior probabilities computed in the E-step. Specifically, for the j^{th} linear classifier working on the l^{th} saliency feature we solve the following classification problem

$$\begin{aligned} \max \sum_i \sum_l q_{i,j,l}^{(t)} \log P(y_i | \mathbf{x}_{i,l}, \theta_{j,l}) + \log P(\theta_{j,l}) \\ = \max \left\{ - \sum_i \sum_l q_{i,j,l}^{(t)} \epsilon(\mathbf{w}_{j,l}; \mathbf{x}_{i,l}, y_i) - \zeta ||\mathbf{w}_{j,l}||^2 \right\} \end{aligned} \tag{13}$$

where $\theta_{j,l} = \{\alpha_j, \beta_l, \mu_{j,l}, \Sigma_{j,l}, \mathbf{w}_{j,l}\}$ and $C = \dfrac{1}{2\zeta}$. This is exactly the same problem as training linear SVMs in Equation 1 whose sample weights are given by $q_{i,j,l}^{(t)}$.

2.3 Convergence

As mentioned in the sections above, the method proceeds in an iterative fashion. At each iteration t, the method comprises the following steps

- Train the SVMs using the sample weights $q_{i,j,l}^t$ so as to recover the probabilities $P(y_i \mid \mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})$. In practice, this is equivalent to obtaining the probabilistic output of the SVM classifiers as shown in [24].
- With $P(y_i \mid \mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})$ at hand, compute the updated weights $q_{i,j,l}^{t+1}$ in Equation 7. These can be computed making use of the probabilities $P(\mathbf{x}_{i,l} \mid \mu_{j,l}^{(t)}, \Sigma_{j,l}^{(t)})$ given in Equation 8 and the probabilities $P(y_i \mid \mathbf{x}_{i,l}, \mathbf{w}_{j,l}^{(t)})$ recovered in the previous step.

– Recover the remaining parameters making use of Equations 9-12.

It should be noted that each EM iteration increases the log-likelihood given by Equation 4. This argument can be easily established by making use of the auxiliary function parameterised with respect to $\Theta^{(t)}$ given by

$$Q(\Theta;\Theta^{(t)}) = \sum_{i,j,l} q_{i,j,l}^{(t)} \log \alpha_j \log \beta_i P(\mathbf{x}_{i,l}|\mu_{j,l},\Sigma_{j,l}) P(y_i|\mathbf{x}_{i,l},\mathbf{w}_{j,l}) \\ - \sum_i \sum_j \sum_l q_{i,j,l}^{(t)} \log q_{i,j,l}^{(t)} + \sum_j \Omega(\mathbf{w}_{j,l}) \tag{14}$$

which is the lower bound of $\mathcal{L}(\Theta)$ since

$$\mathcal{L}(\Theta) - Q(\Theta,\Theta^{(t)}) = q_{i,j,l}^{(t)} \log \frac{q_{i,j,l}^{(t)}}{q_{i,j,l}} \tag{15}$$

The gap is non-negative and varnishes if and only if $\Theta = \Theta^{(t)}$. Hence, the log-likelihood increases with the following relation

$$\mathcal{L}(\Theta^{(t+1)}) \geq Q(\Theta^{(t+1)},\Theta^{(t)}) \geq Q(\Theta^{(t)},\Theta^{(t)}) = \mathcal{L}(\Theta^{(t)})$$

The second inequality is true due to the maximisation step. Therefore, by repeating the EM steps we can obtain a convergent solution of the original maximum likelihood estimation problem. Moreover, we can stop the iteration presented earlier when the quantity $||\Theta^{(t+1)} - \Theta^{(t)}||$ is less or equal to a predefined threshold ρ.

3 Feature Extraction

So far, we have assumed the saliency features are at hand as input to our mixture of linear SVMs. Here, we elaborate further on the saliency features used in our experiments. It is worth noting that the developments above are general in nature and can be applied to a large variety of saliency features. Here, we depart from the feature map extraction methods by Itti et al. [9] and Liu et al. [2]. We extend these two methods by considering the pixel neighbourhood, which permits capturing the image structure during the feature extraction process. The individual features are then used as the input to our structured learning method.

In the Salient Map (SM) method of Itti et al. [9], an input image is first smoothed using Gaussian filters so as to generate a scale pyramid. Simple features are then extracted at each scale to generate three types of visual cues. The first of these is the intensity feature obtained by averaging the red, green and blue channel-values at each pixel in the input image. By computing the differences between seven scales, 6 intensity channels are recovered. The second set of features is based upon color and simulate the function of the cortex, which is represented by a set of color opponency between red, green and blue channel values against the yellow basis. For each set of colour features, differences are recovered over three scales and, hence, yield 12 channels. The third set is comprised by

orientation features, which are given by the responses of a set of even-symmetric Gabor filters [25]. In practice, these are treated as a Gaussian envelope modulated by a complex sinusoidal carrier. Here, we compute the responses at six scales and four orientations, and thus, recover 24 orientation channels.

The method from Liu et al. [2], which we denote LRG, recovers saliency making use of local, regional and global features. The first of these consists of the local feature extracted from multi-scale contrast. For a given pixel, the image contrast is computed as the sum of the 2-norm grayscale differences between a pixel and its neighborhood. Then, contrast at different scales is combined linearly. To extract the regional salient feature-set, two bounding boxes are used. These cover the proposed salient object and its surrounding area. The differences between the RGB color histograms for the bounding boxes are computed so as to find the optimal center-surround aspect ratio of the object. Finally, the global saliency features are computed from spatial color distributions. This feature can be viewed as that represented by spatial color clusters, where colors with small spatial variance are assigned higher salience.

Despite effective, the features above may be prone to corruption due to noise and cluttered background. Furthermore, small objects may generate scattered salient regions during the feature extraction process. These greatly influence the final object of interest detection step. To solve these problems, we extend the above mentioned features to a neighbourhood-based descriptor setting by considering the interaction of image pixels with the neighboring pixels. Here, we adopt a second-order Markov setting, that is, including the saliency features of the pixels in a 3×3 neighborhood. In this way, we can generate a descriptor at each pixel that contains saliency features from both the pixel itself and its neighborhood. It can be seem in the later experiments that such extension helps maintain the local consistency in the object of interest detection.

4 Experiments

We perform experiments on the Microsoft Research Asia (MSRA) Salient Object Database B, which contains 5,000 images. Details on this database can be found in [2]. Our motivation in using this dataset stems in providing results consistent to those reported in [2] and, thus, presenting a fair comparison with the alternatives reported in the literature. We have randomly divided the images in the database into two groups of 2,500 images each. One of these is used for training and the other one for testing. At training, we set the number of SVMs for our mixture to five, i.e. $K = 5$. The SVM parameters have been recovered by ten-fold cross-validation. For our experiments, we have used four sets of features. The first set is the colour, contrast and center-surround features in [2] (LRG), thus, $N = 3$. The second set comprises the 42 channels generated from orientation, intensity and colour features in [9](SM). In this case, $N = 42$. We have also used the extensions of the features in [9] and [2] with a 3×3 neighbourhood $\mathcal{N}$ about each pixel in the imagery, which we denote SM-$\mathcal{N}$ with $N = 42$ and LRG-$\mathcal{N}$ with $N = 3$, respectively.

To compare the learning performance of our mixture of linear SVMs (MLSVM) with alternatives elsewhere in the literature, we also provide results yielded by the Conditional Random Field (CRF) inference algorithm in [2] and the boosting algorithm ADABOOST$_{REG}$ in [26]. For the CRF algorithm, we have used the parameters in [2], whereas for the ADABOOST$_{REG}$ we have used 10 weak learners with ten-fold cross validation so as to obtain the best set of parameters. For our method, we have set the stoping threshold ρ for the EM iteration to 0.001 and initialised the parameters in Θ as follows. The weights $\alpha_j^{(0)}$ are set to $\frac{1}{K}$, i.e. $\alpha_j^{(0)} = \frac{1}{5}$. Similarly, we have set the feature weights to $\frac{1}{N}$, which yields the value for $\beta_j^{(0)}$. The means $\mu_{j,l}^{(0)}$ and covariances $\Sigma_{j,l}^{(0)}$ have been computed via k-means clustering [27]. To do this, we set $k = 5$ and apply k-means to each of the feature-sets under study. With the cluster members at hand, the corresponding means and covariances are computed.

For purposes of testing, we used the trained model to generate saliency values for each pixel. For the three methods, i.e. our approach, the CRF and the ADABOOST$_{REG}$, the testing output is a saliency map which indicates the probability of a testing pixel being the salient object. To detect a salient object region, we apply the optimal threshold recovery method in [28] on the saliency map. Following [2], we assume that there is only one salient object per image. Here, we extract the region whose size is largest amongst those yielded after the method in [28] is applied. Note that such setting is for the sake of providing an equal comparison with results reported elsewhere rather than a limitation on our method. More than one objects may be obtained by sequentially extracting regions in order of their sizes.

To commence, we show sample results for the results yielded by the 12 classifier-feature pairs used in our experiments (three learning methods against four feature sets). Figure 1 shows some examples of saliency maps recovered by our method and the alternatives for the images on the top-most row. The recovered objects of interest for the images shown in Figure 1 are shown in Figure 2. In the panels, the bounding boxes show the recovered regions after the application of the method in [28] to the saliency maps. Note that, despite the LRG-$\mathcal{N}$ features with the CRF inference produces results comparable to our approach, our method provides bounding boxes more in accordance with the ground truth. This is particularly evident for the coloured wine glasses and the tulip images. Moreover, for other images, such as the log-cabin and the CPU images, the LRG-$\mathcal{N}$ features with the CRF has slightly cropped the objects of interest by delivering smaller bounding boxes.

We now provide a quantitative analysis using a number of performance measures. The first of these is the precision-recall measure in [2]. The precision-recall formulation in [2] takes into account the structure of the database in our experiments by using the binary masks provided as ground truth and the ones delivered by our method and the alternatives. The second of the quantitative measures used here is the F-score [29]. The F-score is defined as $F_\eta = \frac{(1+\eta)precision \times recall}{\eta \times precision + recall}$. Following [30], we have set $\eta = 0.5$, which corresponds to the weighted harmonic mean of precision-recall. Finally, we have used the boundary displacement error

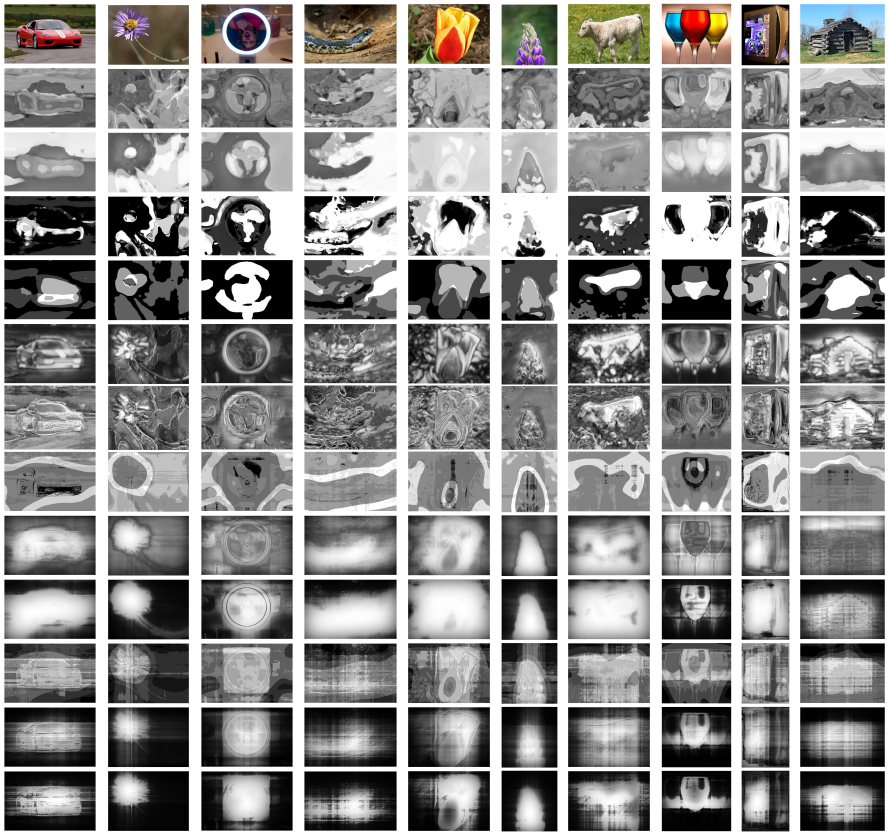

Fig. 1. Saliency map samples computed using different features and learning methods. From top-to-bottom: Ground truth, SM+ADABOOST$_{REG}$, SM+CRF, SM+MLSVM, SM-$\mathcal{N}$+ADABOOST$_{REG}$, SM-$\mathcal{N}$+CRF, SM-$\mathcal{N}$+MLSVM, LRG+ADABOOST$_{REG}$, LRG+CRF, LRG+MLSVM, LRG-$\mathcal{N}$+ADABOOST$_{REG}$, LRG-$\mathcal{N}$+CRF, LRG-$\mathcal{N}$+MLSVM

(BDE) [31]. In our experiments, we have followed [2] and used the fixation area so as to compute our F-score and BDE plots. The fixation area is the smallest rectangle containing a fixed percentage of salient pixels as delivered by our method and the alternatives. As in [2], and so as to provide consistent results to those reported elsewhere, the fixation area has been recovered through exhaustive search.

In Figure 3 we show the overall dataset-average precision-recall plots for the 12 combinations of saliency feature-sets and inference methods used in our experiments. In the figure, for the sake of clarity, we have divided the plots into two panels. On the left-hand-side, we show those plots corresponding to the SM and SM-$\mathcal{N}$ features, whereas the other panels shows the results for the *LRG* and LRG-$\mathcal{N}$ features. Note that our method (MLSVM) performs best with both, the

Fig. 2. Sample object of interest detection results. From top-to-bottom: Ground truth, SM+ADABOOST$_{REG}$, SM+CRF, SM+MLSVM, SM-$\mathcal{N}$+ADABOOST$_{REG}$, SM-$\mathcal{N}$+CRF, SM-$\mathcal{N}$+MLSVM, LRG+ADABOOST$_{REG}$, LRG+CRF, LRG+MLSVM, LRG-$\mathcal{N}$+ADABOOST$_{REG}$, LRG-$\mathcal{N}$+CRF, LRG-$\mathcal{N}$+MLSVM

SM-$\mathcal{N}$ and the LRG-$\mathcal{N}$ features followed by the CRF with LRG-$\mathcal{N}$ features and the ADABOOST$_{REG}$ taking LRG-$\mathcal{N}$ features as input. Note that the varying length of the traces in the plot corresponds to the dependence of the precision-recall measurements upon the fixation area. In our plots, each of the markers corresponds to fixation area variations from 50% to 100% in increments of 5%. As a result, the "flatter" and higher the precision-recall traces in the plot the more stable the classifier-feature pair is to variations of fixation area.

Following the observation that our measures are dependent on fixation area percentages, in Figures 4 and 5 we show the F-scores and BDE as a function of fixation area percentage. As in Figure 3, we have plotted, on the left-hand panels, the traces for the SM and SM-$\mathcal{N}$ features, while the right-hand plots correspond to the *LRG* and LRG-$\mathcal{N}$ feature-sets. On both figures, the neighbourhood-based saliency descriptors are always the best performers, regardless of the inference

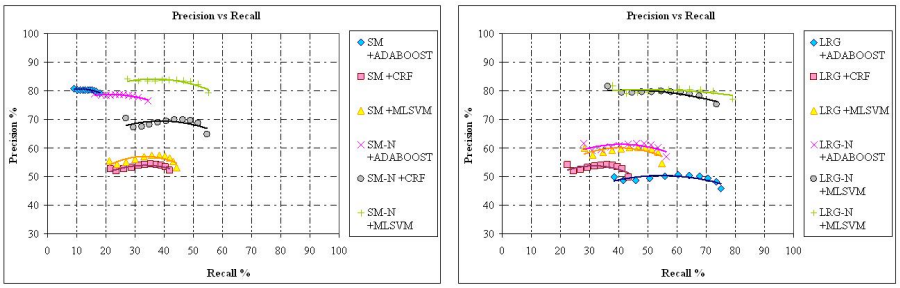

Fig. 3. Average precision-recall.

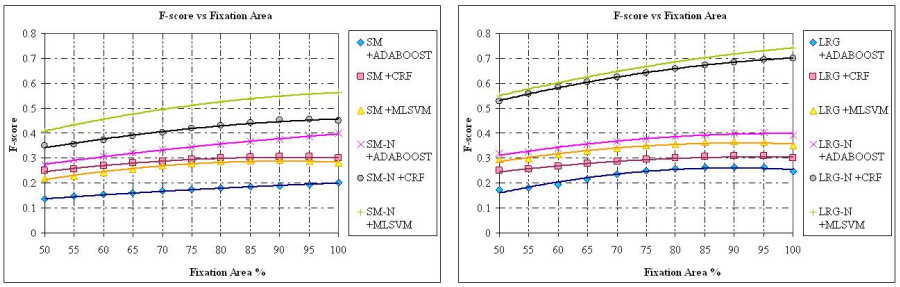

Fig. 4. Average F-score as a function of the fixation area percentage.

method used. In both accounts, the MLSVM with LRG-$\mathcal{N}$ features outperforms the alternatives, with lower BDEs and higher F-scores across the fixation area percentages, with ADABOOST$_{REG}$ consistently delivering the worst results. It is also worth nothing that the LRG based features shows better F-score and BDE results than SM based features. This is consistent with Figures 1, where the topmost six rows, corresponding to the results yielded using the SM and SM-$\mathcal{N}$ features, show regions which are less well defined than the panels in the bottom rows. The notion that the LRG and LRG-$\mathcal{N}$ features provide better performance is confirmed by the F-score results. Nonetheless, for all the quantitative measures in our experiments, the MLSVM provided a margin of advantage over the alternative learning methods.

5 Conclusions

In this paper, we have presented a mixture of Linear SVMs for purposes of learning how to detect a salient object. The method presented here employs a mixture of linear SVMs so as to partition the feature space into sub-regions which are linearly separable. This is a divide-and-conquer approach which allows the

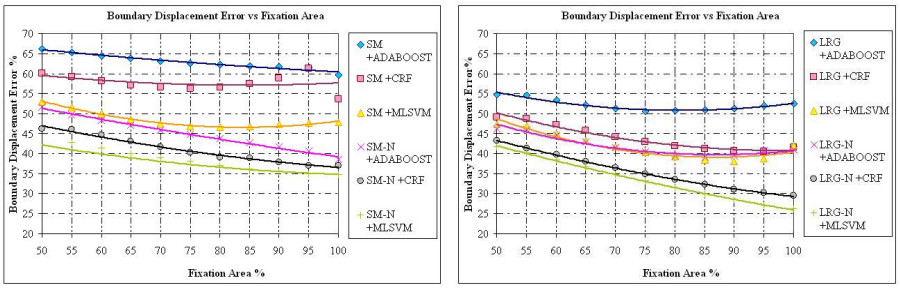

Fig. 5. Boundary Displacement Error as a function of the fixation area percentage.

recovery of the mixture weights and the feature combination coefficients making use of the EM algorithm. We have illustrated the utility of the method for purposes of recovering objects of interest in the MSRA Salient Object Database and compared our results to a number of alternatives. We have also provided neighbourhood-based descriptor extensions to the features presented in [2] and [9]. Note that the proposed method is quite general and can be applied to many other types of features which, in contrast with those used here, may not be local in nature.

References

1. Fecteau, J., Munoz, D.: Salience, relevance, and firing: a priority map for target selection. Trends in Cognitive Sciences **10** (2006) 382–290
2. Liu, T., Sun, J., Zheng, N.N., Tang, X., Shum, H.Y.: Learning to detect a salient object. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1–8
3. Mahamud, S., Williams, L., Thornber, K., Xu;, K.: Segmentation of multiple salient closed contours from real images. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 433 – 444
4. Li, H., Ngan, K.N.: Saliency model-based face segmentation and tracking in head-and-shoulder video sequences. Journal of Visual Communication and Image Representation **19** (2008) 320C333
5. Papageorgiou, C., Poggio, T.: A trainable system for object detection. International Journal of Computer Vision **38** (2004) 15C33
6. Marchesotti, L., Cifarelli, C., Csurka, G.: A framework for visual saliency detection with applications to image thumbnailing. In: Proceedings of the IEEE International Conference on Computer Vision. (2009)
7. Kadir, T., Brady, M.: Saliency, scale and image description. International Journal of Computer Vision **45** (2001) 83–105
8. Koch, C., Ullman, S.: Shifts in selective visual attention: Towards the underlying neural circuitry. Human Neurobiology **4** (1985) 219–227
9. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1254–1259

10. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: Proceedings of Neural Information Processing Systems. (2007) 545–552
11. Rosin, P.L.: A simple method for detecting salient regions. Pattern Recognition **42** (2009) 2363–2371
12. Alter, T., Basri, R.: Extracting salient curves from images: An analysis of the saliency network. International Journal of Computer Vision **27** (1998) 51–69
13. Shaashua, A., Ullman, S.: Structural saliency: The detection of globally salient structures using locally connected network. In: Proceedings of International Conference on Computer Vision. (1988) 321–327
14. Berengolts, A., Lindenbaum, M.: On the distribution of saliency. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 1973 – 1990
15. Dickinson, S.J., Christensen, H.I., Tsotsos, J.K., Olofsson, G.: Active object recognition integrating attention and viewpoint control. Computer Vision and Image Understanding **67** (1997) 239–260
16. Navalpakkam, V., Itti, L.: Modeling the influence of task on attention. Vision Research **45** (2005) 205–231
17. Navalpakkam, V., Itti, L.: Search goal tunes visual features optimally. Neuron **53** (2007) 605–617
18. Vincent, B., Troscianko, T., Gilchrist, I.: Investigating a space-variant weighted salience account of visual selection. Vision Research **47** (2007) 1809–1820
19. Lindeberg, T.: Scale-space behaviour of local extrema and blobs. Journal of Mathematical Imaging and Vision **1** (1992) 65–99
20. Rissanen, J.: Stochastic Complexity in Statistical Inquiry Theory. World Scientific Publishing Co., Inc., River Edge, NJ, USA (1989)
21. Dempster, A.P., Laird, M.N., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **39** (1977) 1–22
22. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation **3** (1991) 79–87
23. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research **9** (2008) 1871–1874
24. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Advances in Large Margin Classifiers. (2000) 61–74
25. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two dimensional visual cortical filters. Journal of the Optical Society of America **2** (1985) 1160C1169
26. Ratsch, G., Onoda, T., Muller, K.R.: Soft margins for adaboost. Machine Learning **42** (2001) 287–320
27. Duda, R.O., Hart, P.E.: Pattern Classification. Wiley (2000)
28. Otsu, N.: A thresholding selection method from gray-level histobrams. IEEE Transactions on Systems, Man, and Cybernetics **9** (1979) 62–66
29. van Rijsbergen, C.J.: Information Retireval. Butterworths (1979)
30. Martin, D.R., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 530–549
31. Freixenet, J., Munoz, X., Raba, D., Martí, J., Cufí, X.: Yet another survey on image segmentation: Region and boundary information integration. In: Proceedings of the 7th European Conference on Computer Vision. (2002) 408–422