

# A Deep Multi-Level Network for Saliency Prediction

Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra and Rita Cucchiara

Dipartimento di Ingegneria “Enzo Ferrari”

Università degli Studi di Modena e Reggio Emilia

Email: {name.surname}@unimore.it

**Abstract**—This paper presents a novel deep architecture for saliency prediction. Current state of the art models for saliency prediction employ Fully Convolutional networks that perform a non-linear combination of features extracted from the last convolutional layer to predict saliency maps. We propose an architecture which, instead, combines features extracted at different levels of a Convolutional Neural Network (CNN). Our model is composed of three main blocks: a feature extraction CNN, a feature encoding network, that weights low and high level feature maps, and a prior learning network. We compare our solution with state of the art saliency models on two public benchmarks datasets. Results show that our model outperforms under all evaluation metrics on the SALICON dataset, which is currently the largest public dataset for saliency prediction, and achieves competitive results on the MIT300 benchmark. Code is available at <https://github.com/marcellacornia/mlnet>.

## I. INTRODUCTION

When human observers look at an image, effective attentional mechanisms attract their gazes on salient regions which have distinctive variations in visual stimuli. Emulating such ability has been studied for more than 80 years by neuroscientists [1] and more recently by computer vision researches [2].

Traditionally, algorithms for saliency prediction focused on identifying the fixation points that human viewers would focus on at first glance. Others have concentrated on highlighting the most important object regions in an image [3], [4], [5]. We focus on the first type of saliency models, that try to predict eye fixations over an image.

Inspired by biological studies, researchers have defined hand-crafted and multi-scale features that capture a large spectrum of stimuli: lower-level features (color, texture, contrast) [6] or higher-level concepts (faces, people, text, horizon) [7]. However, given the large variety of aspects that can contribute to define visual saliency, it is difficult to design approaches that combine all these hand-tuned factors in an appropriate way.

In recent years, Deep learning techniques have shown impressive results in several vision tasks, such as image classification [8] and semantic segmentation [9]. Motivated by these achievements, first attempts to predict saliency map with deep convolutional networks have been performed [10], [11]. These solutions suffered from the small amount of training data compared to the ones available in other contexts requiring the usage of limited number of layers or the usage of pretrained

architectures generated for other tasks. The recent publication of the large dataset SALICON [12], collected thanks to crowd-sourcing techniques, allows researches to increase the number of convolutional layers reducing the overfitting risk [13], [14].

Moreover, it is well known that when observers view complex scenes presented on computer monitors, there is a strong tendency to look more frequently around the center of the scene than around the periphery [15]. This has been exploited in past works on saliency prediction, by incorporating hand-crafted priors into saliency maps, or by learning the relative contribution of different priors.

In this paper we present a deep learning architecture for predicting saliency maps, which exploits multi-level features extracted from a CNN, while still being trainable end-to-end. In contrast to the current trend, we let the network learn its own prior from training data. A new loss function is also designed to train the proposed network and to tackle the imbalance problem of saliency maps.

## II. RELATED WORK

In the last decade saliency prediction has been widely studied. The seminal works by Koch and Ullman [16] and Itti *et al.* [2] introduced a biologically-plausible architecture for saliency detection that extracts multi-scale image features based on color, intensity and orientation. Later, Hou and Zange [6] proposed a method that analyzes the log spectrum of each image and obtains the spectral residual, which allows the estimation of the behavior of pre-attentive visual search. Torralba *et al.* [17] show how global contextual information

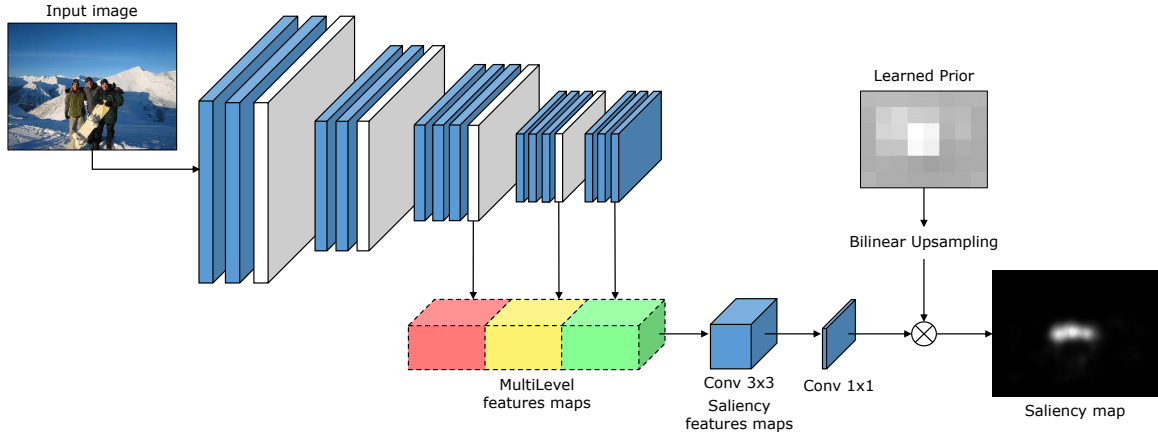


Fig. 1. Overview of our model (see Section III for details). A CNN is used to compute low and high level features from the input image. Extracted features maps are then fed to an Encoding network, which learns a feature weighting function to generate saliency-specific feature maps. A prior image is also learned and applied to the predicted saliency map.

a CNN with three layers. Since the amount of data available to learn saliency is generally limited, this architecture cannot scale to outperform the current state-of-the art.

To address this issue, Kümmerer *et al.* [11] present a way of reusing existing neural networks, trained for image classification, to predict fixation maps. In particular, they present Deep Gaze, a neural network that uses the well-known AlexNet [8] architecture to generate a high dimensional feature space, which is used to create a saliency map. Similarly, Huang *et al.* [12] propose an architecture that integrates saliency prediction into DCNs pretrained for object recognition (AlexNet [8], VGG-16 [20] and GoogLeNet [21]). The key component is a fine-tuning of DNNs weights with an objective function based on the saliency evaluation metrics, such as Normalized Scanpath Saliency (NSS), Similarity and KL-Divergence.

Liu *et al.* [22] propose a multi-resolution CNN that is trained from image regions centered on fixation and non-fixation locations at multi-scales. Recently, Srinivas *et al.* [13] presented DeepFix, in which they introduce Location Biased Convolution filters that allow the network to exploit location dependent patterns. Pan *et al.* [14] present two different architectures: a shallow convnet trained from scratch and a deep convnet that uses parameters previous learned on the ILSVRC-12 dataset.

What the majority of these approaches share is the use of Fully Convolutional Networks which are trained to predict saliency map from a non-linear combination of high level features, extracted from the last convolutional layer. Our architecture learns how to weight features coming from different levels of a CNN, and demonstrates the effectiveness of using medium level features.

### III. PROPOSED APPROACH

Our saliency model is composed by three main parts: given an input image, a CNN extracts low, medium and high level features; then, an encoding network builds saliency-specific

features and produces a temporary saliency map. A prior is then learned and applied to produce the final saliency prediction. Figure 1 reports a summary of the architecture.

**Feature extraction network** The first component of our architecture is a Fully Convolutional network with 13 layers, which takes the input image and produces features maps for the encoding network.

We build our architecture on the popular 16 layers model from VGG [20], which is well known for its elegance and simplicity, and at the same time yields nearly state of the art results in image classification and good generalization properties. However, like any standard CNN architecture, it has the significant disadvantage of reducing the size of feature maps at higher levels with respect to the input size. This is due to the presence of spatial pooling layers which have a stride greater than one: the output of each layer is a three-dimensional tensor with shape  $k \times \lfloor \frac{H}{f} \rfloor \times \lfloor \frac{W}{f} \rfloor$ , where  $k$  is the number of filters of the layer, and  $f$  is the downsampling factor of that stage of the network. In the VGG-16 model there are five max-pooling stages with kernel size  $k = 2$  and stride 2. Given an input image with size  $W \times H$ , the output feature map has size  $\lfloor \frac{W}{2^5} \rfloor \times \lfloor \frac{H}{2^5} \rf$

spectively, conv3, conv4 and conv5, since they come from the third, fourth and fifth convolutional stage of the network. They all share the same spatial size, and are concatenated to form a tensor with 1280 channels, which is fed to a Dropout layer with retain probability 0.5, to improve generalization. A convolutional layer then learns 64 saliency-specific feature maps with a  $3 \times 3$  kernel. A final  $1 \times 1$  convolution learns to weight the importance of each saliency feature map to produce the final predicted feature map.

**Prior learning** Instead of using pre-defined priors as done in the past, we let the network learn its own custom prior. To this end, we learn a coarse  $w' \times h'$  mask (with  $w' \ll w$  and  $h' \ll h$ ), initialized to one, upsample and apply it to the predicted saliency map with pixel-wise multiplication.

Given the learned prior  $U$  with shape  $w' \times h'$ , we interpolate the pixels of  $U$  to produce an output prior map  $V$  of size  $w \times h$ . We compute a sampling grid  $G$  of shape  $w' \times h'$  associating each element of  $U$  with real-valued coordinates into  $V$ . If  $G_{i,j} = (x_{i,j}, y_{i,j})$  then  $U_{i,j}$  should be equal to  $V$  at  $(x_{i,j}, y_{i,j})$ ; however since  $(x_{i,j}, y_{i,j})$  are real-valued, we convolve with a sampling kernel and set

$$V_{x,y} = \sum_{i=1}^{w'} \sum_{j=1}^{h'} U_{i,j} k_x(x - x_{i,j}) k_y(y - y_{i,j}) \quad (1)$$

where  $k_x(\cdot)$  and  $k_y(\cdot)$  are bilinear kernels, corresponding to  $k_x(d) = \max(0, \frac{w}{w'} - |d|)$  and  $k_y(d) = \max(0, \frac{h}{h'} - |d|)$ .  $w'$  and  $h'$  were set to  $\lfloor w/10 \rfloor$  and  $\lfloor h/10 \rfloor$  in all our tests.

**Training** At training time, we randomly sample a minibatch containing  $N$  training saliency maps (in our experiments  $N = 10$ ), and encourage the network to minimize a loss function through Stochastic Gradient Descent. Our loss function is inspired by three objectives: predictions should be pixelwise similar to ground truth maps, therefore a square error loss  $\|\phi(\mathbf{x}_i) - \mathbf{y}\|^2$  is a reasonable choice. Secondly, predicted maps should be invariant to their maximum, and there is no point in forcing the network to produce values in a given numerical range, so predictions are normalized by their maximum. Third, the loss should give the same importance to high and low ground truth values, even though the majority of ground truth pixels are close to zero. For this reason, the deviation between predicted values and ground-truth values  $\mathbf{y}_i$  is weighted by a linear function  $\alpha - \mathbf{y}_i$ , which tends to give more importance to pixels with high ground-truth fixation probability.

The overall loss function is thus

$$L(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N \left\| \frac{\frac{\phi(\mathbf{x}_i)}{\max \phi(\mathbf{x}_i)} - \mathbf{y}_i}{\alpha - \mathbf{y}_i} \right\|^2 + \lambda \|\mathbf{1} - U\|^2 \quad (2)$$

where a  $L_2$  regularization term

TABLE I  
COMPARISON RESULTS ON THE SALICON TEST SET [24].

	CC	AUC shuffled	AUC Judd
<b>Our method</b>	<b>0.7430</b>	<b>0.7680</b>	<b>0.8660</b>
Deep Convnet - Pan <i>et al.</i> [14]	0.6220	0.7240	0.8580
Shallow Convnet - Pan <i>et al.</i> [14]	0.5957	0.6698	0.8364
WHU IIP	0.4569	0.6064	0.7923
Rare 2012 Improved [25]	0.5108	0.6644	0.8148
Xidian	0.4811	0.6809	0.8051
Baseline: BMS [26]	0.4268	0.6935	0.7899
Baseline: GBVS [27]	0.4212	0.6303	0.7899
Baseline: Itti [2]	0.2046	0.6101	0.6669

fixation points, while other directly compare the predicted saliency map with fixation points [31].

The Similarity metric [28] computes the sum of pixel-wise minimums between the predicted saliency map  $SM$  and the human eye fixation map  $FM$ , where  $SM$  and  $FM$  are supposed to be probability distributions and sum up to one. A similarity score of one indicates that the predicted map is identical to the ground truth one.

The linear correlation coefficient (CC), instead is the Pearson's linear coefficient between  $SM$  and  $FM$ . It ranges between  $-1$  and  $1$ , and a score close to  $-1$  or  $1$  indicates a perfect linear relationship between the two maps.

Earth Mover's Distance (EMD) represents the minimal cost to transform the probability distribution of the saliency map  $SM$  into the one of the human eye fixations  $FM$ . Therefore, a larger EMD indicates a larger difference between the two maps.

The Normalized Scanpath Saliency (NSS) metric was defined specifically for the evaluation of saliency models [32]. The idea is to quantify the saliency map values at the eye fixation locations and to normalize it with the saliency map variance

$$NSS(p) = \frac{SM(p) - \mu_{SM}}{\sigma_{SM}} \quad (3)$$

where  $p$  is the location of one fixation and  $SM$  is the saliency map which is normalized to have a zero mean and unit standard deviation. The final NSS score is the average of  $NSS(p)$  for all fixations.

Finally, the Area Under the ROC curve (AUC) is one of the most widely used metrics for the evaluation of maps predicted from saliency models. There are several different implementations of this metric. In our experiments we use AUC Judd, AUC Borji and shuffled AUC. The AUC

TABLE II  
COMPARISON RESULTS ON THE MIT300 TEST SET [28].

	Similarity	CC	AUC shuffled	AUC Borji	AUC Judd	NSS	EMD
Infinite humans	1.00	1.00	0.80	0.87	0.91	3.18	0.00
DeepFix [13]	<b>0.67</b>	<b>0.78</b>	0.71	0.80	<b>0.87</b>	<b>2.26</b>	<b>2.04</b>
SALICON [12]	0.60	0.74	0.74	0.85	<b>0.87</b>	2.12	2.62
<b>Our method</b>	0.59	0.67	0.70	0.75	0.85	2.05	2.63
Pan <i>et al.</i> - Deep Convnet [14]	0.52	0.58	0.69	0.82	0.83	1.51	3.31
BMS [26]	0.51	0.55	0.65	0.82	0.83	1.41	3.35
Deep Gaze 2 [11]	0.46	0.51	<b>0.76</b>	<b>0.86</b>	<b>0.87</b>	1.29	4.00
Mr-CNN [22]	0.48	0.48	0.69	0.75	0.79	1.37	3.71
Pan <i>et al.</i> - Shallow Convnet [14]	0.46	0.53	0.64	0.78	0.80	1.47	3.99
GBVS [27]	0.48	0.48	0.63	0.80	0.81	1.24	3.51
Rare 2012 Improved [25]	0.46	0.42	0.67	0.75	0.77	1.34	3.74
Judd [19]	0.42	0.47	0.60	0.80	0.81	1.18	4.45
eDN [10]	0.41	0.45	0.62	0.81	0.82	1.14	4.56

test image  $j$  and for every saliency pixel  $i$ . To reduce the amount of required computation, instead of computing the gradient of each saliency pixel, we compute the gradient of the mean and variance of the output saliency map, in the neighborhood of each test sample. Applying Eq. 6, we get an indication of the contribution of each feature pixel to the mean and variance of the predicted map.

Figure 3 reports the

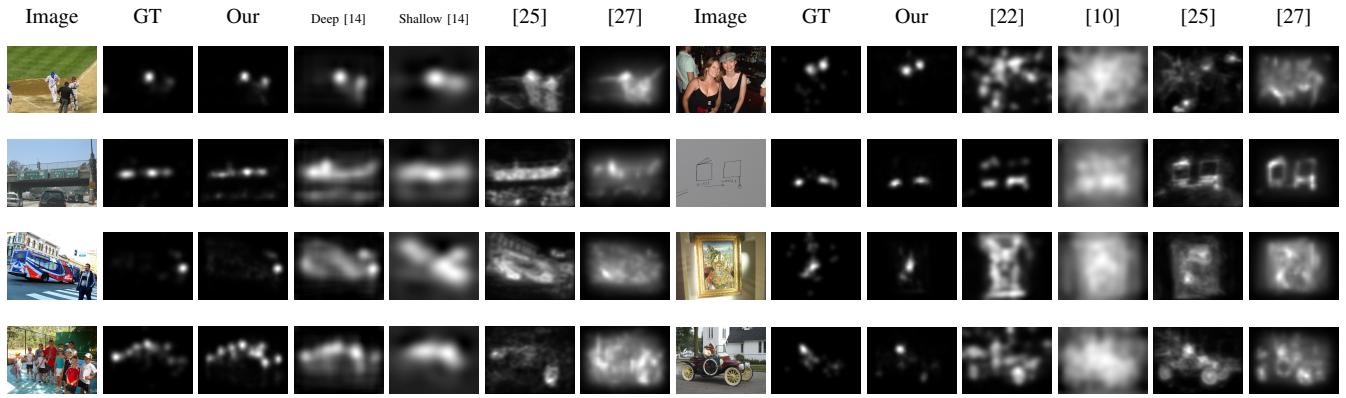


Fig. 4. Qualitative results and comparison to the state of the art. Left: validation images from SALICON dataset [24]. Right: validation images from MIT1003 dataset [19]. Best viewed in color.

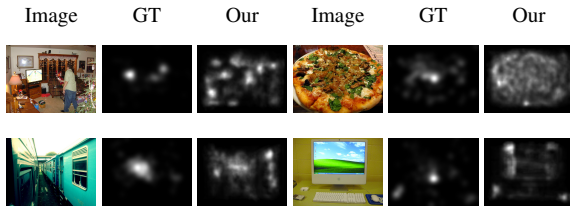


Fig. 5. Example of failure cases on validation images from SALICON dataset [24]. Best viewed in color.