Journal of Electronic Imaging

JElectronicImaging.org

Multiscale fully convolutional network for image saliency

Simone Bianco
Marco Buzzelli
Raimondo Schettini

SPIE. IS&T imaging.org

Simone Bianco, Marco Buzzelli, Raimondo Schettini, "Multiscale fully convolutional network for image saliency," *J. Electron. Imaging* **27**(5), 051221 (2018), doi: 10.1117/1.JEI.27.5.051221.

Multiscale fully convolutional network for image saliency

Simone Bianco, Marco Buzzelli,* and Raimondo Schettini
Università degli Studi di Milano-Bicocca, Dipartimento di Informatica, Sistemistica e Comunicazione, Milano, Italy

Abstract. We focus on saliency estimation in digital images. We describe why it is important to adopt a data-driven model for such an illposed problem, allowing for a universal concept of "saliency" to naturally emerge from data that are typically annotated with drastically heterogeneous criteria. Our learning-based method also involves an explicit analysis of the input at multiple scales, in order to take into account images of different resolutions, depicting subjects of different sizes. Furthermore, despite training our model on binary ground truths only, we are able to output a continuous-valued confidence map, which represents the probability of each image pixel being salient. Every contribution of our method for saliency estimation is singularly tested according to a standard evaluation benchmark, and our final proposal proves to be very effective in a comparison with the state-of-the-art. *© 2018 SPIE and IS&T* [DOI: 10.1117/1.JEI.27.5.051221]

Keywords: saliency estimation; foreground/background segmentation; fully convolutional neural network; multiscale.
Paper 180062SSP received Jan. 15, 2018; accepted for publication Apr. 20, 2018; published online May 12, 2018.

1 Introduction

Estimation of image saliency can be defined as the task of assigning different levels of visual relevance to different regions in a digital image. Automating such process would be a helpful resource for object recognition, adaptive image and video compression, content-aware image editing, image retrieval, and object-level image manipulation.[1] Despite the clear advantage that would be gained from solving this task, there is no universally accepted definition on what makes an element "salient," thus rendering saliency estimation particularly challenging. This can be better seen by observing Fig. 1; while the main object of interest in the first image can be generally recognized as the butterfly itself, the other two examples present less obvious answers. Figure 1(b) shows a crowded dining scene with no clear main subject. The annotators of the corresponding dataset[3] addressed this problem by assigning a decreasing level of saliency to each segmented element in the images. Note that this saliency rank was computed by collecting gaze data from multiple observers. In a similar fashion, Fig. 1(c) provides another nontrivial example, annotated in the corresponding dataset[4] with only the most looked-at region, according to human observers.

Both the extreme subjectivity intrinsic to the annotation task and the criteria heterogeneity adopted by different dataset curators contribute to making even more difficult a problem that is illposed in the first place. Methods for saliency estimation that are based on handcrafted low-level features have always struggled in reaching good performance.[1] The observed characteristics suggest, in fact, that a data-driven model with at least some level of semantic awareness would be essential to properly address the proposed task. This kind of solution would also allow for a universal concept of image saliency to naturally emerge from a large set of heterogeneously annotated data. Bianco et al.[5] successfully embedded these elements and presented a learning-based approach to salient object detection that outperforms all competing methods according to multiple evaluation measures from a standard benchmark. In this work, we complement such method by performing a multiscale analysis of the input image and by producing a continuous-valued output saliency map. The combination of these elements provides an extrajump in saliency estimation accuracy, as proven with experiments on several standard datasets.

Section 2 describes a standard benchmark for salient object detection, as well as an overview of the top performing methods on the benchmark itself. Section 3 shows the proposed method for saliency estimation, defining both the basic idea and the introduced elements. Section 4 presents three sets of experiments, aimed, respectively, at assessing the contribution of producing a continuous-valued output, finding the best configuration for multiscale analysis, and comparing the final solution against the adopted benchmark.

2 Related Works and Evaluation Benchmarks

The literature on the subject of saliency estimation presents a vast landscape of different approaches to the problem. The great majority of such methods involves the definition of handcrafted features and rules[6–8] or the adoption of optimization strategies.[9,10] Machine learning approaches, instead, treat the problem from a data-driven perspective: in Ref. 5, for example, we proposed using a fully convolutional neural network architecture, which makes it possible to produce a dense (i.e., per-pixel) estimation of the saliency feature. By pretraining such model on tasks such as semantic segmentation,[11] it is also possible to introduce in the model middle-level features that prove to be useful for semantically sensitive tasks in general, such as saliency estimation itself. Recurrent fully convolutional networks (RFCN)[12] adopt a different strategy built on top of the concept of FCNs, integrating it with backward selfcorrecting

*Address all correspondence to: Marco Buzzelli, E-mail: marco.buzzelli@disco.unimib.it

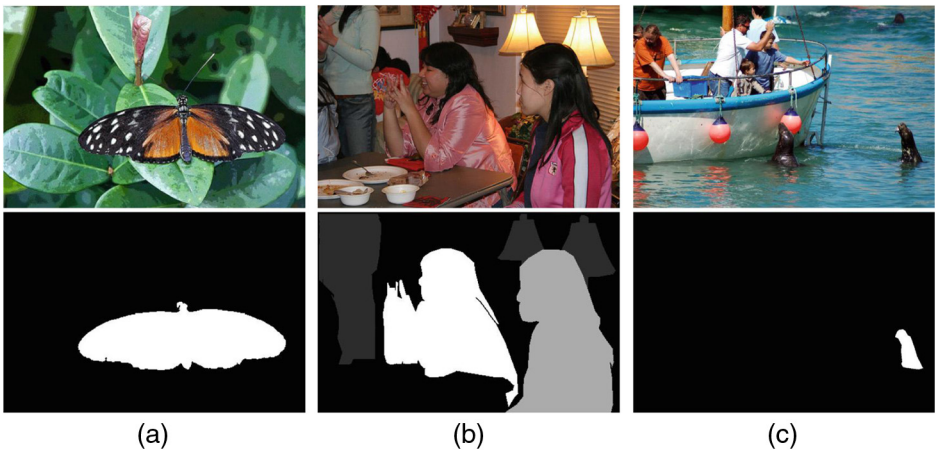

Fig. 1 Difference in saliency annotation strategy for three datasets: (a) THUR15K,[2] (b) PASCAL-S,[3] and (c) JuddDB.[4]

connections as well as saliency prior knowledge. Multicontext deep learning (MC),[13] one of the first solutions to address saliency estimation with the use of convolutional neural networks, defines a unified framework to represent both global and local context in a data-driven fashion. The authors of deeply supervised saliency (DSS)[14] introduce short connections to the skip-layer structures described by the holistically nested edge detector architecture,[15] providing an alternative way to generate rich multiscale feature maps at different layers.

Finding a common evaluation ground for this task is revealed to be particularly challenging, especially among methods that use convolutional neural networks: different authors often test their solutions in different experimental setups, producing an extreme sparsity among datasets and adopted evaluation measures. At the same time, it is important to test under comparable environments, especially in a situation where the observed high annotation heterogeneity might lead to potentially different interpretations of the concepts of saliency. There exist very few works aimed at defining a standardized benchmark for saliency estimation, the most successful being described by Borji et al.[1] This benchmark compares more than 40 methods on seven datasets, using several evaluation measures aimed at assessing different aspects of the analyzed algorithms. The authors also host a public web page,[16] where researchers can submit their own solutions for evaluation and inclusion in the official leaderboards. We choose this particular benchmark for its large availability of datasets, measures, and compared methods, and we intend to re-establish its status of global standard benchmark. We also propose the introduction of a leave-one-dataset-out (LODO) setup for training, later described in Sec. 4, which allows for a fair comparison with existing methods, and at the same time provides an effective learning environment for upcoming deep learning methods.

In the following, we describe the five best-performing methods from the adopted benchmark, which are used in Sec. 4.3 as a direct comparison with our proposed solution. The method presented in discriminative regional feature integration (DRFI)[6] builds a multilevel representation of the input image and creates a regression model mapping the regional feature vector of each level to the corresponding saliency score. These scores are finally fused in order to determine the complete saliency map. In quantum cut (QCUT),[9] authors model salient object segmentation as an optimization problem. They, then exploit the link between quantum mechanics and graph-cuts to develop an object segmentation method based on the ground state solution of a modified Hamiltonian. The authors of minimum barrier distance (MBD)[7] presented an approximation of the MBD transform and combined it with an appearance-based backgroundness cue. The resulting method performs significantly better than other solutions having the same computational requirements. In saliency tree (ST),[8] authors simplify the image into primitive regions, with associated saliency based on multiple handcrafted measures. They generated a saliency tree using region merging and performed a systematic analysis of such tree to derive the final saliency map. Robust background detection (RBD)[10] introduces boundary connectivity: a background measure based on an intuitive geometrical interpretation. This measure is then used along with multiple low-level cues to produce saliency maps through a principled optimization framework.

3 Proposed Method for Saliency Estimation

We propose a fully convolutional neural network (FCN[11]) that exploits layers previously trained on recognizing 1000 object classes[17] as the starting point for a deep analysis of the original input image, in order to produce a per-pixel estimation of its saliency. The resulting architecture, after being properly trained, will be able to generate an estimation of object saliency that transcends the 1000 classes defined for the pretraining. These classes are, in fact, used to build a semantically aware internal representation, but do not constrain the type of objects that can be identified as being "salient." A simple proof of this is the "person" category, which is absent from the original set of classes but well prominent in the final saliency estimation results, as shown in Sec. 4. The overall structure of the proposed architecture is shown in Fig. 2, and details are provided in Table 1: the output of layer conv5-3 from a VGG-19 network (visual geometry group[18]) is mapped to the final problem size (i.e., two channels for "salient" and "nonsalient") by using a series of pooling, convolution, ReLU, and dropout blocks. The result is then combined with the outputs of pool4 and pool3 by direct sum. Since these activations all have a different spatial resolution, two convolutional-transpose layers (also known as fractionally strided convolutions) are used

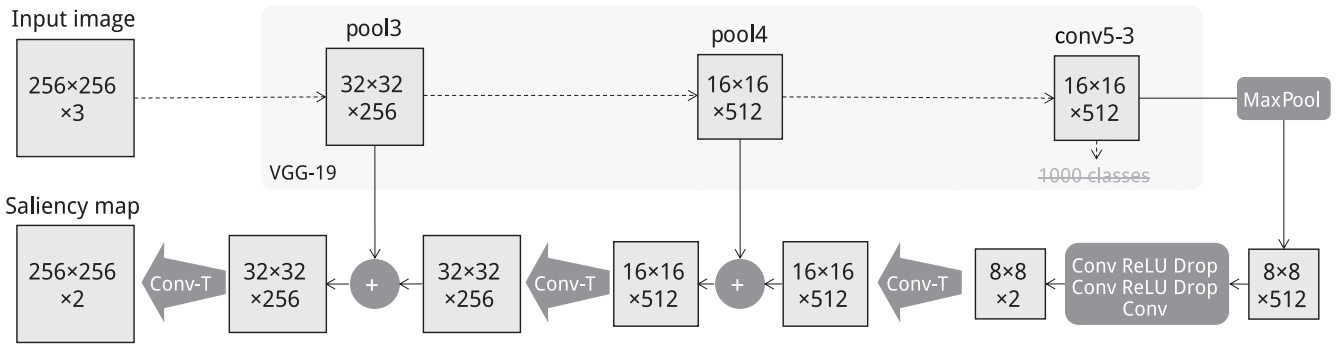

Fig. 2 Activations of the fully convolutional network employed for saliency estimation, with an input example of resolution 256×256 pixels. Details of the VGG-19 network[18] are omitted for ease of visualization.

Table 1 Details of the adopted fully convolutional architecture, with an input example of resolution 256×256 pixels. Layers marked with square brackets come from the original VGG-19 network,[18] whose details are omitted for ease of visualization.

Operation	Input layers	Output layer	Filter size			Stride	Output size
			Kernel	Ch. in	Ch. out		
MaxPool	[conv5-3]	pool5	2×2	—	—	2	$8 \times 8 \times 512$
Conv	pool5	conv6	7×7	512	4096	1	$8 \times 8 \times 4096$
ReLU	conv6	relu6	—	—	—	—	$8 \times 8 \times 4096$
DropOut	relu6	drop6	—	—	—	—	$8 \times 8 \times 4096$
Conv	drop6	conv7	1×1	4096	4096	1	$8 \times 8 \times 4096$
ReLU	conv7	relu7	—	—	—	—	$8 \times 8 \times 4096$
DropOut	relu7	drop7	—	—	—	—	$8 \times 8 \times 4096$
Conv	drop7	conv8	1×1	4096	2	1	$8 \times 8 \times 2$
Conv-T	conv8	convT1	4×4	2	512	1/2	$16 \times 16 \times 512$
Sum	[pool4], convT1	sum1	—	—	—	—	$16 \times 16 \times 512$
Conv-T	sum1	convT2	4×4	512	256	1/2	$32 \times 32 \times 256$
Sum	[pool3], convT2	sum2	—	—	—	—	$32 \times 32 \times 256$
Conv-T	sum2	convT3	16×16	256	2	1/8	$256 \times 256 \times 2$

to bring them to a compatible size, and a third one is used to map the result to the original input size. The whole network is trained end-to-end, eventually updating also the prelearned weights that were used to initialize the VGG-19 module. This starting solution, first presented in Ref. 5 and inspired by Ref. 11, is here extended and complemented with two elements: continuous-valued prediction and multiscale analysis, which are experimentally proven to increase the accuracy of the saliency estimation.

3.1 *Continuous-Valued Prediction*

Most available datasets for saliency estimation and foreground detection are published with a binary ground truth.[4,2,19–21] It is therefore natural to approach the problem as a per-pixel binary classification task, so we train our FCN with a per-pixel softmax cross-entropy loss (the global loss of each minibatch is computed by averaging all loss values from the single pixels involved). For datasets providing discrete annotations,[3,22] we apply a preprocessing threshold, setting to 1 all values greater than 0. At inference time, it is then possible to stop the network processing right after the softmax layer, in order to effectively produce two complementary continuous maps, which respectively represent the probability of each pixel being, or not being, salient. If necessary, the saliency channel can then be binarized by applying a 0.5 threshold (equivalent to taking the argmax

between the two complementary channels, as was previously done[5]), or using any of the thresholding techniques described in Sec. 4 for evaluation.

3.2 *Multiscale Analysis*

As both the input image and the portrayed elements could be of any size, it is fundamental to create a model for saliency estimation that is able to analyze the image at different scales. This kind of multiscale awareness should affect the training procedure, introducing into the learned model pieces of information that come from observing the annotated data at various scales, as well as the inference procedure, collecting saliency cues at different levels and appropriately combining them into one final output.

At training time, we can obtain this effect as shown in Fig. 3(a), by cropping subregions of random size from the input images and annotations, and eventually bringing them to a common resolution (in our case, 256×256 pixels) in order to exploit fast minibatch parallelization. This last resizing step will indeed destroy all information about the relationship between the crop and the rest of the image but will work as a form of data augmentation to take into account the diversity of subject size that can be encountered at test time. The effective advantage of this approach was established in Ref. 5, where it was proven to be the most-effective type of data augmentation for our task, compared to random flip and random gamma correction.

At inference time, the fully convolutional nature of our model makes it possible to process an input of any size and consequently produces an output of the same dimensions. This, however, does not guarantee a multiscale analysis of the whole image. The neural model will, in fact, only analyze the input image on subregions of limited size (the receptive field is 32×32 pixels) and efficiently apply this processing in a sliding-window fashion over the whole image. In order to explicitly perform multiscale analysis, we need to create copies of the input image at different resolutions (a so-called image pyramid[23]), apply the network as a sliding-window over each pyramid level, rescale all the predictions to the size of the original input, and merge the results. This procedure is shown in Fig. 3(b). Different scales and different merging strategies can be adopted as investigated in Sec. 4.

4 Experiments

The following experiments are designed to quantify the practical contribution of each individual element of the proposed method for saliency estimation, in particular: continuous-valued prediction and multiscale analysis at inference time. Our combined solution is then evaluated against a standard set of methods for saliency estimation. All conducted tests follow the benchmark proposed in Ref. 1 in terms of datasets and evaluation measures.

The seven tested datasets, presented in Table 2, offer different types of images and are annotated with sometimes drastically different criteria, as noted in Sec. 1. For the purpose of these experiments, we adopt an LODO setup, as the original benchmark does not provide an official training-test split for each dataset. This solution allows for a fair comparison with methods that do not involve an explicit training phase on a set of annotated examples. At the same time, it guarantees overfitting-free results, as our model is never tested on the same annotation criteria that are used during the training phase. Cross-dataset near-duplicate removal is also conducted according to the methodology described in Refs. 5 and 24, to avoid the same images being present at both training and test time in our LODO setup: based on a search on more than 200 million image pairs using structural similarity (SSIM[25]), five images were found to be shared among different datasets. Although too few to meaningfully influence the final performance evaluation, we nonetheless exclude them from the set of training examples whenever they are present in the test set.

Different metrics are used to analyze different aspects of the estimated image saliency:

F-measure (F_β) is the weighted harmonic mean between precision and recall

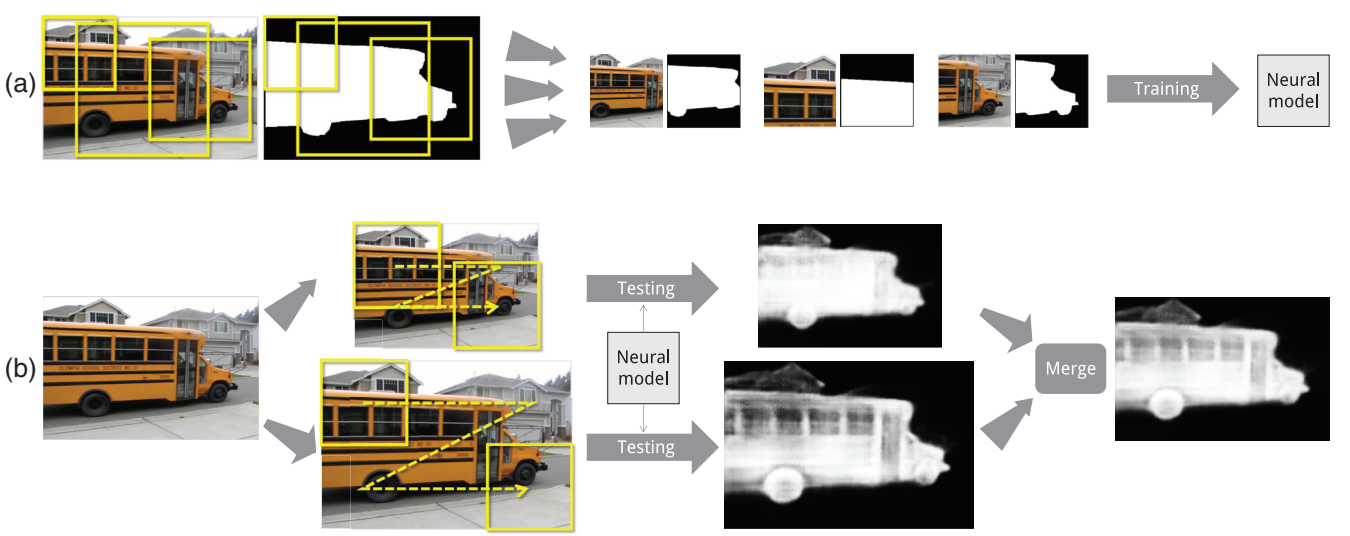

Fig. 3 Schematic view of multiscale analysis at (a) training and (b) inference time.

Table 2 Summary of tested datasets.

Dataset	Images	Average size (px)	Notes
MSRA10K[19]	10,000	400×300	—
THUR15K[2]	6233	450×300	Only 6233/15,000 annotated images
DUT-OMRON[21]	5166	400×300	—
ECSSD[20]	1000	400×300	—
JuddDB[4]	900	1024×768	Salient object typically very small
PASCAL-S[3]	850	500×350	High background clutter
SED2[22]	100	300×250	Two salient objects per picture

$$F_\beta = \frac{(1+\beta^2)\text{precision} \times \text{recall}}{\beta^2 \text{precision} + \text{recall}}. \tag{1}$$

In order to give more weight to precision, which is considered to be more important than recall for our task,[1,19,26] parameter β^2 is set to 0.3. The continuous-valued saliency estimation can be binarized with different techniques before effectively computing precision and recall. The adopted benchmark presents three alternatives way to perform such binarization:

1. Varying fixed threshold: Precision and recall are computed at all integer thresholds between 0 and 255, and then averaged.
2. Adaptive threshold:[26] The threshold for binarization is set to twice the mean value of the predicted saliency map.
3. Saliency cut:[27] The threshold is set to a low value, thus granting high recall rate. Segmentation algorithm GrabCut[28] is then iteratively applied to the binarized prediction, typically producing a saliency estimation with more precise edges.

Area under curve (AUC) is the area under the receiver operating characteristic curve (ROC). The ROC curve is in turn computed by varying the binarization threshold and plotting true-positive rate (TPR) versus false-positive rate (FPR) values

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{3}$$

Mean absolute error (MAE) is computed directly on the prediction, without any binarization step, as

$$\text{MAE} = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |\text{prediction}(x,y) - \text{ground truth}(x,y)|, \tag{4}$$

where W and H refer to image width and height, respectively.

4.1 *Continuous Prediction Versus Binary Prediction*

This first set of the experiments aims at assessing the effect of a continuous-valued estimation of image saliency, as opposed to producing a directly binarized output. Table 3 presents this comparison for each evaluation measure and each dataset proposed in the adopted benchmark.[1] A cross-dataset average is also provided in the last column, in order to get a global view of the impact of such contribution.

The introduction of continuous-valued estimation is mostly benefiting AUC. This curve is drawn by plotting the FPR–TPR value pairs obtained at each possible binarization threshold and collapses to a single point in the degenerate case of an already-binary image. The standard evaluation procedure provided with Ref. 1 generates an additional trivial solution, corresponding to an all-ones saliency estimation (FPR:1 and TPR:1). For an already-binary image, this results in a straight line between two points, and the trapezoid area underlying such line is missing two large chunks when compared to a continuous-valued evaluation, as shown in Fig. 4, resulting in suboptimal performance.

Conversely, mean absolute error (MAE) is impacted negatively by the transition to a continuous-valued prediction. This measure is essentially a direct comparison between prediction and (binary) ground truth, so there is always going to be some residual difference on "true positive" areas, as our continuous-valued prediction is rarely giving 100% confidence on any pixel. Such differences, however small, accumulate over the whole image and result in worse performance according to this particular evaluation measure. It should be noted, though, that even this higher MAE value is still lower (and thus better) than those of any competing method presented in Table 5 from Sec. 4.3.

Finally, the effect on F_β is inconsistent, but on average slightly better than what can be obtained with a binary output. It can be concluded, therefore, that producing a continuous-valued prediction provides a more formally correct setting for the adopted evaluation procedure, while at the same time yielding overall better results across different metrics.

4.2 *Multiscale Prediction Versus Single Scale Prediction*

With the following experiments, we intend verifying the impact of multiscale analysis on the final performance of our method for saliency estimation. In particular, we want to assess the prediction quality that results from rescaling the input image to different specific resolutions, and from combining the predictions from these different inputs into one final saliency map using either per-pixel maximum or per-pixel average.

The first rows in Table 4 show how rescaling the image to one fixed size, as opposed to feeding the original image to the neural network, already brings consistently better performance among all evaluation measures (note that we only

Table 3 Comparison of performance for binary estimation and continuous-valued estimation, on all considered datasets (P, T, J, D, S, M, E). Cross-dataset average is also reported. For all measures, except MAE, a higher value is better.

Measure	Method	P[3]	T[2]	J[4]	D[21]	S[22]	M[19]	E[20]	Average
F_β Varying	Binary[5]	0.763	0.666	0.406	0.706	0.847	0.850	0.864	0.729
	Continuous	0.768	0.722	0.408	0.720	0.850	0.859	0.875	0.743
F_β Adaptive	Binary[5]	0.688	0.620	0.382	0.678	0.857	0.833	0.783	0.692
	Continuous	0.685	0.617	0.380	0.652	0.853	0.834	0.776	0.685
F_β Sal Cut	Binary[5]	0.778	0.702	0.409	0.712	0.791	0.890	0.888	0.739
	Continuous	0.783	0.707	0.404	0.722	0.810	0.909	0.893	0.747
AUC	Binary[5]	0.820	0.851	0.680	0.828	0.844	0.877	0.896	0.828
	Continuous	0.949	0.956	0.807	0.950	0.970	0.971	0.979	0.940
MAE	Binary[5]	0.122	0.106	0.210	0.079	0.080	0.073	0.065	0.105
	Continuous	0.153	0.132	0.272	0.117	0.094	0.110	0.112	0.141

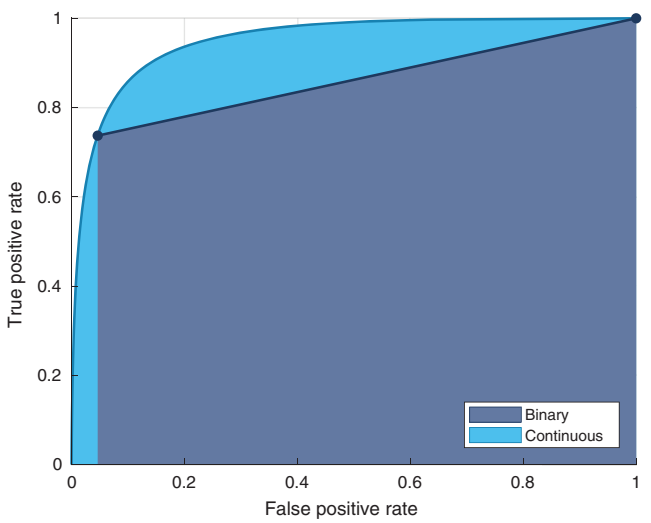

Fig. 4 ROC curves derived from continuous-valued and binary prediction on the DUT-OMRON dataset. The graph visually shows the impact of the two solutions on computing the AUC measure.

report averages across all seven datasets, for reasons of readability). By picking the three most-effective input sizes, i.e., 256×256, 384×384, and 512×512 pixels, we can then try different combinations. The trained model potentially assigns a high saliency score to different regions at different scales, so resizing the predictions to a common resolution, and computing a per-pixel maximum, is going to preserve such high-confidence outputs from all levels of analysis. This merging technique, however, does not produce the expected results, possibly suggesting a different level of relevance for each scale, and possibly requiring a dedicated reasoning for areas where different scales generate highly disagreeing estimations of saliency. Consistently with this last hypothesis, averaging the predictions leads to improved

Table 4 Evaluation results for different input resolutions and combinations (reported values are averages across all datasets). For all measures, except MAE, a higher value is better. The configuration selected for subsequent experiments is highlighted in boldface.

Input size (px)	Merging strategy	F_β varying	F_β adaptive	F_β sal cut	AUC	MAE
Original	—	0.743	0.685	0.747	0.940	0.141
256	—	0.759	0.686	0.771	0.948	0.131
384	—	0.773	0.698	0.772	0.954	0.124
512	—	0.752	0.687	0.754	0.946	0.141
640	—	0.721	0.671	0.731	0.931	0.168
768	—	0.688	0.649	0.710	0.911	0.195
256, 384	Max	0.774	0.677	0.769	0.955	0.136
384, 512	Max	0.767	0.679	0.758	0.953	0.141
256, 512	Max	0.769	0.666	0.759	0.954	0.149
256, 384	**Average**	**0.781**	**0.697**	**0.776**	**0.957**	**0.128**
384, 512	Average	0.773	0.697	0.769	0.955	0.133
256, 512	Average	0.781	0.697	0.774	0.958	0.137

performance under all criteria. Also note that linear combination of independent outputs was shown to be an effective way of model stacking in the past.[29] Overall, the best combination consists of averaging the prediction results from 256-pixel-side images and 384-pixel-side images. This configuration is used in a comparison with other methods for saliency estimation in Sec. 4.3.

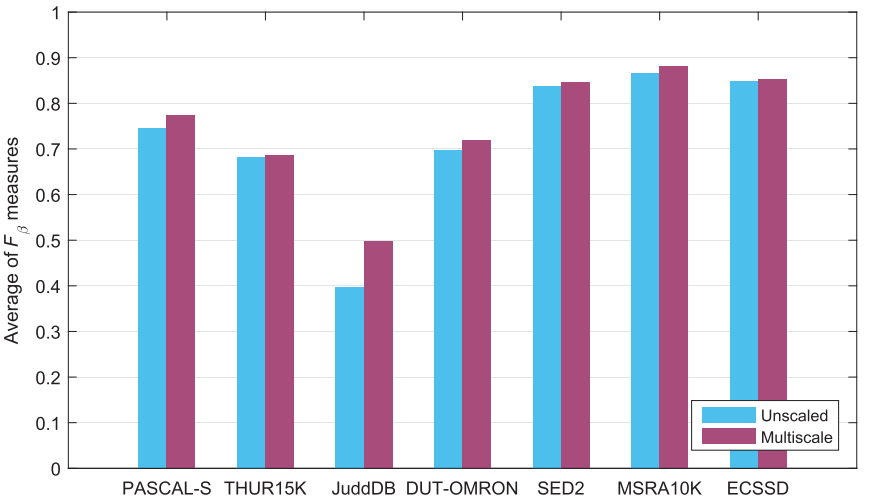

Fig. 5 Average of the three F_β variants on different datasets, for unscaled input and multiscale approach. The impact of multiscale analysis is most prominent on JuddDB dataset.

Figure 5 shows the impact of our multiscale analysis on all seven datasets used for evaluation, reporting the average of the three F_β variants as a reference measure. We can see how dataset JuddDB[4] is the most impacted by this strategy, mainly due to its images being much larger than those of other datasets: the depicted subjects at native resolution, in fact, have a very different size with respect to the training examples. PASCAL-S[3] and DUT-OMRON[21] are also positively affected by multiscale analysis to a meaningful, yet lower, extent.

Figure 6 presents three visual examples of the advantage in applying multiscale analysis. On these images, in fact, a single-scale saliency estimation would generate "holes" in the prediction due to a limited receptive field, whereas our strategy allows the model to consider the whole image. It can be observed, though, that the overall better estimation comes at the price of coarser predictions. This suggests a direction for future developments, where the fine-grained results from analysis at high resolutions might be exploited to provide more accurate details.

4.3 *Comparison with the State-of-the-Art*

We compare our final proposal, characterized by continuous-valued prediction and multiscale analysis at 256×256 and 384×384 pixels combined with per-pixel average, with the

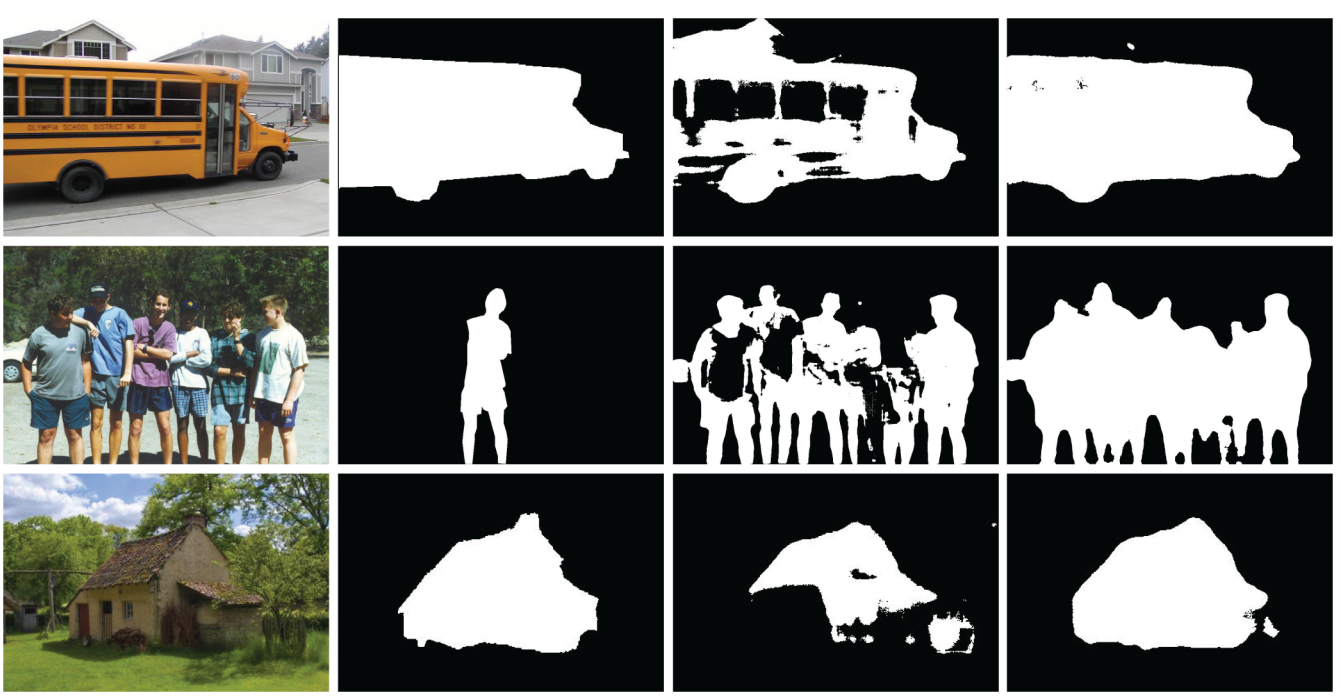

Fig. 6 Effect of multiscale analysis on example images: (a) input image, (b) ground truth annotation, (c) single-scale saliency estimation, and (d) multiscale saliency estimation. The content of columns (c) and (d) is here binarized to facilitate the comparison.

Table 5 Evaluation results for different measures on all datasets.

Measure	Method	P[3]	T[2]	J[4]	D[21]	S[22]	M[19]	E[20]	Average
F_β varying	Ours	**0.811**	**0.726**	**0.536**	**0.744**	**0.877**	0.880	**0.890**	**0.781**
	DRFI[6]	0.679	0.670	0.475	0.665	0.831	**0.881**	0.787	0.713
	QCUT[9]	0.695	0.651	0.509	0.683	0.810	0.874	0.779	0.714
	MBD[7]	N/A	0.622	0.472	0.624	0.799	0.849	0.739	0.684
	ST[8]	0.660	0.631	0.455	0.631	0.818	0.868	0.752	0.688
	RBD[10]	0.652	0.596	0.457	0.630	0.837	0.856	0.718	0.678
F_β adaptive	Ours	**0.697**	0.620	0.424	**0.665**	**0.847**	**0.845**	**0.778**	**0.697**
	DRFI[6]	0.615	0.607	0.419	0.605	0.839	0.838	0.733	0.665
	QCUT[9]	0.654	**0.625**	**0.454**	0.647	0.801	0.843	0.738	0.680
	MBD[7]	N/A	0.594	0.422	0.592	0.803	0.830	0.703	0.657
	ST[8]	0.601	0.580	0.394	0.577	0.805	0.825	0.690	0.639
	RBD[10]	0.607	0.566	0.403	0.580	0.825	0.821	0.680	0.640
F_β sal cut	Ours	**0.812**	**0.713**	**0.530**	**0.751**	**0.813**	**0.918**	**0.894**	**0.776**
	DRFI[6]	0.690	0.674	0.447	0.669	0.702	0.905	0.801	0.698
	QCUT[9]	0.613	0.620	0.480	0.647	0.672	0.843	0.747	0.660
	MBD[7]	N/A	0.642	0.470	0.636	0.759	0.890	0.785	0.697
	ST[8]	0.671	0.648	0.459	0.635	0.768	0.896	0.777	0.693
	RBD[10]	0.667	0.618	0.461	0.647	0.750	0.884	0.757	0.683
AUC	Ours	**0.967**	**0.955**	**0.880**	**0.958**	**0.977**	**0.979**	**0.983**	**0.957**
	DRFI[6]	0.897	0.938	0.851	0.933	0.944	0.978	0.944	0.926
	QCUT[9]	0.870	0.907	0.831	0.897	0.860	0.956	0.909	0.890
	MBD[7]	N/A	0.915	0.838	0.903	0.922	0.964	0.917	0.910
	ST[8]	0.868	0.911	0.806	0.895	0.922	0.961	0.914	0.897
	RBD[10]	0.867	0.887	0.826	0.894	0.899	0.955	0.894	0.889
MAE	Ours	**0.135**	0.133	**0.205**	**0.117**	**0.092**	**0.103**	**0.113**	**0.128**
	DRFI[6]	0.221	0.150	0.213	0.155	0.130	0.118	0.166	0.165
	QCUT[9]	0.195	**0.128**	0.178	0.119	0.148	0.118	0.171	0.151
	MBD[7]	N/A	0.162	0.225	0.168	0.137	0.107	0.172	0.162
	ST[8]	0.224	0.179	0.240	0.182	0.145	0.122	0.193	0.184
	RBD[10]	0.199	0.150	0.212	0.144	0.130	0.108	0.173	0.159

Note: The best result for each dataset-measure combination is highlighted in boldface.

five best-performing methods from the adopted benchmark.[1] Table 5 reports this comparison for all five evaluation measures on the seven datasets, trained in an LODO configuration in order to provide a more fair comparison with methods that do not require an explicit training phase. According to cross-dataset average results, our proposed continuous-valued multiscale saliency estimation performs better than all compared methods under every evaluation measure. By considering the detailed per-dataset performance, we observe how it is only occasionally surpassed by QCUT[9] on specific datasets and metrics combinations, possibly due to the optimization nature of such algorithm. Finally, it can be noted how, for all measures, the performance of our solution is consistently lower on JuddDB[4] when compared to other datasets. This phenomenon affects all analyzed methods, and it is probably due to the peculiar annotation criteria adopted by the dataset curators, so different from those of other collections. As previously noted, in fact, JuddDB typically presents images with multiple subjects, among which only one is labeled as being salient, with information gathered from eye gazes of multiple human observers. Despite this challenging setup, we are still able to create a model that is general enough to outperform all other competing methods, due to a data-driven approach that combines multiscale analysis with a continuous-valued prediction.

Although direct comparison is only possible with methods adhering to the adopted benchmark, we also present results from other data-driven methods under similar (though not identical) evaluation settings. Reference 13 (MC) reports 0.721 F_β and 0.147 MAE on the PASCAL-S dataset, both inferior than our 0.773 average F_β and 0.135 MAE. Reference 12 (RFCN) scores 0.989 F_β on ECSSD and 0.827 F_β on PASCAL-S, compared to our 0.854 average F_β on ECSSD. Although the numbers are not directly comparable, as these methods are trained on different datasets and the reported F_β is the best one obtained with different binarization thresholds, it is interesting to observe how we can obtain similar results with a simpler model, as we do not require any recurrent connection in the image processing. Finally, authors of Ref. 14 (DSS) seem to outperform our solution on both ECSSD and PASCAL-S datasets. Their method produces much sharper saliency estimations, thus further supporting our hypothesis for a possible direction of future improvement.

5 Conclusions

In this paper, we have addressed the problem of image saliency estimation. The task is inherently challenging, as no global agreement exists on what makes an object, in a digital picture, salient. A proof of this is seen in the high heterogeneity among criteria used to annotate public datasets. These reasons led us to propose a data-driven model, with the intent of creating a general concept of "saliency" by observing such large collections of diversely annotated data. The method we presented analyzes the input image at different resolutions to produce a continuous-valued probability map, describing the likelihood of each pixel being salient. This strategy is experimentally shown to be a valid approach to the problem: each of our contributions is rigorously tested on a standard benchmark for salient object detection, consisting of seven datasets and five evaluation measures. Our final proposal presents very good performance in comparison with state-of-the-art methods, demonstrating the value of the proposed solution.

A possible direction for future improvements was found in the excessive coarseness of our saliency estimation. To this extent, we might exploit the already available predictions at high resolutions to create finer details in the final saliency map.

Acknowledgments

We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

1. A. Borji et al., "Salient object detection: a benchmark," *IEEE Trans. Image Process.* **24**(12), 5706–5722 (2015).
2. M.-M. Cheng et al., "Salientshape: group saliency in image collections," *Visual Comput.* **30**(4), 443–453 (2014).
3. Y. Li et al., "The secrets of salient object segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 280–287 (2014).
4. A. Borji, "What is a salient object? A dataset and a baseline model for salient object detection," *IEEE Trans. Image Process.* **24**(2), 742–756 (2015).
5. S. Bianco, M. Buzzelli, and R. Schettini, *A Fully Convolutional Network for Salient Object Detection*, pp. 82–92, Springer International Publishing, Cham (2017).
6. H. Jiang et al., "Salient object detection: a discriminative regional feature integration approach," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2083–2090 (2013).
7. J. Zhang et al., "Minimum barrier salient object detection at 80 fps," in *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 1404–1412 (2015).
8. Z. Liu, W. Zou, and O. Le Meur, "Saliency tree: a novel saliency detection framework," *IEEE Trans. Image Process.* **23**(5), 1937–1952 (2014).
9. C. Aytekin, S. Kiranyaz, and M. Gabbouj, "Automatic object segmentation by quantum cuts," in *22nd Int. Conf. on Pattern Recognition (ICPR)*, pp. 112–117, IEEE (2014).
10. W. Zhu et al., "Saliency optimization from robust background detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2814–2821 (2014).
11. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3431–3440 (2015).
12. L. Wang et al., "Saliency detection with recurrent fully convolutional networks," in *European Conf. on Computer Vision*, pp. 825–841, Springer (2016).
13. R. Zhao et al., "Saliency detection by multi-context deep learning," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1265–1274 (2015).
14. Q. Hou et al., "Deeply supervised salient object detection with short connections," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 5300–5309, IEEE (2017).
15. S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 1395–1403 (2015).
16. M.-M. Cheng, "Salient object detection: a benchmark," 2014, http://mmcheng.net/salobjbenchmark/ (2 May 2018).
17. O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision* **115**(3), 211–252 (2015).
18. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR 2015 Conf.*, https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:main.html (7 May 2015).
19. T. Liu et al., "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(2), 353–367 (2011).
20. Q. Yan et al., "Hierarchical saliency detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1155–1162 (2013).
21. C. Yang et al., "Saliency detection via graph-based manifold ranking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3166–3173 (2013).
22. S. Alpert et al., "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(2), 315–327 (2012).
23. E. H. Adelson et al., "Pyramid methods in image processing," *RCA Eng.* **29**(6), 33–41 (1984).
24. S. Bianco et al., "Deep learning for logo recognition," *Neurocomputing* **245**, 23–30 (2017).
25. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).

26. R. Achanta et al., "Frequency-tuned salient region detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2009)*, pp. 1597–1604, IEEE (2009).
27. M.-M. Cheng et al., "Global contrast based salient region detection," *IEEE Trans. on Pattern Anal. Mach. Intell.* **37**(3), 569–582 (2015).
28. C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: interactive foreground extraction using iterated graph cuts," *ACM Trans. Graphics* **23**(3), 309–314 (2004).
29. J. Sill et al., "Feature-weighted linear stacking," 2009, https://arxiv.org/abs/0911.0460 (2 May 2018).

Simone Bianco received his PhD in computer science at DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione) of the University of Milano-Bicocca, Italy, in 2010. He received his BSc and MSc degrees in mathematics from the University of Milano-Bicocca, Italy, in 2003 and 2006, respectively. He is currently an assistant professor and his research interests include computer vision, machine learning, optimization algorithms, and color imaging.

Marco Buzzelli received his bachelor's degree and master's degree in computer science from the University of Milano-Bicocca, Italy, respectively, in 2012 and 2014, focusing on image processing and computer vision tasks. He is currently a PhD student in computer science. His main topics of research include characterization of digital imaging devices and image understanding in complex scenes.

Raimondo Schettini is a professor at the University of Milano-Bicocca, Italy. He is a vice director of the Department of Computer Science and head of the Imaging and Vision Laboratory. He has been associated with the Italian National Research Council since 1987, leading the color imaging lab from 1990 to 2002. He has been team leader in several research projects, published more than 300 refereed papers and six patents. He is a fellow of the International Association of Pattern Recognition.