Feature Selection in Supervised Saliency Prediction

Ming Liang, *Student Member, IEEE*, and Xiaolin Hu, *Senior Member, IEEE*

***Abstract*—There is an increasing interest in learning mappings from features to saliency maps based on human fixation data on natural images. These models have achieved better results than most bottom-up (unsupervised) saliency models. However, they usually use a large set of features trying to account for all possible saliency-related factors, which increases time cost and leaves the truly effective features unknown. Through supervised feature selection, we show that the features used in existing models are highly redundant. On each of three benchmark datasets considered in this paper, a small number of features are found to be good enough for predicting human eye fixations in free viewing experiments. The resulting model achieves comparable results to that with all features and outperforms the state-of-the-art models on these datasets. In addition, both the features selected and the model trained on any dataset exhibit good performance on the other two datasets, indicating robustness of the selected features and models across different datasets. Finally, after training on a dataset for two different tasks, eye fixation prediction and salient object detection, the selected features show robustness across the two tasks. Taken together, these findings suggest that a small set of features could account for visual saliency.**

***Index Terms*—Eye fixation prediction, feature selection, saliency map, salient object detection.**

I. Introduction

VISUAL attention enables human to fast select relevant information from enormous inputs and facilitate the processing of subsequent complex visual tasks. The mechanism of visual attention has been under extensive investigation in neuroscience and psychology. In computer vision, many biologically inspired and mathematically motivated saliency models have been proposed to mimic this function, which have played significant roles in a variety of applications including region of interest detection [1], robotics [2], image cropping [3], object recognition [4], and yarn surface evaluation [5].

Typical bottom-up saliency models [6] extract low-level visual features from the image, and then activate saliency according to some measures. Because of the direct link between visual attention and eye movements, human eye fixation data in free viewing experiments are often used to evaluate saliency in images. The process of free viewing is influenced by many bottom-up stimulus-driven and top-down knowledge-based factors. To account for this complex process, some recent models [7], [8] adopted machine learning techniques to learn the mapping from visual features to fixation data. These models obtained higher saliency prediction accuracies than bottom-up approaches, but they used many features all together. For example, the model of Judd *et al.* [7] used 33 features covering different processing levels, from low-level feature contrast to high-level object detectors. Borji [8] incorporated two additional saliency maps with different activation mechanisms as features to improve the prediction accuracy. On one hand, more features undoubtedly lead to more time cost in prediction, while saliency computation by nature should be fast. On the other hand, a large feature set makes it hard to understand the computational principles underlying saliency detection, as the roles of effective features may be blurred by many less useful features. It is unclear if there exists a smaller set of features which may lead to competitive prediction results; and if there is, which features should be selected.

We explored this issue by using feature selection methods with supervised learning. The goal was to identify a small set of features, with which the saliency prediction should be both effective and efficient. To achieve this goal, we constructed a large set of candidate features and applied feature selection methods over the candidate set. We experimented on three benchmark datasets [7], [9], [10]. The resulting model, though with much fewer features, achieved comparable results to that with all features and beat existing supervised models. Interestingly, we found that swapping the selected features or trained models on different datasets led to negligible degradation of accuracy. Finally, we tested consistency of features selected on a dataset [10] for two different tasks: free viewing fixation prediction and salient object detection. It was found that swapping the two sets of selected feature did not lead to much performance degradation in either task.

A. Related Work

Saliency models can be classified into two categories, with the criterion of the presence of supervision or not.

Bottom-up saliency models do not involve supervised information. Based on the feature integration theory [11] Koch and Ullman [12] first postulated the concept of saliency map, which was then implemented as a computational model by Itti *et al.* [6]. In the model, the first step is to extract multiscale low-level features over several channels. Then a center surround operation is applied to highlight locations distinct

Manuscript received August 28, 2013; revised April 30, 2014; accepted July 8, 2014. Date of publication August 5, 2014; date of current version April 13, 2015. This work was supported in part by the National Basic Research Program (973 Program) of China under Grant 2013CB329403 and Grant 2012CB316301, in part by the National Natural Science Foundation of China under Grant 61273023 and Grant 91120011, in part by the Beijing Natural Science Foundation under Grant 4132046, and in part by the Tsinghua University Initiative Scientific Research Program under Grant 20121088071. This paper was recommended by Associate Editor H. Zhang.

M. Liang is with the School of Medicine, Tsinghua University, Beijing 100084, China (e-mail: liangm07@mails.tsinghua.edu.cn).

X. Hu is with the State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology and also with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: xlhu@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2338893

from the surroundings. Finally, activated feature maps are normalized and combined together into a master saliency map. Saliency of a location is measured by its center surround contrast. Since then, many other saliency measures have been proposed, such as self-information in [9], equilibrium distribution of graph-based random walk in [13], decorrelation and distinctiveness of neural responses in [14], etc. While these models compute the saliency measures with spatial features, some other models operate in the frequency domain [15], [16]. To combine saliency in different feature channels, weighted sum is often adopted with manually assigned weights. Other combining method is also possible. For example, in a recent study, a Markov chain was used to integrate saliency maps derived at different scales of images [17]. Section II-B presents 13 state-of-the-art bottom-up saliency models.

Saliency models in the other category operate in a supervised manner by learning a mapping from visual features to the eye movement data. The features are chosen to account for as many saliency-related factors as possible, covering different processing levels. For example, low-level features include contrast [18] and orientation [19] and high-level features include specific object detectors such as face and text detectors [7], [20], [21]. A somehow different feature refers to the center prior, which is designed to account for the tendency of people watching image center [7], [8], [22]. See Section II-B for details of these features. In addition, the saliency maps generated by bottom-up saliency models, including those presented in Section II-B, can be also used as input features to supervised learning models.

To effectively combine different features, both regression and classification models can be used. Kienzle *et al.* [23] learned a saliency model directly from image pixels (which can be regarded as the simplest feature) using support vector machine (SVM). They found that the stimuli activating the maximum saliency had a center-surround structure, resembling the receptive fields of neurons in early visual stages of mammals. Zhao and Koch [22] learned a model with low-level features and face channels using a linear regression method. They showed that the optimal feature weights varied across different datasets. Judd *et al.* [7] proposed to learn saliency from a large set of 33 image features. In their model, each pixel location was represented by a 33 dimensional feature vector. Training data included positive and negative labeled pixels extracted from smoothed human fixation maps. The linear SVM was used to train the feature weights. The learned pixel-wise saliency was simply the weighted sum of features. By incorporating the saliency maps produced by other two bottom-up models [13], [14] as extra features, Borji [8] compared different learning methods including linear regression, SVM and AdaBoost. The best performance was achieved by AdaBoost. The new features enhanced the prediction power of the learned model, suggesting that the performance may be further boosted by adding more useful features. So far Judd *et al.*'s model and Borji's model have achieved state-of-the-art accuracies on several benchmark datasets.

Another line of work on saliency-related supervised models focuses on salient object detection. Different from eye fixation prediction, salient object detection uses human annotated bounding boxes or binary maps as labels. Conditional random field (CRF) is a popular tool to learn the contributions of local features and the interaction between neighboring features. Liu *et al.* [24] used CRF to combine a set of local, regional and global features. Yang and Yang [25] employed patch-based sparse features and CRF to train a top-down saliency model. In their model, the dictionary of sparse words and CRF were jointly learned with a max-margin approach. Mai *et al.* [26] used CRF to aggregate a set of complementary saliency maps. Because the performance of component maps varied over different images, they learned an aggregation CRF model for each image based on its neighboring training images.

It is reasonable to assume that the places where eyes are attracted usually contain salient objects and therefore salient object detection should strongly correlates with fixation prediction. Saliency models devised for fixation prediction, however, performed poorly on salient object detection datasets [27], suggesting that this claim is doubtful. A recent paper [28] argues that this discrepancy might be caused by the design bias of the salient object detection datasets.

Feature selection [29] is often required for processing noisy high-dimensional data. It can remove redundant features, speed up model learning and reduce the chance of over-fitting. In addition, feature selection enables people to gain better understanding of the model in specific domains, by identifying a few truly effective features. Based on the relationship between feature evaluation and learning methods, supervised feature selection methods [30] can be classified into three categories: filter, wrapper and embedded.

Filter methods evaluate the feature utility by calculating some general measures between features and labels, such as correlation [31], mutual information [32] and Fisher score [33]. These methods behave as a preprocess step for supervised learning. They are fast to compute and not affected by the bias of learning method. Wrapper methods [34] use the prediction accuracy of a predetermined learning model to evaluate the utility of features, and the subset of features leading to the highest accuracy is selected. Wrapper methods usually performed better than filter methods. However, they are computationally expensive and not suitable for a great number of candidate features. Embedded methods perform feature selection and learning simultaneously. The selection is implemented by applying sparsity constraints to the learning machines, such as L_1-norm SVM [35] (denoted by L_1-SVM throughout the paper) and sparse logistic regression [36]. During learning, the sparsity regularization term drives the weights of non-relevant features toward zero. In many cases, embedded methods can achieve comparable performance with wrapper methods (it is not the case in this paper; see Section III-B), but are more efficient.

II. Methods

A. Learning Procedure

The learning procedure is similar to those in [7] and [8]. See Fig. 1 for illustration. Images are divided into training groups and testing groups. The ground truth saliency

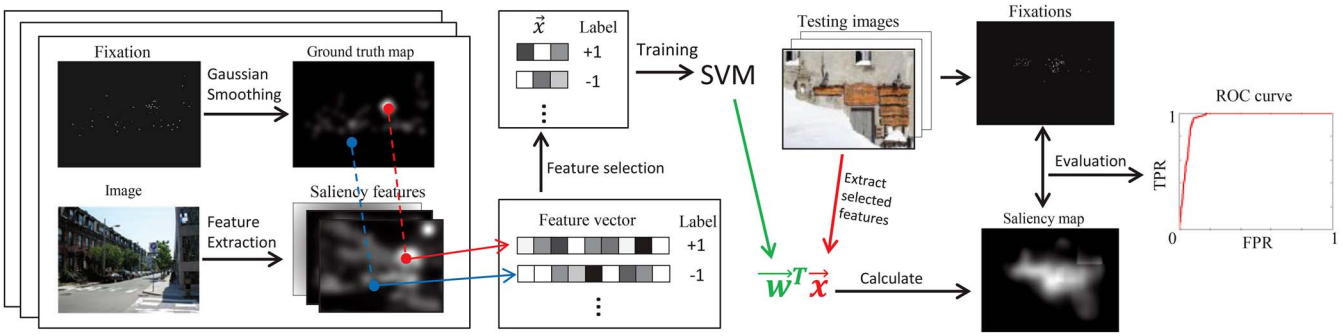


Fig. 1. Overall procedure for training and testing. Best viewed in color.

maps are obtained by convolving a Gaussian filter over the fixation locations of all viewers. For each training image, n_p positive pixel locations are randomly picked from the top 20% salient regions, and n_n negative pixel locations are picked from the bottom 70% salient regions. At each extracted location, many saliency features are extracted and form a vector $\mathbf{x}_i$. The training vectors $\mathbf{x}_i$'s and their corresponding saliency labels (1 denotes positive and -1 denotes negative) are then used to train a classifier parameterized by G. Meanwhile feature selection is carried out to determine which features are useful. Then the saliency label at each location of a testing image with feature vector $\mathbf{x}$ is predicted. In practice, to obtain a continuous saliency map a continuous prediction function $f(\mathbf{x}|G)$ instead of a binary function is used.

B. Existing Candidate Features

To provide a rich pool of candidates, we collect 48 existing saliency features, among which 33 have been used in a previous study [7], which can be classified into four categories as follows.

1) *28 Low-Level Features:* Eleven of them are based on color, including three RGB color channels (R, G, B), three single channel color probability features (RProb, GProb, BProb) and five 3-D color probability features (3DProb1-3DProb5). Fourteen of them are based on subband pyramid, including thirteen steerable pyramid subbands (Subband1-Subband13) in different scales and orientations and a map of Torralba saliency model (Torralba). Three of them are based on feature contrast, including color, intensity and orientation contrast map computed by Itti model (IttiC, IttiI, IttiO).
2) *A Middle-Level Feature (Horizon):* A horizontal detector trained from gist features.
3) *Three High-Level Features:* Face, people, and car detectors, denoted by Face, People, and Car, respectively.
4) *A Center Prior (Center):* A Gaussian-like function used to account for the center bias.

The details of these features can be found in [7]. The other 15 features are generated by 13 bottom-up saliency models, which are briefly described as follows.

1) *Graph-Based Visual Saliency [13] (GBVS):* In this model, a fully connected directed graph is first built on the feature maps. Then a Markov chain is defined on the graph. Saliency is computed as the equilibrium distribution of the Markov chain.
2) *Adaptive Whitening Saliency [14] (AWS):* A whitening process is applied to the multiscale feature maps to remove correlations and highlight the distinctive features. Saliency is then measured as the Hotelling's T-squared statistics.
3) *Attention Based on Information Maximization [9] (AIM):* A set of ICA bases are first trained on a natural image dataset, then the probability distributions of bases coefficients across the entire image are estimated. In each location, saliency is computed as the Shannon's self-information of the coefficients.
4) *Context Aware Saliency [37] (CAS):* Raw image patches are extracted and vectorized in the CIE Lab color space. Saliency is activated as the patch distinctiveness defined in both local and global contexts and based on some visual organization rules. High-level face detector is also incorporated.
5) *Image Manipulation Saliency [38] (IMS):* The patch distinctiveness in the context aware saliency [37] is combined with an object probability map. A coarse saliency map and a fine saliency map are generated with different parameters, which are denoted as IMS-C and IMS-F, respectively.
6) *Dynamic Visual Attention Based on Incremental Coding Length [39] (ICL):* Saliency is activated based on the rarity of sparse features. The rarity is measured by coding length increments.
7) *Spectral Residual Saliency [15] (SRS):* The image is transformed to the frequency domain, and the spectral residual is extracted, which corresponds to the unexpected information in the image. The spectral residual is then transformed back to the spatial domain and serves as the saliency map.
8) *Image Signature [16] (IS):* The image is first transformed to frequency domain using discrete cosine transform (DCT). Then the amplitude information is discarded and only the sign of DCT components are preserved. The preserved information is transformed back to the spatial domain and forms the saliency map.
9) *Random Center Surround Saliency [40] (RCSS):* Center Surround saliency is computed over rectangular regions with random sizes and locations. Then they are fused into the final saliency map.

10) *Frequency Tuned Saliency [1] (FTS):* The contrast of Gaussian blurred image to the mean pixel value is used to measure saliency.
11) *Saliency Detection by Self-Resemblance [41] (SDSR):* Self-resemblance is proposed to activate saliency on local steering kernel features, measuring the distinctness of a pixel from its surroundings. Both local and global self-resemblance can be defined, which leads to two saliency maps SDSR-L and SDSR-G.
12) *Saliency Using Natural statistics [42] (SUN):* A Bayesian framework of attention is proposed, combining both bottom-up saliency and top-down information. Saliency is computed as the self-information of image features, whose statistics are obtained from natural image datasets rather than the processed image.
13) *Segmentation Saliency [43] (SS):* Saliency is measured by the distribution contrast of CIELab color value between the inner and outer parts of a sliding window on the image.

C. New Candidate Features

Most existing saliency features listed above are based on simple image features to activate saliency such as Gabor filtered orientations. These features have relatively low description power. In contrast, more complex descriptors extensively used in object recognition, such as HOG [44] and SIFT [45], are rarely used as features in saliency models. These descriptors provide invariant representations of image patches, i.e., they are robust to geometric and photometric transformations. Given the distinction between these descriptors and simple image features, saliency activated by these descriptors may provide additional information for supervised saliency prediction. We explore this possibility by computing a new saliency feature based on HOG.

HOG extraction involves the following steps. The image is first divided into non-overlapping squared grids (or cells), and a histogram of pixel-wise oriented gradients is computed within each cell. The histogram is then normalized over its neighborhoods. Finally, a descriptor is obtained for each cell. A multiscale pyramid representation is usually used to provide both coarse and fine descriptions. This is achieved by iteratively subsampling the image by half on each side and dividing the resulting images into the same number of cells such that the cells on different levels cover the same sized region on the original image.

HOG mainly describes texture information. But we postulate that edge information is also important for generating saliency. To enhance the edge information another feature is introduced based on HOG, called Canny-HOG (CHOG). It differs from HOG only in the input: the input is the edge map obtained by applying the Canny edge detector [46] to the original image. Fig. 2 shows the input of CHOG. Multiscale HOG descriptors are extracted from the edge map [47].

Essentially HOG gains its invariant representation by histograms in different scale of cells. Inspired by this idea, three additional saliency features are devised for color representation. The original image is transformed into HSV color space,

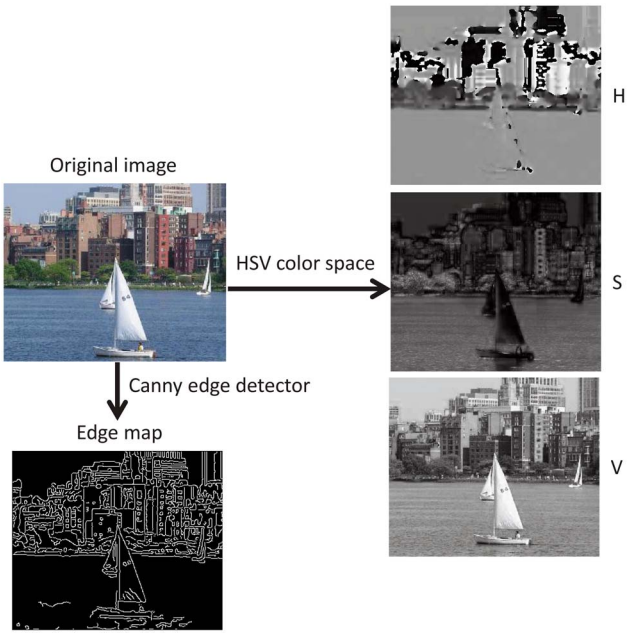


Fig. 2. Input to the CHOG, HH, SH, and VH descriptors. Best viewed in color.

as shown in Fig. 2. Within each cell a color-histogram is constructed. Then three simple color histogram descriptors (HH, SH and VH) are extracted from hue, saturation and value channels, respectively.

After extracting these descriptors, the activation algorithm in AWS [14] is used to compute saliency. The multiscale descriptors corresponding to the same location i are concatenated to form a vector $\mathbf{h}_i$, and the image is described by a matrix $H = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N)$, where N is the number of locations. At each location the saliency feature is computed by the Mahalanobis distance between $\mathbf{h}_i$ and the global mean

$$s_i = \left(\mathbf{h}_i - \bar{\mathbf{h}}\right)^{\mathbf{T}} W^{-1} \left(\mathbf{h}_i - \bar{\mathbf{h}}\right) \tag{1}$$

where $\bar{\mathbf{h}}$ denotes the mean of $\mathbf{h}_1$, $\mathbf{h}_2$, $\ldots$, $\mathbf{h}_N$, and W is the covariance matrix of H. This measure is computed for all of the five descriptors, and the resulting saliency features are denoted by HOGS, CHOGS, HHS, SHS, and VHS, respectively. Fig. 3 illustrates the procedure for computing HOGS. The other four features are computed in similar ways.

Fig. 4 shows some sample images on which the proposed features performed better than existing features. These samples suggest that the proposed features can complement the existing ones for supervised saliency prediction when existing ones fail to provide a good prediction. For example, in the first two images, texts are salient, and HOGS performed better than other features. This might be due to its strong description ability for textures.

D. Feature Selection Methods

1) L_1-SVM: A popular feature selection method refers to L_1-SVM [35], which solves the optimization problem (unbiased SVMs are used in this paper)

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 + c \sum_i \max\left(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\right)^2 \tag{2}$$

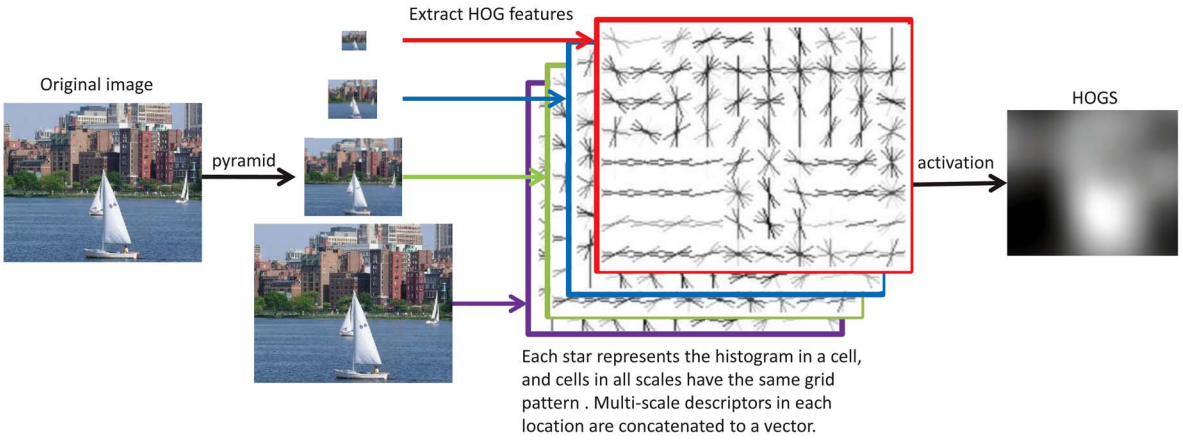


Fig. 3. Procedure for computing HOGS. Best viewed in color.

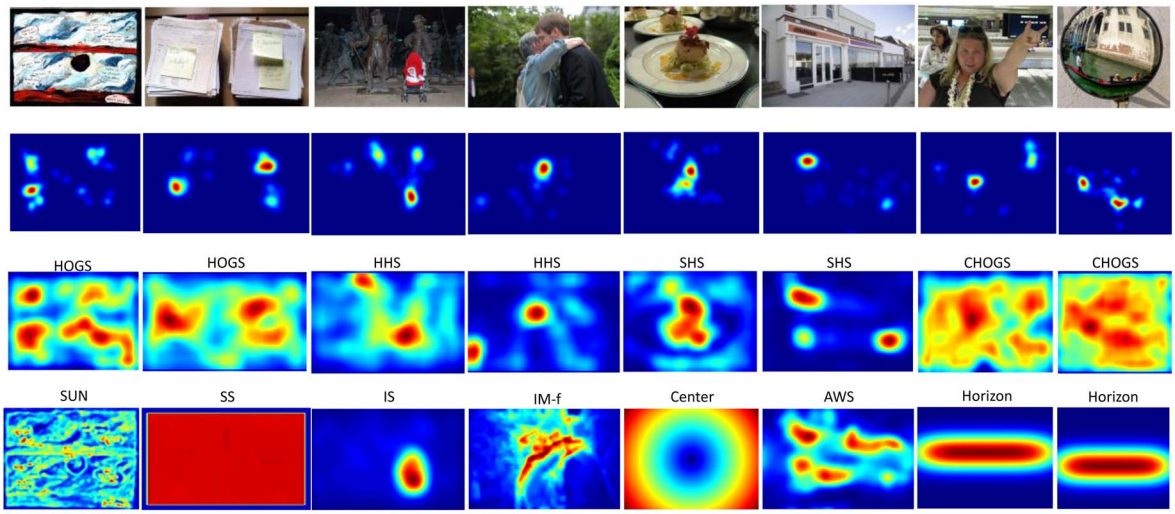


Fig. 4. Sample results of the proposed features and existing features. First row: sample images. Second row: the fixation density maps. Third row: the saliency maps obtained by the proposed features. Fourth row: the saliency maps obtained by the best one among the 48 existing features. Best viewed in color.

where $\mathbf{x}_i$ denotes the feature vector of sample i, y_i denotes the classification label (1 or -1), $\mathbf{w}$ denotes the weight vector and c is a cost parameter. In prediction, the saliency is computed as the weighted sum of features, that is

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}. \tag{3}$$

Note that in standard classification, the sign of $f(\mathbf{x})$ is used as the predicted label. To obtain a continuous saliency map, here we do not binarize the results.

The L_1-norm term in (2) forces the weights of unimportant features to be zero. c controls the tradeoff between the L_1 regularization term and the hinge loss term. When c is smaller the L_1 regularization term is weighted more, and the resulting weight vector will become sparser. By adjusting the value of c, different number of features will be selected which correspond to those with non-zero weights.

2) AdaBoost: AdaBoost iteratively trains a set of weak classifiers on the training data and combines them into a strong classifier. In each iteration a weak classifier is trained to minimize the weighted classification error based on the performance of previous classifiers. In this paper, the weak classifier is a threshold on a selected feature. Then from iteration to iteration, discriminative features will be selected. Denote the output of each weak classifier by $h_i(\mathbf{x})$. Then a continuous saliency value is calculated as

$$f(\mathbf{x}) = \sum_i \alpha_i h_i(\mathbf{x}) \tag{4}$$

where α_i stands for the weight of the i-th weak classifier. Again we do not binarize the prediction results.

3) Greedy Feature Selection (GFS): Unlike L_1-SVM or AdaBoost, wrapper methods [48] assess feature subsets according to their usefulness to a given predictor. To avoid exhaustive search for optimal feature combination, greedy selection is used. In each iteration, candidate features are ranked by a scoring function S and the one maximizing S is selected

$$\nu_k = \underset{i\notin\{\nu_1,\ldots,\nu_{k-1}\}}{\operatorname{argmax}} \; S\left(X_i, Y|X_{\nu_1}, X_{\nu_2}, \ldots, X_{\nu_{k-1}}\right) \tag{5}$$

where k denotes the iteration number, X_i denotes the i-th feature variable, Y denotes the label variable, and v_k denotes the index of the feature selected in the k-th iteration.

In the paper, S is defined as the cross-validation prediction accuracy obtained by the standard linear L_2-norm SVM (replace $\|\mathbf{w}\|_1$ in (2) with $\|\mathbf{w}\|_2^2$; denoted by L_2-SVM throughout the paper) on the training set. Again, the saliency at each testing location is calculated according to the continuous function (3). The algorithm stops when some condition is reached.

III. Experiment Results

A. Experiment Setup

Three public human eye fixation datasets were used in our experiments. The first is the MIT dataset [7]. It is a large human fixation dataset consisting of 1003 indoor and outdoor images. The longest image dimension is 1024 pixels and the other dimension ranges from 405 to 1024 pixels. Each image was viewed by 15 subjects for 3 s and the eye movement data were recorded. The second is the Toronto dataset [9]. It consists of 120 indoor and outdoor images with 681×511 pixels. Each image was viewed by 20 subjects for 4 s. The third is the ImgSal dataset [10], which consists of 235 natural images with 640×480 pixels. Different from the other two datasets, it has two sets of labels for free viewing fixations and human annotated salient objects, respectively.

Four metrics were used to quantitatively evaluate the performance of models, including area under the (ROC) curve (AUC) [49], normalized scanpath saliency (NSS) [50], histogram intersection (HI, in [51] this is called similarity score), and linear correlation coefficient (CC) [52]. AUC is the most widely used metric for saliency prediction. Given a threshold *th*, pixels with saliency values higher than *th* are predicted positive and others are predicted negative. Based on the classification results and the ground truth, a true positive rate TPR and a false positive rate FPR are calculated. By varying *th* a set of TPR and FPR are obtained, resulting in an ROC curve, and the area under this curve measures how well a saliency map predicts the human fixation locations on an image. An AUC of 1 corresponds to perfect prediction and an AUC of 0.5 corresponds to random guess (chance level). NSS is the average saliency value at human fixation locations, and the saliency map has been prenormalized to have zero mean and unit variance. Higher NSS means better performance, and NSS $= 0$ corresponds to random guess. HI is defined as $\sum_i \min(P_i, Q_i)$, where i denotes the spatial location, P and Q denote the spatial distribution histograms of a saliency map and the smoothed fixation map, respectively. HI $= 1$ means that the two histograms are the same and HI $= 0$ means that the two histograms have no overlap. CC is calculated as $\mathrm{cov}(F, S)/(\sqrt{\mathrm{var}(F)\mathrm{var}(M)})$, which measures the linear relationship between the smoothed fixation map F and a saliency map M. CC $= 1$ indicates a perfect positive linear relationship and CC $= 0$ indicates no linear relationship at all.

Note that NSS, HI and CC can be boosted by an appropriate monotonous transformation of saliency values. In Borji's model [8] the saliency maps were passed through an exponential function before calculating NSS and CC. For fair comparison, we applied a similar operation to the saliency maps obtained by all other models presented in the paper including our models and Judd *et al.*'s model [7]: $\tilde{s} = \exp(s)$, where s denotes the saliency value. Note that this operation has no effect on AUC.

All of the 48 existing saliency maps on the three benchmark datasets were obtained by executing their original implementations. The HOGS and CHOGS saliency maps were obtained with the aid of VLFeat library [53], while the HHS, SHS, VHS saliency maps were obtained with our own implementations. The five new features used a four-level pyramid representation. The side lengths of cells on different scale of images (from fine to coarse) were 32, 16, 8, and 4, respectively.

The supervised learning methods L_1-SVM and L_2-SVM were based on the Liblinear [54] implementations, and AdaBoost was based on Gentle AdaBoost [55] implementation. For GFS, The average AUC of 5-fold cross validation on the training set was used as the score S. For the MIT dataset the numbers of positive samples n_p and negative samples n_n extracted from each image were both set to 10, same as in [7]. Since the Toronto and ImgSal datasets had much fewer images, to collect enough samples, the two numbers were set to 100 for the two datasets.

GFS needs L_2-SVM and the latter has a cost parameter c. We found that the performance of GFS was insensitive to this parameter. The results reported in the paper were obtained with $c = 1$.

B. Feature Selection for Free Viewing Fixation Prediction

The feature selection methods were applied on the MIT, Toronto and ImgSal datasets for free viewing fixation prediction. Table I details the prediction accuracies of the L_2-SVM and three feature selection methods (GFS, L_1-SVM, and AdaBoost) portrayed by the metrics AUC, HSS, HI and CC. Different hyper-parameters or stopping criteria of the three feature selection methods were adopted, which led to two versions of each method in the table. The details are as follows.

Except the cost parameter c for the L_2-SVM which is used in GFS, GFS does not have any other hyper-parameters and the only thing it has to choose is the stopping condition. In general we cannot expect better performance with fewer features if the candidate feature set is given, but this actually happened in experiments. In fact, more features yielded worse results on the training set due to over-fitting (Fig. 6). In view of this, we chose the following stopping condition, denoted by GFS-fewest: the average AUC of the 5-fold cross validation on the training set first reached the average AUC obtained with all features. The resulting feature subsets consisted of only 9.2, 10.4, and 10.6 features on the MIT, Toronto and ImgSal datasets on average. However, the performance was nearly the same as that with all 53 features (see Table I). This result suggests that the candidate features are highly redundant.

Fig. 5 shows the feature selection processes in all five training trials. On each dataset the order of features entering the subset were highly consistent, especially in early iterations.

TABLE I

COMPARISON OF MODELS FOR FREE VIEWING FIXATION PREDICTION. VF DENOTES THE NUMBER OF VALID FEATURES. THE COST PARAMETER c FOR L_1-SVM1 WAS SET TO 2^{-11}, 2^{-10}, AND 2^{-11} ON THE MIT, TORONTO, AND IMGSAL DATASETS, RESPECTIVELY. FOR L_1-SVM2 IT WAS SET TO 2^{-8}, 2^{-8}, AND 2^{-9} ON THE THREE DATASETS, RESPECTIVELY. THE ITERATION NUMBERS OF ADABOOST1 AND ADABOOST2 WERE SET TO 15 AND 57, RESPECTIVELY. FIVEFOLD CROSS VALIDATION RESULTS ARE REPORTED FOR THE MODELS IN THE FIRST AND SECOND GROUPS AND JUDD *et al.*'S MODEL [7]. TENFOLD CROSS VALIDATION RESULTS ARE REPORTED FOR BORJI'S MODEL [8]

	MIT					Toronto					ImgSal				
Model	VF	AUC	NSS	HI	CC	VF	AUC	NSS	HI	CC	VF	AUC	NSS	HI	CC
L_2-SVM (all)	53	0.864	1.801	0.391	0.532	53	**0.863**	1.937	0.551	0.689	53	**0.852**	1.899	**0.677**	0.772
GFS-fewest	9.2	0.864	1.805	0.389	0.529	10.4	0.862	1.973	0.550	0.693	10.6	0.851	1.915	0.672	0.768
GFS-max	24.4	**0.866**	**1.816**	0.391	**0.534**	27	0.862	1.957	0.552	0.692	24.2	**0.852**	**1.924**	0.676	0.774
L_1-SVM1	9.6	0.858	1.689	0.368	0.508	12	0.861	1.910	0.524	0.689	12.8	0.850	1.870	0.662	0.769
L_1-SVM2	23.4	0.863	1.776	0.386	0.528	29.2	**0.863**	1.958	0.548	**0.698**	23.4	0.851	1.902	0.674	**0.776**
AdaBoost1	9.2	0.847	1.566	0.411	0.484	10.4	0.848	1.661	0.559	0.627	9.2	0.839	1.584	0.649	0.691
AdaBoost2	25.2	0.856	1.672	**0.422**	0.510	28.2	0.855	1.766	**0.578**	0.650	24.4	0.846	1.640	0.649	0.697
Optimal	9	0.864	1.804	0.389	0.529	8	0.862	**1.977**	0.549	0.692	9	0.851	1.918	0.670	0.765
Naïve	9	0.854	1.697	0.385	0.503	8	0.853	1.924	0.543	0.677	9	0.847	1.842	0.661	0.747
Judd *et al.* [7]	33	0.843	1.572	0.364	0.474	33	0.843	1.686	0.509	0.611	33	0.838	1.672	0.630	0.687
Borji [8]	35	0.854	1.635	0.369	0.501	35	0.846	1.674	0.498	0.634	—	—	—	—	—

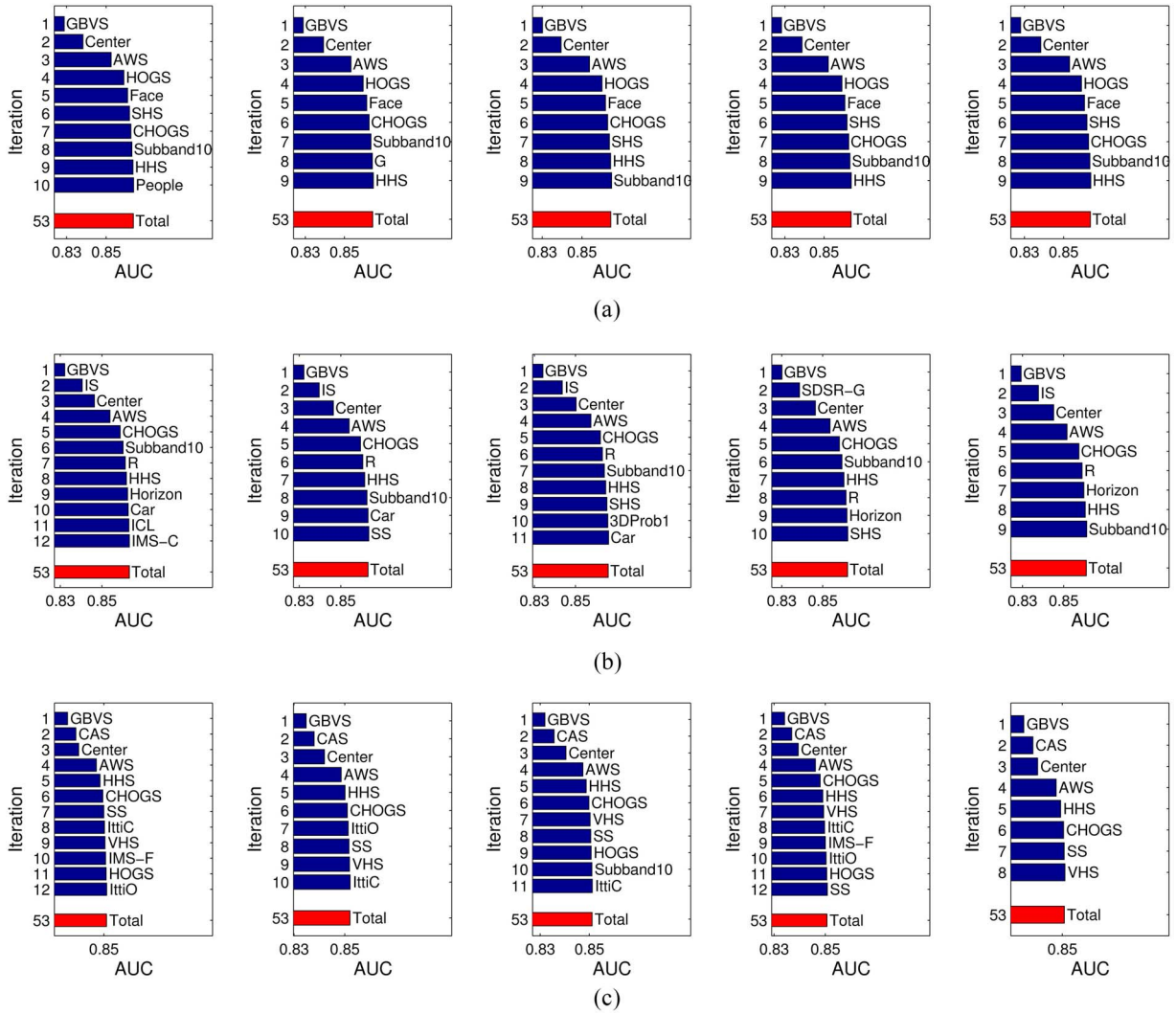


Fig. 5. Feature selection processes of GFS-fewest for free viewing fixation prediction. All of the five trials on the (a) MIT, (b) Toronto, and (c) ImgSal datasets are shown. The y-axis denotes the iteration number and the x-axis denotes the AUC obtained by the selected features so far.

On the MIT dataset, GBVS, Center, AWS, HOGS, and Face were always selected in the first five iterations. On the Toronto dataset, GBVS, IS, Center, AWS, and CHOGS were selected in the first five iterations most of the time, except in the fourth trial where IS was replaced by SDSR-G. On the ImgSal dataset, GBVS, CAS, Center, AWS, HHS, and CHOGS were

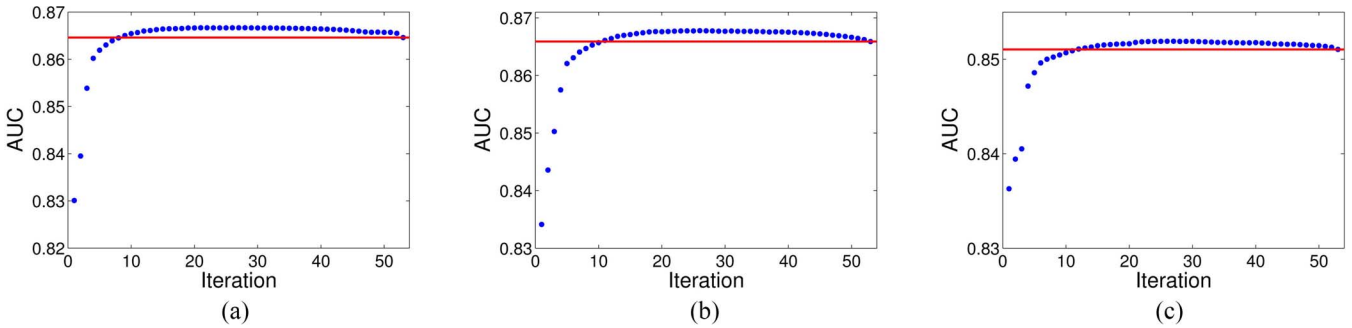


Fig. 6. Average AUC of fivefold cross validation on the training set with respect to the iterations in a sample trial on the (a) MIT, (b) Toronto, and (c) ImgSal datasets. The horizontal lines indicate the average AUC with all features.

always selected in the first six iterations. Other observations are as follows.

First, three features, i.e., GBVS, Center and AWS were always selected in the first few iterations on the three datasets, which implies that they are critical for predicting eye fixations.

Second, some newly designed features in this paper including HHS and CHOGS were often selected in the early iterations of many training trials. This observation supports the effectiveness of the new features.

Third, the features selected on each dataset are significantly different from each other in construction. For example, among the five features selected on the MIT dataset, Center is a high-level summarization about the possible salient locations. Face feature reflects the tendency to fixate at interesting objects. GBVS and AWS are based on similar low-level features. The former uses a distance-dependent activation algorithm and the later uses a distance-independent one. HOGS is based on different image descriptors and adopts the same activation algorithm as AWS.

To explore whether the performance of GFS can be further improved by adding more features, we set the iteration number to 53 and took the maximum average AUC on the training set over all iterations. This condition is denoted by GFS-max in Table I. Note that the maximum AUC may not be achieved on the final iteration due to over-fitting (see Fig. 6). This resulted in 24.4, 27 and 24.2 features on average over the MIT, Toronto and ImgSal datasets, respectively (Table I). Compared with GFS-fewest, GFS-max yielded better performance on testing sets under most metrics, but the improvements were very small. This indicates that the GFS-fewest subsets have included enough useful features.

By adjusting the hyper-parameter c for L_1-SVM and the total number of iterations for AdaBoost, we can obtain different number of valid features and different prediction accuracies. In experiments, we found that with appropriate hyper-parameters the two methods could also achieve good results, though not as good as GFS in general. As we were mostly concerned with the number of valid features, we tuned these hyper-parameters to achieve roughly the same number of features as GFS-fewest and GFS-max, respectively, which has led to *L_1-SVM1, L_1-SVM2, AdaBoost1*, and *AdaBoost2*. See Table I. L_1-SVM did not perform as well as GFS, especially for the version with fewer features. AdaBoost achieved the highest HI scores on the MIT and Toronto datasets, but performed worse than GFS and L_1-SVM under the other three metrics.

For comparison, the results of two state-of-the-art supervised saliency models [7], [8] are also presented in Table I. The results of Judd *et al.*'s model [7], which adopted 33 features, were obtained by executing the codes downloaded from an author's website. The results of Borji's model [8], which adopted 35 features, were obtained by using the saliency maps downloaded from the author's website. Only the saliency maps of the MIT and Toronto datasets were available. The saliency maps of the MIT dataset were obtained by 10-fold cross validation. The saliency maps of the Toronto dataset were obtained by a model trained over the MIT dataset. With about ten features on average, GFS and L_1-SVM beat the two models under nearly all metrics. With about 26 features on average, AdaBoost also beat the two models.

It has been shown that among the three feature selection methods, GFS performed the best. However, it is computationally expensive. On a mainstream PC one GFS training trial took about 14 hours on the MIT dataset while L_1-SVM only took a few seconds and AdaBoost took tens of seconds. If feature selection is needed in some online applications, one should consider L_1-SVM instead. But in this paper, feature selection was conducted offline and we were more interested in the properties of selected features. Therefore, in what follows, we base our analysis on GFS.

C. Optimal Feature Sets

Though the GFS-fewest method obtained excellent results with 9.2, 10.4, and 10.6 features on average over the three datasets, respectively, the features selected in different trials were not identical, as shown in Fig. 5. Two interesting questions arise. First, is there a fixed optimal feature set that can achieve similar accuracy to GFS-fewest on each dataset? Such a set would make it clearer which features are indeed useful for saliency prediction on the particular dataset. Second, are these optimal feature sets robust across different datasets? If yes, then we can directly apply these features on new datasets without performing feature selection again, which is time-consuming.

We investigated the first question by constructing the optimal feature set for each dataset as follows: select features that were selected at least four times in the total of five GFS-fewest

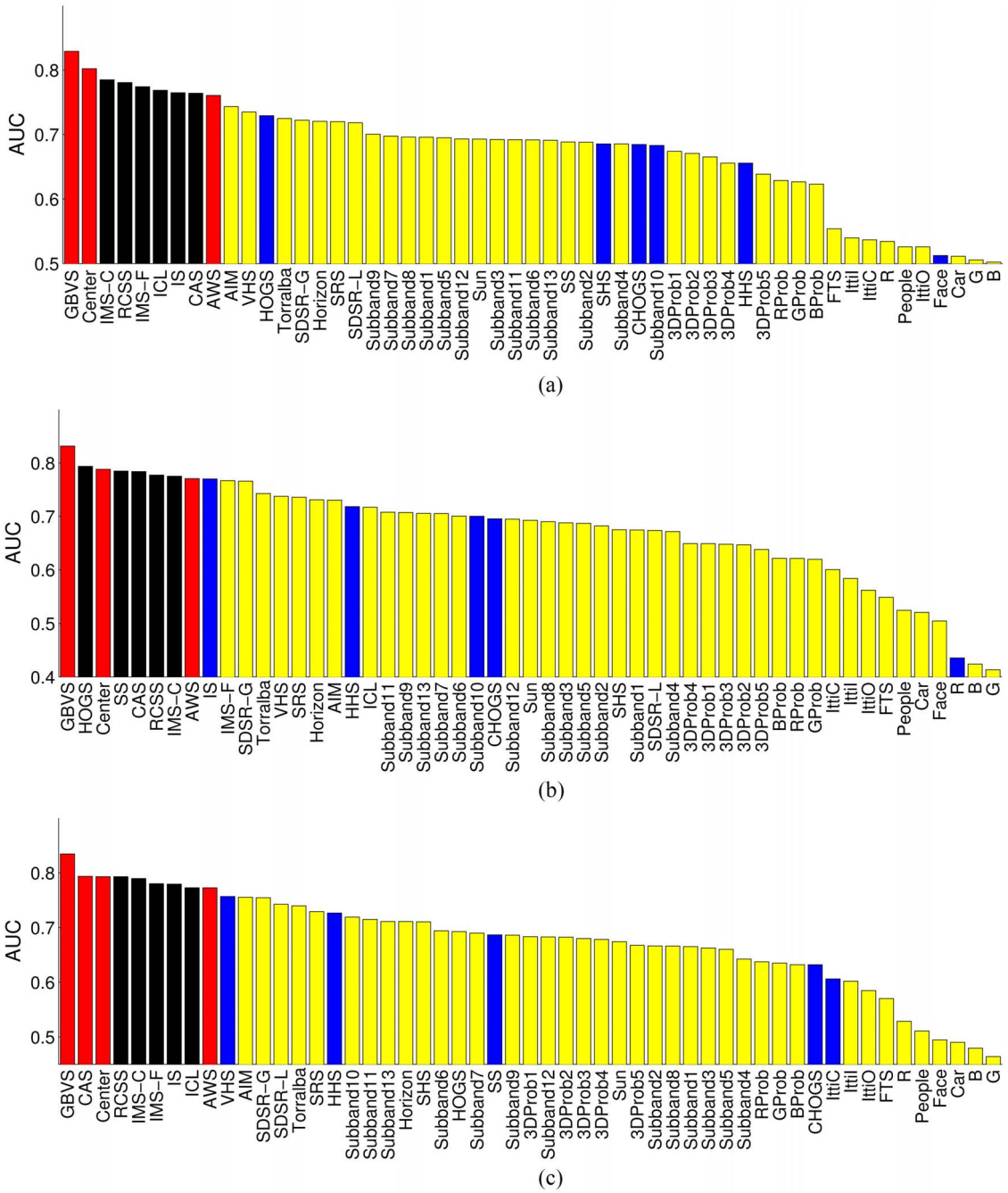


Fig. 7. AUC achieved by individual features without supervised information. Black, blue and red bars correspond to features belonging to naïve set, optimal set, and both, respectively. Yellow bars correspond to other features. Best viewed in color. (a) MIT. (b) Toronto. (c) ImgSal.

trials. This resulted in a feature set with size 9, 8, and 9 for the MIT, Toronto and ImgSal datasets, respectively.

1) *MIT*: GBVS, Center, AWS, HOGS, Face, SHS, HHS, CHOGS, and Subband10.
2) *Toronto*: GBVS, IS, Center, AWS, HHS, CHOGS, Subband10, and R.
3) *ImgSal*: GBVS, CAS, Center, AWS, SS, HHS, VHS, CHOGS, and IttiC.

These features led to nearly identical scores with GFS-fewest and outperformed the two state-of-the-art models (see Table I, Optimal).

A naïve control approach would be selecting the same number of features with the best individual performances on each dataset. For this purpose, we calculated the AUC for each of the 53 features individually (Fig. 7). The following features were selected to construct the control sets.

1) *MIT*: GBVS, Center, IMS-C, RCSS, IMS-F, ICL, IS, CAS, and AWS.
2) *Toronto*: GBVS, HOGS, SS, Center, CAS, RCSS, IMS-C, and AWS.
3) *ImgSal*: GBVS, CAS, Center, RCSS, IMS-C, IMS-F, IS, ICL, and AWS.

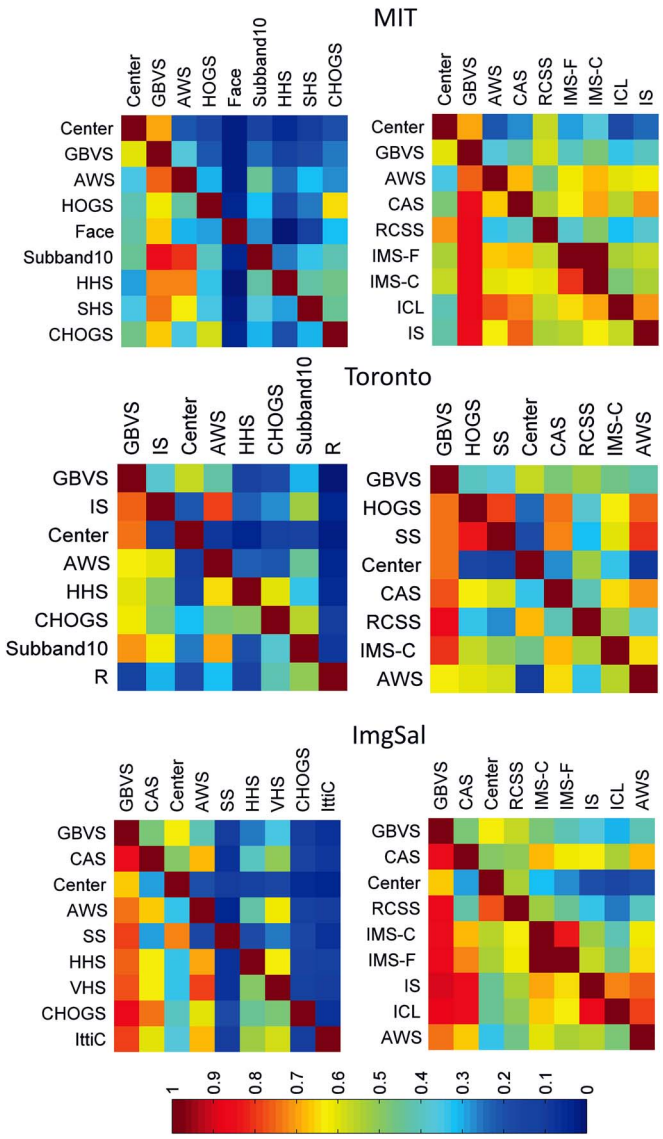


Fig. 8. Redundancy matrices for the optimal features (left) and naïve features (right) on the MIT (top), Toronto (middle), and ImgSal (bottom) datasets. Best viewed in color.

Under all four evaluation metrics these control sets were outperformed by the optimal sets (see Table I, Naïve).

For each dataset, the optimal feature set and the control set had three (MIT and Toronto) or four (ImgSal) features in common (red bars in Fig. 7). Other features alone in the optimal sets (blue bars) could not compete with other features in the control sets (black bars). The better performance achieved by the optimal sets suggests more redundancy among features in the control sets than in the optimal sets. To verify this, we defined a redundancy measure $R(i, j)$ between two features X_i and X_j with respect to the ground-truth saliency labels (see the Appendix). $R(i, j)$ is between 0 and 1, and a higher value indicates more redundancy. Fig. 8 shows the redundancy matrices for the optimal features and the naïve features on the three datasets. It is evident that the optimal features contain less redundant information than the naïve features.

We then investigated the second question raised in the beginning of this subsection. Note that the three datasets have rather different properties. The MIT dataset contains many daily life objects such as faces and texts, while the Toronto dataset rarely contains these objects. The ImgSal dataset contains salient objects of different sizes. In spite of these differences, the three optimal feature sets had five features in common: GBVS, AWS, Center, CHOGS, and HHS.

We then conducted fivefold cross validation on a dataset with the optimal features for another dataset. In this case, the optimal features for different datasets were swapped (Swap1 condition). The average scores of four metrics were calculated. For better illustration, the scores on each dataset were then normalized by dividing the scores obtained with its own optimal features (Table I, Optimal). Most of the scores were very close to 1 (Fig. 9, top), suggesting that the optimal features were robust across different datasets. Next, we trained a model on a dataset with its own optimal features and test on the other dataset. In this case, not only the optimal features but also the models (i.e., feature weights) were swapped (Swap2 condition). Even in this case, the performance did not degrade much (Fig. 9, bottom), suggesting that the trained models were robust across different datasets. In addition, one can verify that the models in the two swap conditions performed better than the two existing models [7], [8] by transforming the normalized scores in Fig. 9 to original scores.

D. Feature Selection Without the Center Prior

The center prior is undoubtedly useful for saliency prediction during free viewing static images because of the strong center bias. However, for other tasks such as visual tracking center bias may not be useful anymore. In this case it will be important to find the indeed useful features other than the spatial bias of specific tasks. We explored this issue by discarding the center prior from the candidate feature set, and then selecting features using GFS. The resulting GFS-fewest feature sets consisted of 11, 10.2 and 8.4 features on average for the MIT, Toronto and ImgSal datasets, respectively (see Table II).

We then constructed the optimal features using the same method as in Section III-C. The resulting optimal features were as follows.

1) *MIT*: GBVS, AWS, HOGS, Face, CHOGS, HHS, Subband10, RCSS, SRS, and FTS.
2) *Toronto*: GBVS, AWS, IS, CHOGS, HHS, and Subband10.
3) *ImgSal*: CAS, GBVS, AWS, HHS, IttiO, RCSS, and HOGS.

Many of these optimal features were shared with the optimal feature set with the center prior (seven for MIT dataset, all for Toronto dataset and four for ImgSal dataset). As expected, when there was no center prior, the features with implicit center bias were favored by GFS such as RCSS, which was not selected when the center prior was present.

Note that the optimal models for the MIT and ImgSal datasets have yielded nearly identical scores to GFS-fewest (Table II). The selected features on the Toronto dataset had larger variations across different trials, resulting in an optimal set with only six features. As a result, its performance was

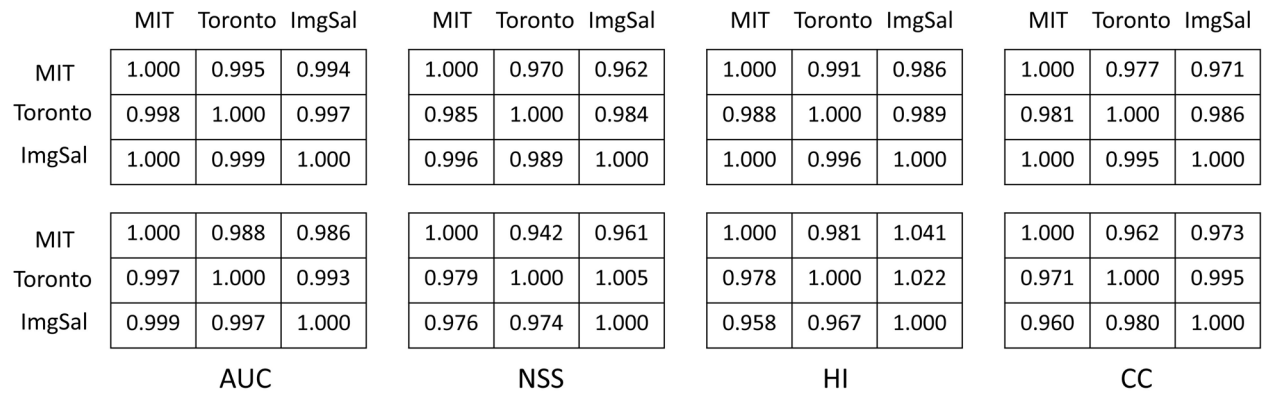


Fig. 9. Normalized scores in two swap conditions. Top (Swap1): in each matrix the element on the *i*th row and *j*th column corresponds to the average score of fivefold cross validation on dataset *i* using optimal features for dataset *j*. Bottom (Swap2): in each matrix the element on the *i*th row and *j*th column corresponds to the testing score on dataset *i* with a model trained on dataset *j* using optimal features for dataset *j*. All scores in the figure have been normalized by dividing the average score of fivefold cross validation on dataset *i* using optimal features for dataset *i*.

TABLE II
COMPARISON OF THE MODELS FOR FREE VIEWING FIXATION PREDICTION WITHOUT THE CENTER PRIOR. PARAMETER SETTINGS ARE THE SAME AS IN TABLE I

	MIT					Toronto					ImgSal				
Model	VF	AUC	NSS	HI	CC	VF	AUC	NSS	HI	CC	VF	AUC	NSS	HI	CC
L_2-SVM (all)	52	0.852	1.671	**0.376**	**0.491**	52	0.852	1.838	0.526	0.646	52	0.842	1.766	0.636	**0.709**
GFS-fewest	11	0.853	1.683	0.370	0.483	10.2	0.849	1.834	0.521	0.631	8.4	0.842	1.749	0.631	0.688
GFS-max	25	**0.854**	**1.694**	0.372	0.488	24.2	**0.853**	**1.872**	**0.527**	**0.649**	19.6	**0.844**	**1.770**	0.630	0.696
Optimal	10	0.853	1.680	0.369	0.482	6	0.847	1.815	0.518	0.618	7	0.841	1.744	**0.637**	0.688
Judd *et al.* [7]	32	0.785	1.206	0.332	0.353	32	0.794	1.304	0.454	0.462	32	0.787	1.392	0.548	0.555
Borji [8]	34	0.840	1.438	0.328	0.437	34	0.837	1.519	0.452	0.573	—	—	—	—	—

TABLE III
COMPARISON OF THE MODELS ON THE IMGSAL DATASET FOR BOTH FREE VIEWING FIXATION PREDICTION AND SALIENT OBJECT DETECTION. PARAMETER SETTINGS ARE THE SAME AS IN TABLE I

	Eye fixation					Salient object				
Model	VF	AUC	NSS	HI	CC	VF	AUC	NSS	HI	CC
L_2-SVM (all)	53	**0.852**	1.899	**0.677**	0.772	53	**0.976**	3.037	0.664	0.698
GFS-fewest	10.6	0.851	1.915	0.672	0.768	14.6	0.973	3.029	0.663	0.696
GFS-max	24.2	**0.852**	**1.924**	0.676	**0.774**	30.2	0.975	3.045	0.665	0.699
Optimal	9	0.851	1.918	0.670	0.765	13	**0.976**	**3.093**	**0.670**	**0.710**
Swap1	13	0.849	1.851	0.669	0.766	9	0.971	3.023	0.650	0.681
Swap2	13	0.839	1.804	0.639	0.713	9	0.955	2.688	0.620	0.626
Judd *et al.* [7]	33	0.838	1.672	0.630	0.687	33	0.955	2.566	0.629	0.596

not as good as the GFS-fewest. A possible reason is that this dataset has too few images for yielding a robust fixed set of features.

E. Free Viewing Fixation Versus Salient Object Detection

There are two types of attention, bottom-up saliency and top-down attention. The former is stimulus-driven and the latter is goal-directed. In the task of free viewing, eye fixations reflect bottom-up saliency, while the annotation in the task of salient objects detection may be more influenced by top-down attention. To find out how the features are weighted by different tasks, we also conducted GFS feature selection over the ImgSal dataset using the human annotated salient object labels. The results were compared with that using eye fixation labels.

The task of detecting salient objects resulted in a larger GFS-fewest set and an optimal set than those for the fixation prediction task (see Table III). This is reasonable because top-down attention involves more complex high-level factors than bottom-up saliency. The optimal features for the free viewing task were GBVS, CAS, Center, HHS, VHS, AWS, SS, CHOGS, and IttiC; for the salient object detection task were GBVS, CAS, Center, HHS, VHS, IMS-F, IMS-C, SDSR-L, SDSR-G, HOGS, R, GProb, and People. The two sets had five features in common.

As before we conducted two swapping experiments across the two tasks. Swap1 achieved very similar scores to the optimal features. Swap2 achieved lower scores, though the gap was not so big. See Table III. These results suggest that the definitions of saliency in the two tasks are consistent, which is in agreement with [28].

IV. CONCLUSION

We investigated feature selection in supervised saliency learning. A rich set of candidate features was first constructed

consisting of many existing ones in the literature and several newly designed ones. After feature selection, we found that a few features could achieve comparable results to all features. This was validated with extensive experiments on three human eye fixation prediction datasets. The model with these selected features beat existing models on these benchmark datasets. In addition, the features selected on one dataset, as well as the model trained on it, were also effective on other datasets, indicating their robustness.

This paper has some other implications which are worth further investigations. First, we have seen that during the process of feature selection, some newly designed histogram-based features were often selected in the first few iterations, which implies that complex image descriptors are useful for predicting eye fixations. Such descriptors have been rarely studied in the context of saliency prediction. Second, experimental results on a dataset for both free viewing fixation prediction and salient object detection suggest consistency of the two tasks in terms of effective features. But this conclusion needs empirical support on other salient object detection datasets, especially those without dataset design bias [28].

APPENDIX

REDUNDANCY BETWEEN TWO FEATURES

We treat the feature X_i and the ground-truth label Y as random variables and define the redundancy between features X_i and X_j as

$$R(i,j) = \frac{I(Y;X_i) + I(Y;X_j) - I(Y;X_i,X_j)}{I(Y;X_i)} \tag{6}$$

where $I(U;V)$ denotes the mutual information between two random variables U and V

$$I(U;V) = H(U) + H(V) - H(U,V)$$

where $H(U) = -\sum_U P(U)\log P(U)$ is the entropy of U with the probability distribution $P(U)$. Intuitively, mutual information measures the information that U and V share. In general, higher R value between two features indicates more redundancy between them. On one extreme, if the information shared by X_i and Y is totally different from that shared by X_j and Y, then the two features X_i and X_j contain no redundant information about Y and $R(i,j) = 0$. On the other extreme, if the information shared by X_i and Y is the same as that shared by X_j and Y (e.g., the two features are identical), then the information about Y contained in X_i and X_j are totally redundant and $R(i,j) = 1$. Note that $R(i,j)$ is normalized by $I(Y;X_i)$, so the redundancy matrix R is nonsymmetric. This similarity measure differs from that in [51], which is used for computing the similarity between pairs of saliency maps obtained by different models, where the ground-truth label information is not considered.

The problem then reduces to determining the probability distributions of the random variables X_i and Y. Since Y has only two possible values ± 1, it is natural to treat it as a binary variable. Its probability distribution is estimated by sampling a large set of pixels on images over the dataset

$$P(Y=1) = \frac{\|\{j{:}y^j = 1\}\|}{N}$$
$$P(Y=-1) = \frac{\|\{j{:}y^j = -1\}\|}{N} \tag{7}$$

where y^j denotes the label of sample j, N denotes the number of samples and $\|\|$ denotes the cardinal of a set. The feature X_i is real-valued, but for simplicity we convert it to binary values with a threshold t_i. The probability distribution is estimated by sampling a large set of samples on saliency maps produced by X_i

$$P(X_i=1) = \frac{\|\{j{:}{x_i}^j > t_i\}\|}{N}$$
$$P(X_i=-1) = \frac{\|\{j{:}{x_i}^j \le t_i\}\|}{N} \tag{8}$$

where ${x_i}^j$ denotes the value of sample j on the saliency map produced by X_i. The threshold t_i is chosen to maximize $I(Y;X_i)$. The joint distributions $P(X_i,Y)$ and $P(X_i,X_j,Y)$ are estimated similarly, for example

$$P(X_i=1,Y=1)\frac{\|\{j{:}{x_i}^j > t_i, y^j = 1\}\|}{N}$$
$$P(X_i=1,X_j=1,Y=1) = \frac{\|\{k{:}x_i^k > t_i, x_j^k > t_j, y^k = 1\}\|}{N}. \tag{9}$$

REFERENCES

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, 2009, pp. 1597–1604.

[2] Y. Yu, G. K. Mann, and R. G. Gosine, "An object-based visual attention model for robotic applications," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 5, pp. 1398–1412, Oct. 2010.

[3] B. Suh, H. Ling, B. Bederson, and D. Jacobs, "Automatic thumbnail cropping and its effectiveness," in *Proc. 16th Annu. ACM Symp. User Interface Softw. Technol.*, 2003, pp. 95–104.

[4] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2004, pp. 37–44.

[5] Z. Liang, B. Xu, Z. Chi, and D. Feng, "Relative saliency model over multiple images with an application to yarn surface evaluation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1249–1258, Aug. 2014.

[6] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[7] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Kyoto, Japan, 2009, pp. 2106–2113.

[8] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2012, pp. 438–445.

[9] N. Bruce and J. Tsotsos, "Saliency based on information maximization," in *Advances in Neural Information Processing Systems*, 2005, pp. 155–162.

[10] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 996–1010, Apr. 2013.

[11] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognit. Psychol.*, vol. 12, no. 1, pp. 97–136, 1980.

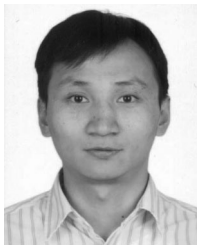
[12] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," *Human Neurobiol.*, vol. 4, no. 4, pp. 219–227, 1985.

[13] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in Neural Information Processing Systems*, 2006, pp. 545–552.

[14] A. Garcia-Diaz, X. Fdez-Vidal, X. Pardo, and R. Dosil, "Decorrelation and distinctiveness provide with human-like saliency," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Systems*, Bordeaux, France, 2009, pp. 343–354.
[15] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Minneapolis, MN, USA, 2007, pp. 1–8.
[16] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 194–201, Jan. 2012.
[17] T. Shi, M. Liang, and X. Hu, "A reverse hierarchy model for predicting eye fixations," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 2822–2829.
[18] P. Reinagel and A. M. Zador, "Natural scene statistics at the centre of gaze," *Network*, vol. 10, no. 4, pp. 341–350, 1999.
[19] D. Parkhurst, K. Law, and E. Niebur, "Modeling the role of salience in the allocation of overt visual attention," *Vis. Res.*, vol. 42, no. 1, pp. 107–123, 2002.
[20] W. Einhäuser, M. Spain, and P. Perona, "Objects predict fixations better than early saliency," *J. Vis.*, vol. 8, no. 14, 2008, Art. ID 18.
[21] M. Cerf, J. Harel, W. Einhäuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Advances in Neural Information Processing Systems*, 2008.
[22] Q. Zhao and C. Koch, "Learning a saliency map using fixated locations in natural scenes," *J. Vis.*, vol. 11, no. 3, 2011, Art. ID 9.
[23] W. Kienzle, M. O. Franz, B. Schölkopf, and F. A. Wichmann, "Center-surround patterns emerge as optimal predictors for human saccade targets," *J. Vis.*, vol. 9, no. 5, pp. 1–15, 2009.
[24] T. Liu *et al.*, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb. 2011.
[25] J. Yang and M.-H. Yang, "Top-down visual saliency via joint CRF and dictionary learning," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2012, pp. 2296–2303.
[26] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: A data-driven approach," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, 2013, pp. 1131–1138.
[27] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: A benchmark," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Florence, Italy, 2012, pp. 414–429.
[28] Y. Li, X. Hou, C. Koch, J. M. Regh, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2014.
[29] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, "Feature selection: An ever evolving frontier in data mining," *J. Mach. Learn. Res.*, vol. 10, pp. 4–13, 2010.
[30] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," in *Data Classification: Algorithms and Applications*, C. Aggarwal, Ed. CRC Press, Jul. 2014.
[31] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 3. 2003, pp. 856–863.
[32] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[33] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," in *Proc. 27th Conf. Annu. Conf. Uncertainty Artif. Intell. (UAI)*, 2011, pp. 266–273.
[34] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.
[35] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Advances in Neural Information Processing Systems*, vol. 15. 2003, pp. 49–56.
[36] G. C. Cawley, N. L. Talbot, and M. Girolami, "Sparse multinomial logistic regression via Bayesian L1 regularisation," in *Advances in Neural Information Processing Systems*, vol. 19. 2006, p. 209.
[37] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1915–1926, Oct. 2012.
[38] R. Margolin, L. Zelnik-Manor, and A. Tal, "Saliency for image manipulation," *Vis. Comput.*, vol. 29, no. 5, pp. 381–392, May 2013.
[39] X. Hou and L. Zhang, "Dynamic visual attention: Searching for coding length increments," in *Advances in Neural Information Processing Systems*, 2008, pp. 681–688.
[40] T. N. Vikram, M. Tscherepanow, and B. Wrede, "A saliency map based on sampling an image into random rectangular regions of interest," *Pattern Recognit.*, vol. 45, no. 9, pp. 3114–3124, 2012.
[41] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *J. Vis.*, vol. 9, no. 12, pp. 1–27, 2009.
[42] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: A Bayesian framework for saliency using natural statistics," *J. Vis.*, vol. 8, no. 7, pp. 1–20, 2008.
[43] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Heraklion, Greece, 2010, pp. 366–379.
[44] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, 2005, pp. 886–893.
[45] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[46] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[47] L. Xie, Q. Tian, and B. Zhang, "Spatial pooling of heterogeneous features for image applications," in *Proc. ACM Int. Conf. Multimedia*, Nara, Japan, 2012, pp. 539–548.
[48] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.
[49] D. M. Green *et al.*, *Signal Detection Theory and Psychophysics*, vol. 1. New York, NY, USA: Wiley, 1966.
[50] R. J. Peters, A. Iyer, L. Itti, and C. Koch, "Components of bottom-up gaze allocation in natural images," *Vis. Res.*, vol. 45, no. 18, pp. 2397–2416, 2005.
[51] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," MIT Comput. Sci. Artif. Intell. Lab., Cambridge, MA, USA, Tech. Rep. MIT-CSAIL-TR-2012-001, Jan. 2012.
[52] T. Jost, N. Ouerhani, R. V. Wartburg, R. Müri, and H. Hügli, "Assessing the contribution of color in visual attention," *Comput. Vis. Image Underst.*, vol. 100, no. 1, pp. 107–123, 2005.
[53] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proc. Int. Conf. Multimedia*, 2010, Firenze, Italy, pp. 1469–1472.
[54] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Aug. 2008.
[55] A. Torralba, K. Murphy, and W. Freeman, "Sharing features: Efficient boosting procedures for multiclass object detection," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2004, pp. 762–769.

Ming Liang (S'13) received the B.E. degree in computer science and technology from the Beijing University of Aeronautics and Astronautics, Beijing, China, and the M.E. degree in computer science and technology from Tsinghua University, Beijing, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree at Tsinghua University, Beijing.

His current research interests include neural networks and their applications in computer vision.



Xiaolin Hu (S'01–M'08–SM'13) received the B.E. and M.E. degrees in automotive engineering from the Wuhan University of Technology, Wuhan, China, 2001 and 2004, respectively, and the Ph.D. degree in automation and computer-aided engineering from the Chinese University of Hong Kong, Hong Kong, in 2007.

He is currently an Associate Professor at the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His current research interests include artificial neural networks, computer vision, and computational neuroscience. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.