Large-Scale Optimization of Hierarchical Features for Saliency Prediction in Natural Images

Eleonora Vig*
Harvard University
vig@fas.harvard.edu

Michael Dorr
Harvard Medical School
michael.dorr@schepens.harvard.edu

David Cox
Harvard University
davidcox@fas.harvard.edu

Abstract

Saliency prediction typically relies on hand-crafted (multiscale) features that are combined in different ways to form a "master" saliency map, which encodes local image conspicuity. Recent improvements to the state of the art on standard benchmarks such as MIT1003 have been achieved mostly by incrementally adding more and more hand-tuned features (such as car or face detectors) to existing models [18, 4, 22, 34]. In contrast, we here follow an entirely automatic data-driven approach that performs a large-scale search for optimal features. We identify those instances of a richly-parameterized bio-inspired model family (hierarchical neuromorphic networks) that successfully predict image saliency. Because of the high dimensionality of this parameter space, we use automated hyperparameter optimization to efficiently guide the search. The optimal blend of such multilayer features combined with a simple linear classifier achieves excellent performance on several image saliency benchmarks. Our models outperform the state of the art on MIT1003, on which features and classifiers are learned. Without additional training, these models generalize well to two other image saliency data sets, Toronto and NUSEF, despite their different image content. Finally, our algorithm scores best of all the 23 models evaluated to date on the MIT300 saliency challenge [16], which uses a hidden test set to facilitate an unbiased comparison.

1. Introduction

The visual world surrounding us is astonishingly complex. Yet, humans appear to perceive their environment and navigate in it almost effortlessly. One biological key strategy to reduce the computational load and bandwidth requirements is selective, space-variant attention. Effective attentional mechanisms guide the gaze of the observer to *salient* and informative locations in the visual field. Mimicking such a selective processing has been the subject of intense research both in neuroscience [15] and in computer vision, where saliency-based preprocessing has found a wide applicability from image/video compression and quality assessment to object and action recognition.

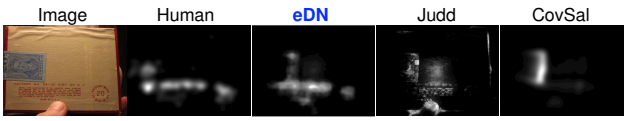

Figure 1. Saliency prediction example from the MIT300 challenge. Our model, Ensemble of Deep Networks (eDN), comes closest to human data on this benchmark; also shown are Judd and CovSal (ranked 2nd and 3rd, respectively).

Early algorithms for saliency prediction typically followed the Feature Integration Theory [31] and fused together hand-crafted image features — such as orientation, contrast, color — extracted on multiple scales. These models differ in their approach to combine individual feature maps into one "master" map. Many approaches, including the classical Itti and Koch [14] model, compute normalized center-surround difference maps of the individual features and combine these using a weighting scheme. A more recent fusion scheme expresses feature conspicuity by the equilibrium distribution of a fully connected graph [10].

Other approaches defined saliency in terms of information theory, *e.g.* by self-information [33], information maximization [5], or discriminant saliency that distinguishes target from null hypotheses [7]. Recently, spectrum-based methods demonstrated good performance despite low computational complexity [12, 11, 28].

Finally, several data-driven algorithms have been proposed that use ML techniques to predict saliency. While Kienzle *et al.* [19] directly learned a classifier from fixated image patches, many authors learned the weights associated with a set of predefined features, *e.g.* [18, 34, 4].

Recent improvements to the state of the art have been achieved mostly by incrementally adding more and more hand-tuned features to such existing models [18, 4, 22, 34]. For example, state-of-the-art performance on the MIT1003 eye-tracking benchmark is achieved by training a classifier

*Now at Xerox Research Centre Europe.

on a combination of a large variety of both low-level and high-level features, including "object" detectors for cars, faces, persons, and the horizon [18, 4].

More recently, however, image feature learning has gained momentum in the computer vision community, as a result of the approach's superior performance on several vision tasks ranging from scene classification to object and face recognition (*e.g.* [20, 24, 1]). In the context of saliency, first attempts employed independent component analysis to learn V1-like features [5] for gaze prediction, whereas the authors in [29] learned high-level concepts in an unsupervised way from fixated image patches.

In this work, we explore the usefulness of such *bio-inspired hierarchical features* to predict where people look in natural images. Our approach is structured as follows. We generate a large number of instances of a richly-parameterized bio-inspired hierarchical model family, and select those that are predictive of image saliency. The combination of several independent models should improve performance [27], but a brute-force search for the best mixture is computationally prohibitive. Therefore, we use hyperparameter optimization [2, 3] to speed up the search both for individual models and their combinations and find a discriminative image representation for saliency. We then demonstrate that a simple linear classifier that operates on such representation outperforms the state of the art on the MIT1003 benchmark [18]. Moreover, these representations that were learned on MIT1003 generalize well to two other image saliency data sets, Toronto [5] and NUSEF [26], despite their different image content. Additionally, we show that our model outperforms all 22 algorithms evaluated to date on the MIT300 saliency benchmark (see Figure 1 and [16]). Our results demonstrate that a richer, automatically-derived base of hierarchical features can challenge the state of the art in saliency prediction.

In summary, we make several contributions to the field. First, we introduce hierarchical feature learning to the area of visual saliency, yielding state-of-the-art results on four benchmarks, including an independent third-party evaluation with a hidden test set. Furthermore, we implemented an *automated* approach to optimize hierarchical features for saliency prediction, as opposed to a more traditional use of hand-tuned features. Finally, we make publicly available the software to compute our saliency maps at `http://coxlab.org/saliency`

2. Bio-inspired saliency features

We start by reviewing the broad class of biologically-inspired visual representations that we use here and present adjustments to this architecture to suit the task at hand. We then outline ways to efficiently search this vast representation space for instances that are particularly discriminative for saliency. We combine these building blocks and describe our feature learning pipeline in Section 2.3.

2.1. Richly-parameterized multilayer visual representations

Typical saliency features are hand-tuned and have a loose connection to the architecture of biological visual systems. To derive more complex, biologically more plausible saliency features, we consider a broad class of bio-inspired hierarchical models [24] that has previously been shown to excel in various recognition tasks from face verification [24] and identification to object recognition [3]. These models belong to the more general class of convolutional neural networks [21] and, accordingly, have a hierarchical multilayer structure. Inspired by the organizational principles in the primate visual cortex, hierarchies are assembled from linear filtering and non-linear transformation stages.

More specifically, each network layer is comprised of a set of operations that correspond to basic mechanisms known to take place in the visual cortex. These operations are parameterized by a large number of architectural variables, hence, are highly configurable. We can here only give a very brief overview and refer the reader to [24, 25] for more details of exact layer layout, range of parameters, and a demonstration of the effectiveness of such representations in the context of face recognition. The set of operations:

1. *Linear filtering* by convolution with a bank of random uniform filters: $F_i^l = N^{l-1} * \Phi_i^l$, where N^{l-1} is the normalized input (multichannel image or feature map) of layer l, $l \in \{1,2,3\}$, and $\Phi_i^l, i \in \{1,\dots,k^l\}$, is a random filter. This operation produces a stack of k^l feature maps F^l.
   Parameters: filter shapes $s_f^l \times s_f^l$, $s_f^l \in \{3,5,7,9\}$ and filter count $k^l \in \{16,32,64,128,256\}$
2. *Activation* with a bounded activation function: $A^l = Activate(F^l)$ such that
$$Activate(x) = \begin{cases} \gamma_{max}^l & \text{if } x > \gamma_{max}^l \\ \gamma_{min}^l & \text{if } x < \gamma_{min}^l \\ x & \text{otherwise} \end{cases} \quad (1)$$
   Parameters: $\gamma_{min}^l \in \{-\infty, 0\}$, $\gamma_{max}^l \in \{1, +\infty\}$
3. *Spatial smoothing* by pooling over a spatial neighborhood $a^l \times a^l$: $P^l = Pool(A^l)$ such that
$$P^l = Downsample_\alpha\left(\sqrt[p^l]{(A^l)^{p^l} * \mathbf{1}_{a^l \times a^l}}\right) \quad (2)$$
   Parameters: $a^l \in \{3,5,7,9\}$, exponent $p^l \in \{1,2,10\}$, α downsampling factor
4. Local *normalization* by the activity of neighbors across space ($b^l \times b^l$ neighborhood) and feature maps k^l: $N^l = Normalize(P^l)$ such that
$$N^l = \begin{cases} \frac{C^l}{\|\hat{C}^l\|_2} & \text{if } \rho^l \|\hat{C}^l\|_2 > \tau^l \\ \rho^l C^l & \text{otherwise} \end{cases} \quad (3)$$

where $C^l = P^l - \delta^l \hat{P}^l$ with $\delta^l \in \{0, 1\}$ controlling whether or not the local mean $\hat{P}^l$ is subtracted, and $\hat{C}^l = C^l * \mathbf{1}_{b^l \times b^l \times k^l}$

Parameters: ρ^l stretching param., τ^l threshold, δ^l, b^l

Any individual instantiation of this family of visual representations can contain an arbitrary number of layers that are stacked on top of each other. To constrain the computational complexity, we here consider only one- to three-layer (L1–L3) feature extractors (also called "models").

In the following, we describe our changes to generalize this architecture to the task of saliency prediction.

A major difference of this work is related to the general purpose of these features. The original approach aimed at obtaining a *compact, fixed-size representation* of an image that can then be fed into classifiers for a single, global decision (*e.g*. face identity). In contrast, here we seek *localized representations* of images. We keep the input image at high resolution and label individual pixels of the output feature map by their saliency.

Furthermore, the models in [24] were limited to grayscale input. Because of the importance of color in determining the saliency of a region we extend the models to multispectral input. Additionally, we also consider the YUV color space, which provides decorrelated luminance and chrominance channels (similar to the color space employed by the human visual system). YUV has been shown to give better results than RGB for saliency prediction (*e.g*. [28]).

In addition to evaluating individual models, we augment our feature set by *representation blending*. By combining together multiple good representations one can take advantage of the structural diversity of the individual models in the blend. Such blending strategies have been shown to significantly boost performance [27, 24].

2.2. Guided search for optimal saliency features

The guiding principle in the architectural design of the above model family was configurability. To allow for a great variety of feed-forward architectures, the model class is richly parameterized. Depending on the number of layers, models have up to 43 *architectural parameters* (see Sec. 2.1). The performance of any single model instantiation may range from chance to state-of-the-art performance depending on parameter configurations. To find good representations for a specific task, we perform a large-scale search over the space of all possible model candidates, which are cross-validated on a separate screening set.

In principle, models can be drawn entirely randomly. Indeed, random search was found to be an effective approach to search for particularly discriminative representations for recognition tasks [24]. However, for large enough problems, random search is still prohibitively computationally expensive. For example, in [24], "good" representations were found only after an exhaustive search of almost 13,000 models. Therefore, the model search should not be random but guided towards better regions of the parameter space. Recently, automated hyperparameter optimization [2, 3] was proposed to use Bayesian optimization methods to guide search in a large parameter space. In an object recognition scenario, these optimization algorithms achieved the same results as an exhaustive random search algorithm, in a fraction of the time required by random search [3]. Here, we use the publicly available toolbox of Bergstra *et al*. [3] to more efficiently search the vast bio-inspired model space for optimal saliency features. This method involves defining (i) a search space (as an expression graph) and (ii) a loss function to be minimized. In addition to a description of the bio-inspired model class, the search space also contains the hyperparameters of classifiers, such as the strength of regularization. The loss function we use is defined in the next section.

2.3. Feature learning pipeline

To evaluate the performance of our biologically-inspired representations, we follow the standard saliency learning pipeline of Judd *et al*. [18]. This offers a standardized way to evaluate new features and makes our approach directly comparable with the baseline method of [18].

Saliency prediction is formalized as a supervised learning problem. A training set of salient (*i.e*. the positive class) and non-salient (negative class) image regions is obtained from ground-truth empirical saliency maps (also called fixation density maps) derived from real eye movement data. For each image in the training set, we randomly pick 10 salient samples from the top 20% salient regions and 10 non-salient samples from the bottom 70% salient areas of the empirical saliency map. At these selected locations, features are extracted from the image and normalized (over the entire training set) to zero mean and unit variance. Finally, the labeled feature vectors are fed into an L2-regularized, linear, L2-loss SVM, which is trained to predict for each location in a new test image its probability of fixation (see Fig. 2).

To search for good representations for saliency prediction, we consider a subset of 600 images of the MIT1003 eye movement data set [18], and perform model search on this set. To estimate the prediction error, we perform 6-fold cross-validation. In the testing phase, we consider the continuous output $w^T x + b$ of the SVM. By thresholding these continuous saliency maps, image regions above the threshold are classified as salient. A systematic variation of the threshold leads to an ROC curve; the loss function to be minimized by the hyperparameter optimization algorithm (Sec. 2.2) is then 1-AUC (*i.e*. the Area Under the ROC Curve). Note that both the architectural and learning parameters (parameter C of the linear SVM) were tuned simulta-

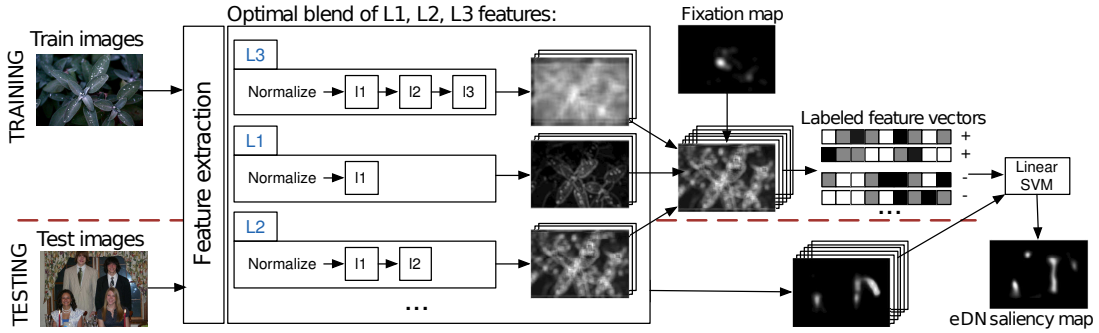

Figure 2. Schematic diagram of our pipeline. Good Li multilayer feature extractors are found by guided hyperparameter search (not shown) and combined into an optimal blend. Resulting feature vectors are labeled with empirical gaze data and fed into a linear SVM. For details on operations inside each li layer see Sec. 2.1.

neously by hyperparameter optimization.

To identify those instances (or combinations thereof) of the above described bio-inspired model family that successfully predict image saliency, we adopt a two-step search procedure. First, we search for well-performing *individual* L1, L2, and L3 models, keep these and perform another search for *optimal combinations* of these selected models. Our final saliency model is an ensemble of individual models, hence the name *Ensemble of Deep Networks (eDN)*.

In the first screening stage, we independently screened approximately 2000 L1, 2200 L2, and 2800 L3 models on RGB input. For the sake of comparison, a fraction of these models was screened with random search ($\sim$1200 L2 and $\sim$2400 L3 models) and the rest with the more efficient guided search (Sec. 2.2). We partially rely on random search, because the best configurations returned by this type of search tend to be less correlated than those found by guided search. The rationale is that a greater diversity in individual models may have a positive effect on the performance of model blends.

In addition, we separately screened about 1900 L1, 1700 L2, and 1900 L3 models on YUV input. Of these, about 600 L3 models were drawn randomly.

In the second search stage, we selected 5 to 10 top-performing models from *each* model class (a class being defined by the number of layers, the input color space, and the type of screening), and screened unconstrained combinations of up to 8 of these models. The best ensemble was found after around 1000 trials and consists of 6 architecturally diverse models: three L3-RGB models found with random search, and one L2-YUV and two L3-YUV models found with guided search. We note that other combinations, including those with L1 models, were also found among top-performing results.

Fig. 3 shows example outputs of individual models and the optimal blend. The entire screening was performed on a large CPU cluster and took approximately 1 week.

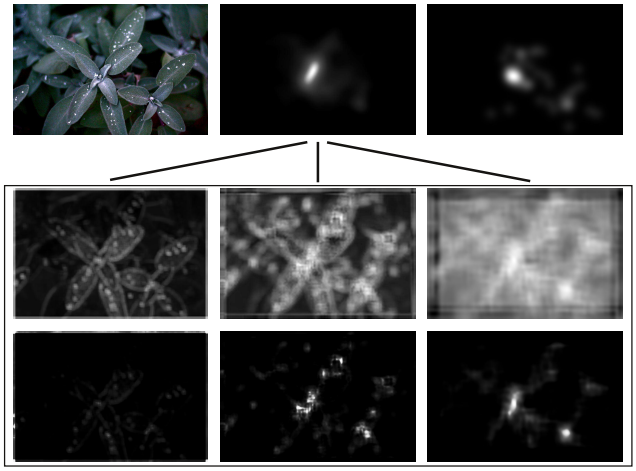

Figure 3. Output of best individual L1, L2, L3 (YUV) models and the optimal blend. Left to right, top: input, histogram-matched eDN saliency map (blend of 6 Li models), human saliency map; Middle: raw L1, L2, and L3 output; Bottom: L1, L2, L3 output histogram-matched to the human saliency map. While L1 models detect edges, L2 and L3 capture higher-order structures.

2.4. Center bias and smoothness

Several studies have pointed out that gaze is biased to the image center (often irrespective of the center features). More and more models explicitly incorporate this bias (e.g. [17, 32]). We follow [18] and extend our learned features with a simple 1-D measure of the distance of each pixel to the image center. Because some reference models do not account for this bias, for a fair comparison, we incorporate the same distance-to-center feature in these models as well. In addition, we report results also without this center feature. However, it should be noted that some algorithms already have an implicit center bias due to border effects.

As a final step, saliency maps are often smoothed to bet-

ter match the distribution of eye movements [17]. We determine the optimal Gaussian blur level (σ=30px) through cross-validation on a subset of the training set.

3. Evaluation: eye movement prediction

Because of the tight link between saliency, attention, and eye movements, saliency models are typically evaluated in terms of how well they predict human gaze in natural scenes. We follow the same approach here.

3.1. Data sets

Feature search and performance evaluation were conducted on the MIT1003 [18] data set. To test the generalizability of the model, we use three additional benchmarks, Toronto [5], NUSEF [26], and MIT300 [16].

The MIT1003 data set [18] consists of 1003 images and the eye movement data of 15 observers who free-viewed these images. Because the feature search stage used 600 of these images for training, the remaining 403 served as test images. The Toronto benchmark [5] contains eye movement data from 20 subjects watching 120 images. It is considered a "hard" data set due to the lack of particularly salient regions (such as faces/people). The recently proposed NUSEF [26] is made up of 758 semantically-rich images including affective content (expressive faces, nudes, unpleasant concepts). A total of 75 observers viewed parts of the image set. We here only consider the publicly available part (441 images).

Finally, for an independent evaluation, we submitted our model for third-party evaluation on the MIT300 saliency benchmark [16]. This set consists of 300 images viewed by 39 observers whose gaze data is not public. To date, 23 saliency models have been evaluated on this platform.

3.2. Evaluation protocol

We follow the evaluation procedure outlined in Sec. 2.3. In addition to ROC analysis, for the sake of comprehensiveness, we consider three other common evaluation metrics: the Earth Mover's Distance (EMD, in the context of saliency see [34]), Normalized Scanpath Saliency (NSS) [23], and a similarity score [17].

As reference for comparisons, we consider 11 state-of-the-art saliency algorithms: GBVS [10], the multi-scale quaternion DCT signature on YUV input (denoted ΔQDCT) [28], Judd [18], AWS [8], CovSal [6] (with covariances + means), Tavakoli [30], AIM [5], Goferman [9], ICL [13], Image Signature (with LAB color space) [11], and Boost [4], all with default parameters. This selection is based on the top-performing models from two recent and comprehensive reviews/taxonomies [4, 16].

To estimate the effective performance range, we used two control measures. First, we computed "leave-one-subject-out" empirical saliency maps and used these to predict the eye movements of the left-out viewer. Since people are still the best predictors of where other people look, this measure constitutes an upper bound for prediction. Our lower bound comes from the above-mentioned center bias (*e.g.* [32]): a simple measure based on the distance of each pixel to the image center predicts eye movements well.

Model	RGB	YUV
L1	0.6744	0.6705
L2	0.6737	0.7401
L3	0.7207	0.7977
eDN	0.8227	

Table 1. AUC scores of best individual models and the optimal ensemble of top models on MIT1003. Performance increases with model complexity (*i.e.* # layers) and the use of YUV. The optimal ensemble of Deep Networks (also found through automated screening) gives highest performance. No center bias at this point.

3.3. Results on the MIT1003 benchmark

First, we analyzed individual model performance by systematically varying two meta-parameters: the number of layers in each model (1, 2, or 3) and the input color space (RGB or YUV). Performance (AUC) of the best class-specific models is shown in Table 1. Consistent with expectations, performance increases with model complexity (L1 $<$ L2 $<$ L3), *i.e.* models with more layers achieve higher invariance. Confirming previous findings, use of the YUV color space gives significantly higher prediction performance than RGB. For the choice of the search algorithm, guided search not only is more efficient than random search (360 iterations give better results than 2400 w/o guidance) but also exceeds the best random search performance.

The most significant performance boost is achieved with ensembles of multiple good representations (see Table 1, eDN). In contrast to [25], however, blends are not limited to the combination of the top models, but derived — again in an automated fashion — through guided screening. Blending is so beneficial because combinations take advantage of the diversity of individual models. This explains why individually weaker models (such as L3-RGB or L2-YUV) are also represented in the best blend. One aspect of diversity is the scale on which individual models operate. From the combination of multiple models tuned to different scales, a multiscale model emerges.

Performance of the various saliency algorithms (baselines, controls, and our best blend) is summarized as averages over all MIT1003 test images in the top part of Table 2. eDN outperforms all individual saliency algorithms for all four metrics. A small further performance gain (for AUC and EMD) can only be achieved by Boosting [4], *i.e.* optimally blending many top-performing algorithms from this

		AUC		EMD		sim		NSS	
		w/ C	w/o C	w/ C	w/o C	w/ C	w/o C	w/ C	w/o C
MIT1003	Chance	–	0.4997	–	6.8160	–	0.2183	–	0.0092
	Cntr	0.7933	–	3.9998	–	0.3501	–	0.9906	–
	ΔQDCT	0.8148	0.7628	4.0672	4.9867	0.3623	0.3039	1.1769	0.9279
	ICL	0.8213	0.7720	3.8159	4.8848	0.4023	0.3222	1.4029	1.0442
	CovSal	0.8214	0.7641	3.7077	4.9680	0.3952	0.2924	1.2787	0.9101
	Signature	0.8248	0.7665	3.7262	4.9376	0.4095	0.3190	1.4036	1.0468
	GBVS	0.8266	0.8097	3.7077	3.9457	0.3800	0.3467	1.2818	1.1168
	Tavakoli	0.8314	0.7711	3.8219	4.6978	0.4085	0.2978	1.3931	0.8716
	Goferman	0.8323	0.7625	3.6082	5.1569	0.4167	0.2970	1.4512	0.9002
	AIM	0.8384	0.7716	3.6240	4.8270	0.4207	0.3018	1.4798	0.9400
	Judd	0.8395	0.7892	3.5446	4.5729	0.4226	0.3425	1.5133	1.1637
	AWS	0.8429	0.7530	3.5658	5.4005	0.4461	0.3129	1.6951	1.0965
	eDN	0.8504	0.8227	3.4513	3.9099	0.4425	0.3780	1.6131	1.2765
	Boost	0.8512	–	3.4174	–	0.4370	–	1.5282	–
	Boost+eDN	0.8546	–	3.4616	–	0.4473	–	1.6577	–
	Human	0.9008	–	0	–	1	–	3.2123	–
Toronto	Chance	–	0.4988	–	4.1171	–	0.3040	–	-0.0038
	Cntr	0.7836	–	2.5828	–	0.4116	–	0.8180	–
	CovSal	0.8147	0.7616	2.3712	3.1187	0.5122	0.3812	1.4587	1.0090
	ICL	0.8169	0.7807	2.4190	2.9943	0.5253	0.4333	1.6199	1.2483
	Signature	0.8188	0.7903	2.3776	2.9054	0.5311	0.4658	1.5888	1.4464
	GBVS	0.8274	0.8152	2.2494	2.2922	0.5335	0.4955	1.6293	1.4494
	ΔQDCT	0.8276	0.7777	2.3663	3.0311	0.5045	0.4178	1.4901	1.1647
	Goferman	0.8295	0.7795	2.2490	3.1013	0.5553	0.4270	1.7712	1.2134
	Tavakoli	0.8330	0.7742	2.6605	2.8214	0.5422	0.4068	1.5851	0.9972
	AIM	0.8341	0.7853	2.2517	2.7369	0.5491	0.4262	1.6600	1.1640
	Judd	0.8381	0.7913	2.2651	3.0407	0.5520	0.4428	1.7197	1.2777
	AWS	0.8400	0.7542	2.3883	3.5976	0.5577	0.4073	1.7817	1.1991
	eDN	0.8407	0.8152	2.2394	2.4384	0.5730	0.4870	1.7149	1.3126
	Boost	0.8436	–	2.2217	–	0.5653	–	1.6954	–
	Boost+eDN	0.8467	–	2.3077	–	0.5741	–	1.8213	–
	Human	0.8820	–	0	–	1	–	2.5463	–
Nusef	Chance	–	0.4995	–	5.6304	–	0.3356	–	0.0048
	ΔQDCT	0.7847	0.7357	3.7385	4.3315	0.4556	0.3947	0.9523	0.7554
	Cntr	0.7851	–	3.6228	–	0.4395	–	0.7710	–
	Tavakoli	0.7948	0.7198	3.9202	4.1310	0.4865	0.3677	1.0800	0.5800
	ICL	0.7963	0.7354	3.5599	4.3918	0.4989	0.3985	1.1038	0.7625
	GBVS	0.8032	0.7884	3.5051	3.5754	0.4858	0.4567	1.0696	0.9559
	Signature	0.8033	0.7327	3.5382	4.4265	0.5205	0.3929	1.2143	0.7667
	CovSal	0.8046	0.7162	3.5064	4.4868	0.5212	0.3661	1.1875	0.6285
	Goferman	0.8062	0.7342	3.5211	4.4372	0.5172	0.3879	1.1977	0.7068
	Judd	0.8130	0.7562	3.5319	4.3700	0.5263	0.4302	1.2814	0.9389
	AIM	0.8133	0.7476	3.5364	4.2570	0.5281	0.4013	1.2790	0.8189
	AWS	0.8144	0.7290	3.5550	4.7787	0.5350	0.4007	1.3260	0.8681
	eDN	0.8242	0.8019	3.5360	3.7997	0.5509	0.4973	1.3879	1.2177
	Human	0.8407	–	0	–	1	–	1.9543	–

Table 2. Performance of saliency algorithms – with (w/ C) and without (w/o C) center bias – on the MIT1003, Toronto, and NUSEF benchmarks for four metrics: AUC, similarity (sim), Normalized Scanpath Saliency (NSS) (the higher the better for all three) and EMD (lower better). Feature search only used AUC as objective function. eDN outperforms individual models on MIT1003, on which features and classifier are learned. A small further improvement can only be achieved by blending multiple individual top-performing algorithms (see Boost [4] and Boost+eDN). eDN generalizes well to Toronto and NUSEF, despite their different image content. Small border artifacts (due to repeated filtering) make eDN inherently biased to the center – hence the performance advantage in "w/o C" case.

table (Judd [18], GBVS [10], AWS [8] and Torralba). We also note that Boosting results get better with inclusion of eDN features, *e.g.* 0.8546 AUC for MIT1003 (see Table 2).

Many image processing steps used in saliency computations introduce border artifacts (see [33] for a detailed analysis of several algorithms). Because of repeated filtering across multiple layers, our saliency maps suffer from some border effects as well. Hence, similarly to GBVS, even our non-centered maps ("w/o C" in Table 2) are still implicitly biased to the center and have a performance advantage.

Example saliency maps for our algorithm and some reference methods are shown in Fig. 4.

3.4. Generalization to other data sets

To assess how well our learned features generalize to other data sets, we first evaluated them on the Toronto and NUSEF benchmarks. Results are shown in Table 2 (lower part). Despite being trained on a different data set, eDN outperforms the state-of-the-art approaches. This is surprising, considering the significant database-specific differences in image content. NUSEF was deliberately created to investigate semantics-driven attention to affective content. Conversely, Toronto lacks particular regions of interest (people, faces), so that gaze is less coherent across viewers. Finally, on the MIT300 saliency benchmark, our model achieves 0.8192 AUC (0.5123 similarity and 3.0129 EMD), slightly better than the second best model of Judd et al. with 0.811 AUC (0.506 similarity and 3.13 EMD) — see [16].

4. Discussion and conclusion

Hierarchical feature learning has become a common approach in computer vision, but has not been adequately explored in the context of saliency prediction. Here, we addressed this issue and efficiently searched a large pool of richly parameterized neuromorphic models for those representations that are discriminative for saliency. Through automated representation blending, we assembled powerful combinations of diverse multilayer architectures that outperform the state of the art. In notable contrast to top-performing hand-tuned models, our approach makes no assumptions about what lower-level features (color, contrast, etc.) or higher-level concepts (faces, cars, text, horizon) attract the eyes. Instead, we allow the hierarchical models to learn such complex patterns from gaze-labeled natural images (*e.g.* see top row in Figure 4). We believe our integrated and biologically-plausible approach is therefore conceptually cleaner and more generic than approaches that rely on domain-specific hand-crafted features. Although trained only on part of the MIT1003 benchmark, our representations generalize well to three other eye movement data sets, in spite of their different image content. Despite the large size of the model space (*i.e.* the model family is as inclusive as possible), good candidates are found quickly (within a couple of hundred trials) through novel hyperparameter optimization algorithms.

Our results show that methods employing rich, automatically-derived feedforward representations can challenge the state of the art in the field of saliency prediction.

Acknowledgments.

This work was supported by the National Science Foundation (IIS 0963668) and a gift from Intel Corporation. Eleonora Vig was supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD, D/11/41189).

References

[1] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *arXiv preprint arXiv:1206.5538*, 2012. 2

[2] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *NIPS 25*, 2011. 2, 3

[3] J. Bergstra, D. Yamins, and D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *ICML*, 2013. 2, 3

[4] A. Borji. Boosting bottom-up and top-down visual features for saliency estimation. In *CVPR*, 2012. 1, 2, 5, 6

[5] N. Bruce and J. Tsotsos. Saliency based on information maximization. In *NIPS 18*, pages 155–162. 2006. 1, 2, 5

[6] E. Erdem and A. Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of vision*, 13(4), 2013. 5

[7] D. Gao and N. Vasconcelos. Bottom-up saliency is a discriminant process. In *ICCV*, pages 1–6, 2007. 1

[8] A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil. Decorrelation and distinctiveness provide with human-like saliency. In *Advanced Concepts for Intelligent Vision Systems*, pages 343–354, 2009. 5, 7

[9] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *PAMI*, 34(10):1915–1926, 2012. 5

[10] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS 19*, pages 545–552, 2007. 1, 5, 7

[11] X. Hou, J. Harel, and C. Koch. Image signature: Highlighting sparse salient regions. *PAMI*, 34(1), 2012. 1, 5

[12] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, pages 1–8, 2007. 1

[13] X. Hou and L. Zhang. Dynamic visual attention: Searching for coding length increments. In *NIPS*, 2008. 5

[14] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE PAMI*, 20(11):1254–1259, 1998. 1

[15] L. Itti, G. Rees, and J. K. Tsotsos, editors. *Neurobiology of Attention*. Elsevier, San Diego, CA, 2005. 1

[16] T. Judd, F. Durand, and A. Torralba. MIT saliency benchmark. http://people.csail.mit.edu/tjudd/SaliencyBenchmark/. 1, 2, 5, 7

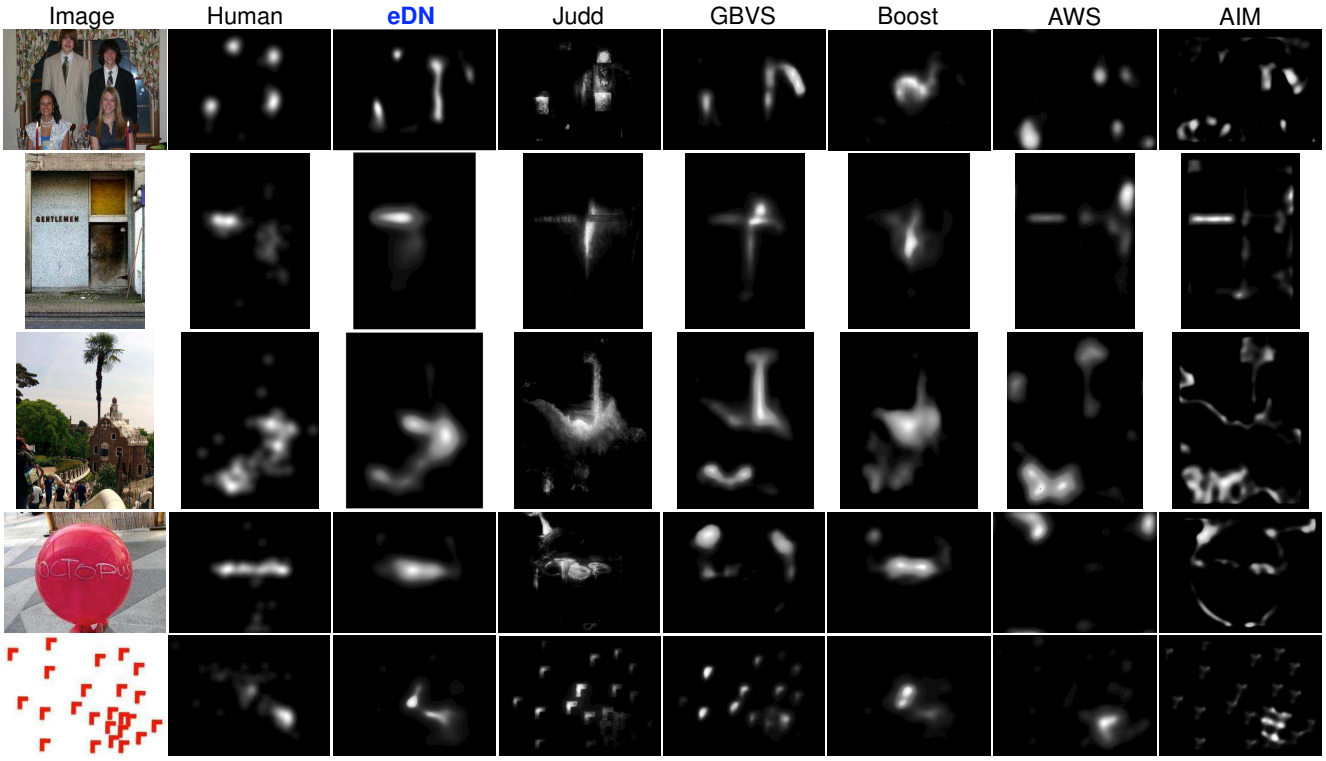

Figure 4. Sample images from MIT1003 [18] with ground truth (Human), proposed (eDN) and reference saliency maps. Saliency maps were histogram-matched to the corresponding fixation density map for visualization. eDN maps are more consistent with human maps.

[17] T. Judd, F. Durand, and A. Torralba. A benchmark of computational models of saliency to predict human fixations. *MIT tech report*, 2012. 4, 5

[18] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, pages 2106–2113, 2009. 1, 2, 3, 4, 5, 7, 8

[19] W. Kienzle, F. A. Wichmann, B. Schölkopf, and M. O. Franz. A Nonparametric Approach to Bottom-Up Visual Saliency. In *NIPS*, pages 689–696, 2007. 1

[20] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS 25*, pages 1106–1114, 2012. 2

[21] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361, 1995. 2

[22] Y. Lu, W. Zhang, C. Jin, and S. Xue. Learning attention map from images. In *CVPR*, pages 1067–1074, 2012. 1

[23] R. J. Peters, A. Iyer, L. Itti, and C. Koch. Components of bottom-up gaze allocation in natural images. *Vision research*, 45(18):2397–2416, 2005. 5

[24] N. Pinto and D. D. Cox. Beyond Simple Features: A Large-Scale Feature Search Approach to Unconstrained Face Recognition. *IEEE Automated Face and Gesture Recognition*, 2011. 2, 3

[25] N. Pinto, D. Doukhan, J. J. DiCarlo, and D. D. Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Comput Biol*, 2009. 2, 5

[26] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T.-S. Chua. An eye fixation database for saliency detection in images. In *ECCV*, pages 30–43, 2010. 2, 5

[27] R. E. Schapire. The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification. Lecture Notes in Statist.*, pages 149–172, 2003. 2, 3

[28] B. Schauerte and R. Stiefelhagen. Quaternion-based spectral saliency detection for eye fixation prediction. In *ECCV*, pages 116–129, 2012. 1, 3, 5

[29] C. Shen, M. Song, and Q. Zhao. Learning high-level concepts by training a deep network on eye fixations. In *NIPS Deep Learning and Unsup Feat Learn Workshop*, 2012. 2

[30] H. R. Tavakoli, E. Rahtu, and J. Heikkilä. Fast and efficient saliency detection using sparse sampling and kernel density estimation. In *Image Analysis*, pages 666–675. 2011. 5

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

[32] P. Tseng, R. Carmi, I. Cameron, D. Munoz, and L. Itti. Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of Vision*, 9(7), 2009. 4, 5

[33] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):1–20, 12 2008. 1, 7

[34] Q. Zhao and C. Koch. Learning a saliency map using fixated locations in natural scenes. *Journal of Vision*, 11(3), 2011. 1, 5