Multi-Level Net: a Visual Saliency Prediction Model

Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, Rita Cucchiara

Department of Engineering "Enzo Ferrari",
University of Modena and Reggio Emilia
{name.surname}@unimore.it

Abstract. State of the art approaches for saliency prediction are based on Fully Convolutional Networks, in which saliency maps are built using the last layer. In contrast, we here present a novel model that predicts saliency maps exploiting a non-linear combination of features coming from different layers of the network. We also present a new loss function to deal with the imbalance issue on saliency masks. Extensive results on three public datasets demonstrate the robustness of our solution. Our model outperforms the state of the art on SALICON, which is the largest and unconstrained dataset available, and obtains competitive results on MIT300 and CAT2000 benchmarks.

Keywords: Visual Saliency, Saliency Prediction, Convolutional Neural Network, Deep Learning

1 Introduction

For many applications in image and video compression, video re-targeting and object segmentation, estimating where humans look in a scene is an essential step [9,22,6]. Neuroscientists [2], and more recently computer vision researches [13], have proposed computational saliency models to predict eye fixations over images.

Most traditional approaches typically cope with this task by defining hand-crafted and multi-scale features that capture a large spectrum of stimuli: lower-level features (color, texture, contrast) [11] or higher-level concepts (faces, people, text, horizon) [5]. In addition, since there is a strong tendency to look more frequently around the center of the scene than around the periphery [33], some techniques incorporate hand-crafted priors into saliency maps [36,35,20,19]. Unfortunately, eye fixation can depend on several aspects and this makes it difficult to design properly hand-crafted features.

Deep learning techniques, with their ability to automatically learn appropriate features from massive annotated data, have shown impressive results in several vision applications such as image classification [18] and semantic segmentation [24]. First attempts to define saliency models with the usage of deep convolutional networks have recently been presented [35,20]. However, due to the small amount of training data in this scenario, researchers have presented

networks with few layers or pretrained in other contexts. By publishing the large dataset SALICON [12], collected thanks to crowd-sourcing techniques, researches have then increased the number of convolutional layers reducing the overfitting risk [19,25].

In this paper we present a general deep learning framework to predict saliency maps, called ML-Net. Differently from the previous deep learning approaches, that build saliency images based on the last convolutional layer, we propose a network that is able to combine multiple features coming from different layers of the network. The proposed solution is also able to learn its own prior from the training data, avoiding an hand-crafted definition. Finally, a new loss function is presented to tackle the imbalance problem of saliency maps, in which salient pixels are usually a minor percentage. Experimental results on three public datasets validate our solution.

2 Related works

Early works on saliency detection were concerned with defining biologically-plausible architecture inspired by the human visual attention system. Koch, Ullman [17] and Itti *et al.* [13] were among the earliest ones. In particular, they proposed to extract multi-scale image features based on color, intensity and orientation, mimicking the properties of primate early vision. More recently, Hou and Zange [11] presented a technique based on log spectral representations of images, which extracted the spectral residual of an image, thus simulating the behavior of pre-attentive visual search. Differently, Torralba *et al.* [34] showed how the human visual system makes extensive use of contextual information in natural scenes. Similarly, Goferman *et al.* [8] proposed an approach that detects salient regions which are distinctive with respect to both their local and global surroundings. Judd *et al.* [16] and Cerf *et al.* [5] presented two techniques based on the combination of low-level features (color, orientation and intensity) and high-level semantic information (i.e. the location of faces, cars and text) and showed that this strategy significantly improves saliency prediction. However, all these methods employed hand-tuned features or trained specific higher-level classifiers.

Recently, Deep Convolutional Networks (DCNs) were used by several authors and appear much more appropriate to support saliency detection. Indeed, DCNs have been proved to be able to build descriptive features. Vig *et al.* [35] presented Ensembles of Deep Networks (eDN), a convolutional neural network with three layers. Since the annotated data available at that time to learn saliency was limited, their architecture could not outperform the current state-of-the art. To overcome this problem, Kümmerer *et al.* [20] suggest to reuse existing neural networks trained for object recognition and propose Deep Gaze, a neural network based on the AlexNet [18] architecture. Similarly, Huang *et al.* [12] present a DCN architecture for saliceny prediction that combines multiple DCNs pretrained for object recognition (AlexNet [18], VGG-16 [30] and GoogLeNet [32]). The fine-tuning procedure of this architecture is performed using an objective

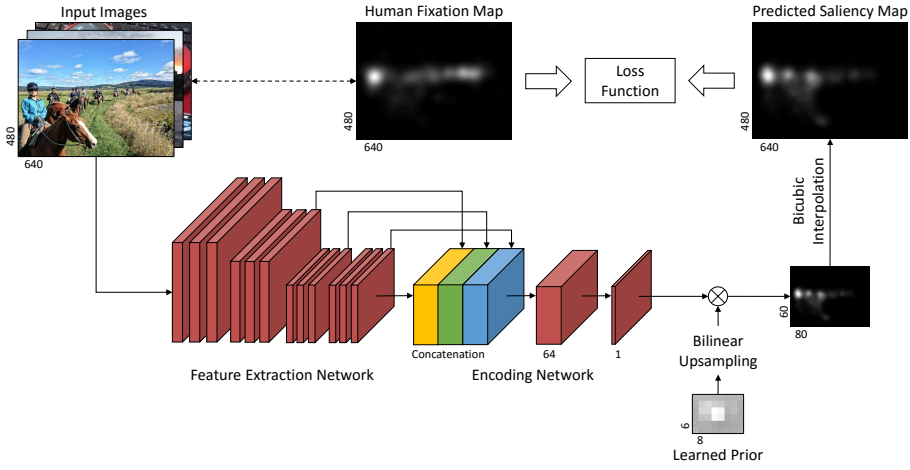

Fig. 1. Architecture of ML-Net.

function based on saliency evaluation metrics, such as the Normalized Scanpath Saliency, Similarity and KL-Divergence.

Liu *et al.* [23] present a multi-resolution Convolutional Neural Network which is trained from image regions centered on fixation and non-fixation locations over multiple resolutions. Srinivas *et al.* [19] propose a network, called DeepFix, that includes Location Biased Convolution filters able to identify location dependent patterns. Pan *et al.* [25] show how two different architectures, a shallow convent trained from scratch and a deep convent that uses parameters previous learned on the ILSVRC-12 dataset [29], can achieve state of the art results.

3 Our Approach

We argue that saliency prediction can benefit from both low level and high level features. For this reason, we build a saliency prediction model which combines features extracted at multiple levels from a Fully Convolutional Neural Network (FCN). Since the role of this network in our model is that of extracting features, instead of predicting a saliency map, we call this component *Feature extraction network*. An *Encoding network* is then designed to weight and combine feature maps extracted from the FCN, and training is performed by means of a loss function which tackles the problem of imbalance on saliency maps. An overview of our architecture, which we call ML-Net, is presented in Fig. 1.

3.1 Feature extraction network

Current Fully Convolutional models can be described as sequences of convolutional and max-pooling layers, which process an input tensor to produce activation maps. Due to the presence of spatial pooling layers, convolutional layers with

Table 1. Output size of each layer of the FCN models used in our architecture. First column is the model inspired by VGG-16, second column is the one inspired by VGG-19 and the last one is inspired by AlexNet.

Input	$3 \times 480 \times 640$
conv1-1	$64 \times 480 \times 640$
conv1-2	$64 \times 480 \times 640$
maxpool1	$64 \times 240 \times 320$
conv2-1	$128 \times 240 \times 320$
conv2-2	$128 \times 240 \times 320$
maxpool2	$128 \times 120 \times 160$
conv3-1	$256 \times 120 \times 160$
conv3-2	$256 \times 120 \times 160$
conv3-3	$256 \times 120 \times 160$
maxpool3	$256 \times 60 \times 80$
conv4-1	$512 \times 60 \times 80$
conv4-2	$512 \times 60 \times 80$
conv4-3	$512 \times 60 \times 80$
maxpool4	$512 \times 60 \times 80$
conv5-1	$512 \times 60 \times 80$
conv5-2	$512 \times 60 \times 80$
conv5-3	$512 \times 60 \times 80$

Input	$3 \times 480 \times 640$
conv1-1	$64 \times 480 \times 640$
conv1-2	$64 \times 480 \times 640$
maxpool1	$64 \times 240 \times 320$
conv2-1	$128 \times 240 \times 320$
conv2-2	$128 \times 240 \times 320$
maxpool2	$128 \times 120 \times 160$
conv3-1	$256 \times 120 \times 160$
conv3-2	$256 \times 120 \times 160$
conv3-3	$256 \times 120 \times 160$
conv3-4	$256 \times 120 \times 160$
maxpool3	$256 \times 60 \times 80$
conv4-1	$512 \times 60 \times 80$
conv4-2	$512 \times 60 \times 80$
conv4-3	$512 \times 60 \times 80$
conv4-4	$512 \times 60 \times 80$
maxpool4	$512 \times 60 \times 80$
conv5-1	$512 \times 60 \times 80$
conv5-2	$512 \times 60 \times 80$
conv5-3	$512 \times 60 \times 80$
conv5-4	$512 \times 60 \times 80$

Input	$3 \times 480 \times 640$
conv1	$96 \times 118 \times 158$
maxpool1	$96 \times 58 \times 78$
conv2	$256 \times 58 \times 78$
maxpool2	$256 \times 56 \times 76$
conv3	$384 \times 56 \times 76$
conv4	$384 \times 56 \times 76$
conv5	$256 \times 56 \times 76$

stride greater than one, or border effects, activation maps are usually smaller than input images.

The spatial resolution of an intermediate activation map, with respect to the input of the layer, can be written as $\left(\left\lfloor \frac{H+2p-k}{s} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{W+2p-k}{s} \right\rfloor + 1\right)$, where $H \times W$ is the spatial resolution of the input, s is the stride, p is the padding and k is the kernel size. For instance, the AlexNet model [18] by Krizhevsky *et al.* uses different values of s, p and k across different layers ($s = 4$, $p = 0$ and $k = 11$ in the first convolutional layer, $s = 1, p = 1, k = 3$ for the last convolutional layer), while VGG-16 and VGG-19 models [31] use $s = 1$, $p = 1$ and $k = 3$ for convolutional layers and $s = 2, p = 0, k = 2$ for max-pooling layers.

To combine low level and high level features extracted from a FCN model, one could in principle reduce activation maps to a common spatial resolution, through downsampling or upsampling operations, and then concatenate them to form a single feature tensor. In contrast to this approach, which would imply a loss of information, in the case of downsampling, or a non-exact reconstruction of missing information, in the case of upsampling, we modify the stride of some layers in order to maintain the same spatial resolution across different layers. We apply this technique to three popular CNN models: VGG-16, VGG-19 and AlexNet.

In the case of the VGG-16 model, we set the stride on layer `maxpool4` to one, so to have activation maps from layers `conv5-3`, `maxpool4` and `maxpool3` with the same spatial size. We do the same in the VGG-19 model, again by setting the stride of `maxpool4` to one and considering feature maps from layers `conv5-4`, `maxpool4` and `maxpool3`. Finally, for the AlexNet model, we set the stride of layer `maxpool2` equal to one, to have the output of layers `maxpool1`, `maxpool2` and `conv5` having almost the same spatial support. These activation maps are then zero-padded to bring them to the same spatial resolution. All three models, as well as the output size of each of their layers, are reported in Table 1 for reference.

3.2 Encoding network

Since feature maps extracted from the FCN model have the same spatial resolution, it is reasonable to concatenate them to form a single feature tensor. It is worth mentioning that the resulting tensor encodes features extracted from different levels of a FCN, and thus it is far more informative than the activation tensor coming from the last convolutional layer, which is usually employed to predict fixation maps. Beside containing high level features, like the responses to object detectors and part of object detectors, indeed, it contains responses to middle level features, like textures.

To combine features maps coming from different levels, and in order to form the final saliency map, we build an encoding network, whose aim is to weight low level, middle level and high level features to produce a provisional saliency prediction. The encoding network is composed of two convolutional layers, the first one having kernel size 3×3 and 64 feature maps, and the last one having a 1×1 kernel and a single feature map. Being the two convolutional layers separated by a ReLU activation stage, the provisional prediction can be a non-linear combination of input activation maps.

3.3 Prior learning

The combination of a FCN model with the previously defined encoding network lets the network learn more robust saliency features, thus increasing the accuracy of predicted saliency maps. However, what the encoding network can not deal with is the role of the relative and absolute position of salient areas in the image. Indeed, the center of an image is well known to be more salient than the periphery, and this notion is usually incorporated in saliency models by means of a prior. Instead of using an hand-crafted prior, as done in the past, we let the network learn its own prior.

In particular, we learn a coarse $w' \times h'$ mask, which is upsampled and applied to the predicted saliency map with pixel-wise multiplication. The mask is initialized to one, so that the network can learn a prior by reducing excessive values.

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 $w \times h$, being w and h respectively the

width and height of the predicted saliency map. 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

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

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

3.4 Training

For training, we randomly sample a minibatch containing N training saliency maps, and encourage the network to minimize a loss function through Stochastic Gradient Descent. While the majority of saliency prediction models employ a MSE or a KL-Divergence loss, we build a custom loss function which tackles the problem of imbalance in saliency maps.

Our loss function is motivated by three observations: first of all, predictions should be pixelwise similar to ground truth maps, therefore a square error loss $\|\phi(\mathbf{x}_i) - \mathbf{y}_i\|^2$, between the predicted saliency map $\phi(\mathbf{x}_i)$ and the ground-truth map $\mathbf{y}_i$, is a reasonable starting model. 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 and ground-truth values 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

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

where a L_2 regularization term is added to penalize the deviation of the prior mask U from its initial value, thus encouraging the network to adapt to ground truth maps by changing convolutional weights rather than modifying the prior.

4 Experimental evaluation

4.1 Datasets

For training and evaluation we employ the following datasets: SALICON [14], MIT1003 [16], MIT300 [15] and CAT2000 [1].

SALICON contains 20,000 images taken from the Microsoft CoCo dataset [21] and divided in 10,000 training images, 5,000 validation images and 5,000 testing images. It is currently the largest public dataset available for saliency prediction though its saliency maps were not collected with eye-tracking systems as in classical datasets for saliency prediction. Saliency maps were indeed generated by collecting mouse movements, and authors showed, both qualitatively and quantitatively, an high degree of similarity between their maps and those created from eye-tracking data.

MIT1003 includes 1003 random images taken from Flickr and LabelMe. Its saliency maps were generated using eye-tracking data from fifteen participants. MIT300 contains 300 natural images from both indoor and outdoor scenarios. Despite its limited size, it is the one of the most commonly used datasets for saliency prediction. Its saliency maps, that have been created from eye-tracking data of 39 observers, are not public available. To evaluate the effectiveness of our model on this dataset, we submitted our predictions to the MIT saliency benchmark [3].

CAT2000 is a collection of 4,000 images divided in 20 different categories such as *Cartoons*, *Art*, *Satellite*, *Low resolution images*, *Indoor*, *Outdoor*, *Line drawings*, ect. and each category contains 200 images. Saliency maps of this dataset have been created using eye-tracking data from 24 users. Images are divided in training set and test set where each of them consists of 2,000 images. Saliency maps of the test set are held-out and also in this case we submitted our predictions to the MIT saliency benchmark to evaluate performances of our model.

4.2 Evaluation metrics

Several evaluation metrics have been proposed for saliency predictions: Normalized Scanpath Saliency (NSS), Earth Mover's Distance (EMD), Linear Correlation Coefficient (CC), Similarity, AUC Judd, AUC Borji and AUC shuffled (sAUC). Some of these metrics consider saliency at discrete fixation locations, while others treat both predicted saliency maps and ground truth maps, generated from fixation points, as distributions [4], [27].

The Normalized Scanpath Saliency (NSS) metric was introduced specifically for the evaluation of saliency models [26]. The idea is to quantify the saliency map values at the eye fixation locations and to normalize it whit the saliency map variance

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

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

$$NSS = \frac{1}{N} \sum_{p=1}^{N} NSS(p) \tag{4}$$

where N is the total number of eye fixations.

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 Linear Correlation Coefficient (CC) instead is the Pearson's linear coefficient between SM and FM and is computed as

$$CC = \frac{conv(SM, FM)}{\sigma_{SM} * \sigma_{FM}} \tag{5}$$

It ranges between -1 and 1, and a score close to -1 or 1 indicates a perfect linear relationship between the two maps.

The Similarity metric [15] is computed as the sum of pixel-wise minimums between the predicted saliency map SM and the human eye fixation map FM, after normalizing the two maps

$$S = \sum_{x=1}^{X} min(SM(x), FM(x)) \tag{6}$$

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.

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. The saliency map is treated as a binary classifier of fixations at various threshold values, and a ROC curve can be drawn by measuring the true and false positive rates under each binary classifier. There are several different implementations of this metric which differ in how true and false positives are calculated. In our experiments we use AUC Judd, AUC Borji and shuffled AUC. The AUC Judd and the AUC Borji choose non-fixation points with a uniform distribution, otherwise shuffled AUC uses human fixations of other images in the dataset as non-fixation distribution. In that way, centered distribution of human fixations of the dataset is taken into account.

4.3 Implementation details

Using the three feature extraction networks described in Section 3.1 (inspired by AlexNet, VGG-16 and VGG-19), we build three different variations of our saliency prediction model. Weights of all feature extraction networks are initialized to those of pre-trained models on the ILSVRC-12 dataset [29], while weights of the encoding networks are initialized according to [7], and biases are initialized to zero. SGD is applied with Nesterov momentum 0.9, weight decay 0.0005 and learning rate 10^{-3}. Parameters α and λ are respectively set to 1.1 and $1/(w' \cdot h')$ in all our experiments. Finally, the batch size N is set to 10.

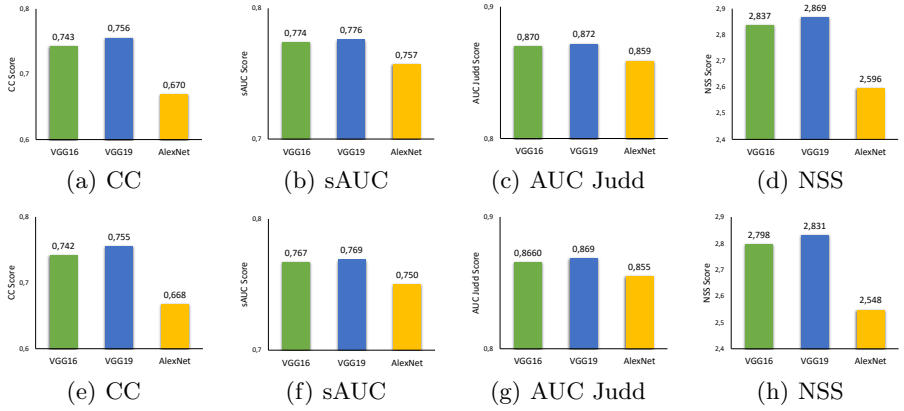

Fig. 2. Comparison between our three ML-Nets on SALICON dataset [14]. Each plot corresponds to a different evaluation metric (i.e. CC, sAUC, AUC Judd and NSS). Plots a-d correspond to the results on SALICON validation set, while plots e-h correspond to the results on SALICON test set.

We evaluate on the SALICON, on the MIT300 and on the CAT2000 datasets. First of all, we train our network on SALICON training set using the 5,000 images of SALICON validation set to validate the model. Secondly, we finetune our architecture on the MIT1003 dataset and on the CAT2000 training set to evaluate our model also on MIT300 dataset and CAT2000 testing set, respectively. In particular, we randomly split images of MIT1003 in 900 training images and 103 validation images and, after the training, we test our model on MIT300. For the CAT2000 instead, we randomly choose 200 images of training set (10 images for each category) as validation and we finetune the network on remaining images. Finally we test our network on the CAT2000 testing set.

Images from all datasets were resized to 640×480. In particular, images of MIT1003 and MIT300 datasets were zero-padded to fit a $4:3$ aspect ratio and then resized to 640×480, while images from CAT2000 dataset were resized and then cropped to have a dimension of 640×480. Predicted saliency maps are upsampled with bicubic interpolation to the original image size before evaluation.

4.4 Quantitative results

To investigate the performance of our solution, we first conduct a series of experiments on the SALICON dataset using the three different feature extraction networks. Fig. 2 reports the results of our architecture when using the three FCN in terms of CC, AUC shuffled, AUC Judd and NSS. VGG-16 and VGG-19 can clearly extract better features than the AlexNet model, and VGG-19 achieves the best performance according to all performances measures.

In Table 2 we then compare the performance of our model on the SALICON test set with respect to the current state of the art, in terms of CC, AUC shuf-

Table 2. Comparison results on the SALICON test set [14].

	CC	sAUC	AUC Judd
ML-Net (VGG-19)	**0.7562**	**0.7782**	**0.8721**
Pan *et al.* [25] - Deep	0.6220	0.7240	0.8580
Pan *et al.* [25] - Shallow	0.5957	0.6698	0.8364
WHU IIP	0.4569	0.6064	0.7923
Rare 2012 Improved [28]	0.5108	0.6644	0.8148
Xidian	0.4811	0.6809	0.8051
Baseline: BMS [37]	0.4268	0.6935	0.7899
Baseline: GBVS [10]	0.4212	0.6303	0.7899
Baseline: Itti [13]	0.2046	0.6101	0.6669

Table 3. Comparison results on the MIT300 dataset [15].

	Sim	CC	sAUC	AUC Borji	AUC Judd	NSS	EMD
Infinite humans	1.00	1.00	0.80	0.87	0.91	3.18	0.00
DeepFix [19]	0.67	0.78	0.71	0.80	0.87	2.26	2.04
SALICON [12]	0.60	0.74	0.74	0.85	0.87	2.12	2.62
ML-Net (VGG-19)	**0.60**	**0.69**	**0.70**	**0.77**	**0.85**	**2.06**	**2.45**
Pan *et al.* - Deep [25]	0.52	0.58	0.69	0.82	0.83	1.51	3.31
BMS [37]	0.51	0.55	0.65	0.82	0.83	1.41	3.35
Deep Gaze 2 [20]	0.46	0.51	0.76	0.86	0.87	1.29	4.00
Mr-CNN [23]	0.48	0.48	0.69	0.75	0.79	1.37	3.71
Pan *et al.* - Shallow [25]	0.46	0.53	0.64	0.78	0.80	1.47	3.99
GBVS [10]	0.48	0.48	0.63	0.80	0.81	1.24	3.51
Rare 2012 Improved [28]	0.46	0.42	0.67	0.75	0.77	1.34	3.74
Judd [16]	0.42	0.47	0.60	0.80	0.81	1.18	4.45
eDN [35]	0.41	0.45	0.62	0.81	0.82	1.14	4.56

fled and AUC Judd. As it can be noticed, our solution outperforms all other approaches by a significant margin on all evaluation metrics.

We also evaluate our model on two others publicly available saliency benchmarks, MIT300 and CAT2000. Table 3 compares the results of our approach to the top performers of MIT300, while Table 4 reports performances on the CAT2000 benchmark. Our method outperforms the majority of the solutions in both leaderboards, and achieves competitive results when compared to the top ranked approaches.

4.5 Qualitative results

Figures 3 and 4 present instead a qualitative comparison showing ten randomly chosen input images from SALICON and MIT1003 datasets, their corresponding ground truth annotations and predicted saliency maps. These examples show how our approach is able to predict saliency maps that are very similar to the

Table 4. Comparison results on the CAT2000 test set [1].

	Sim	CC	sAUC	AUC Borji	AUC Judd	NSS	EMD
Infinite humans	1.00	1.00	0.62	0.84	0.90	2.85	0.00
DeepFix [19]	0.74	0.87	0.58	0.81	0.87	2.28	1.15
ML-Net (VGG-19)	**0.68**	**0.78**	**0.58**	**0.81**	**0.86**	**2.00**	**1.16**
BMS [37]	0.61	0.67	0.59	0.84	0.85	1.67	1.95
eDN [35]	0.52	0.54	0.55	0.84	0.85	1.30	2.64
Rare 2012 Improved [28]	0.54	0.57	0.59	0.81	0.82	1.44	2.72
GBVS [10]	0.51	0.50	0.63	0.79	0.80	1.23	2.99
Judd [16]	0.46	0.54	0.56	0.84	0.84	1.30	3.61

ground truth, while saliency maps generated by other methods are far less consistent with the ground truth.

5 Conclusions

In this paper we presented a new end-to-end trainable network for saliency prediction called ML-Net. Our solution learns a non-linear combination of multi-level features extracted from different layer of the CNN and a prior map. Qualitative and quantitative results on three public benchmarks show the validity of our proposal.

References

1. Borji, A., Itti, L.: Cat2000: A large scale fixation dataset for boosting saliency research. CVPR 2015 workshop on "Future of Datasets" (2015), arXiv preprint arXiv:1505.03581
2. Buswell, G.T.: How people look at pictures: a study of the psychology and perception in art. (1935)
3. Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A., Torralba, A.: Mit saliency benchmark. http://saliency.mit.edu/
4. Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., Durand, F.: What do different evaluation metrics tell us about saliency models? arXiv preprint arXiv:1604.03605 (2016)
5. Cerf, M., Frady, E.P., Koch, C.: Faces and text attract gaze independent of the task: Experimental data and computer model. Journal of vision 9(12), 10–10 (2009)
6. Gao, D., Vasconcelos, N.: Discriminant saliency for visual recognition from cluttered scenes. In: ANIPS (2004)
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International conference on artificial intelligence and statistics. pp. 249–256 (2010)
8. Goferman, S., Zelnik-Manor, L., Tal, A.: Context-aware saliency detection. IEEE TPAMI 34(10), 1915–1926 (2012)
9. Hadizadeh, H., Bajic, I.V.: Saliency-aware video compression. IEEE Trans. Image Process. 23(1), 19–33 (2014)

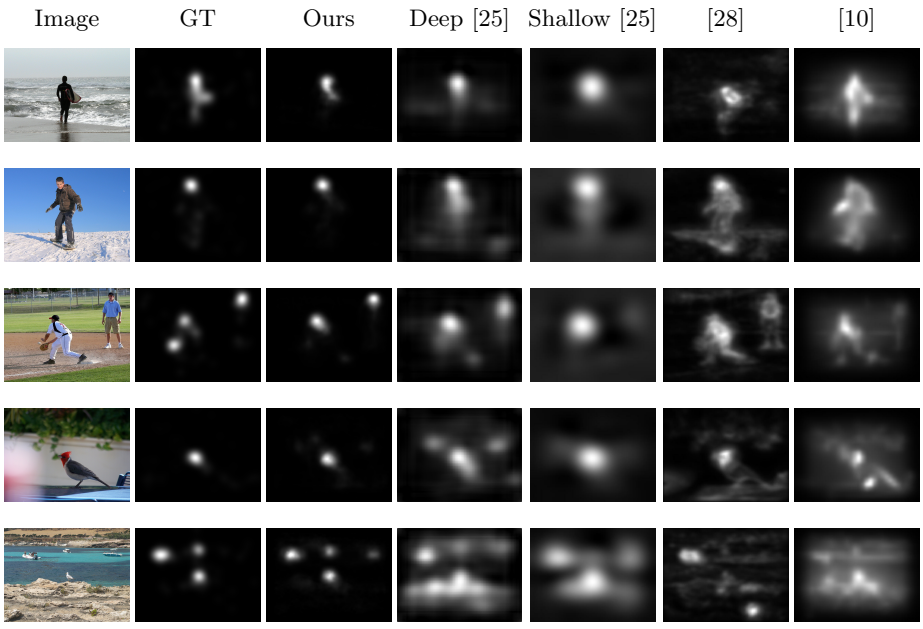

Fig. 3. Qualitative results on validation images from SALICON dataset [14].

10. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: ANIPS. pp. 545–552 (2006)
11. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: IEEE International Conference on Computer Vision and Pattern Recognition (2007)
12. Huang, X., Shen, C., Boix, X., Zhao, Q.: SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks. In: IEEE International Conference on Computer Vision. pp. 262–270 (2015)
13. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE TPAMI (11), 1254–1259 (1998)
14. Jiang, M., Huang, S., Duan, J., Zhao, Q.: Salicon: Saliency in context. In: IEEE International Conference on Computer Vision and Pattern Recognition. pp. 1072–1080. IEEE (2015)
15. Judd, T., Durand, F., Torralba, A.: A benchmark of computational models of saliency to predict human fixations (2012)
16. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: IEEE International Conference on Computer Vision (2009)
17. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. In: Matters of intelligence, pp. 115–141. Springer (1987)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: ANIPS. pp. 1097–1105 (2012)
19. Kruthiventi, S.S., Ayush, K., Babu, R.V.: DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations. arXiv preprint arXiv:1510.02927 (2015)
20. Kümmerer, M., Theis, L., Bethge, M.: Deep Gaze I: Boosting saliency prediction with feature maps trained on ImageNet. arXiv preprint arXiv:1411.1045 (2014)

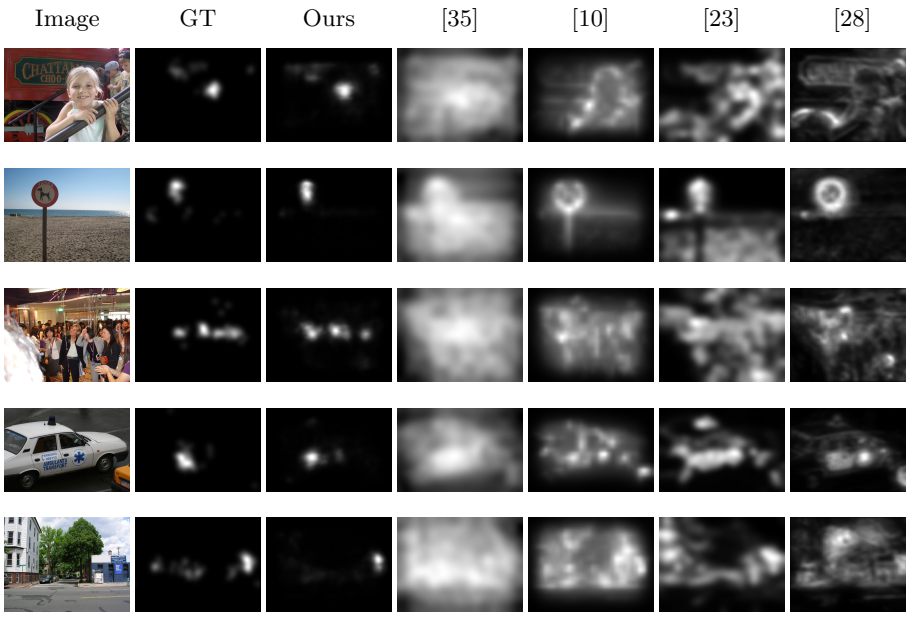

Fig. 4. Qualitative results on validation images from MIT1003 dataset [16].

21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision, pp. 740–755. Springer (2014)
22. Liu, F., Gleicher, M.: Video retargeting: automating pan and scan. In: ACM International Conference on Multimedia (2006)
23. Liu, N., Han, J., Zhang, D., Wen, S., Liu, T.: Predicting eye fixations using convolutional neural networks. In: IEEE International Conference on Computer Vision and Pattern Recognition (2015)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE International Conference on Computer Vision and Pattern Recognition (2015)
25. Pan, J., McGuinness, K., E., S., O'Connor, N., Giró-i Nieto, X.: Shallow and Deep Convolutional Networks for Saliency Prediction. In: IEEE International Conference on Computer Vision and Pattern Recognition (2016)
26. Peters, R.J., Iyer, A., Itti, L., Koch, C.: Components of bottom-up gaze allocation in natural images. Vision research 45(18), 2397–2416 (2005)
27. Riche, N., Duvinage, M., Mancas, M., Gosselin, B., Dutoit, T.: Saliency and human fixations: state-of-the-art and study of comparison metrics. In: IEEE International Conference on Computer Vision. pp. 1153–1160 (2013)
28. Riche, N., Mancas, M., Duvinage, M., Mibulumukini, M., Gosselin, B., Dutoit, T.: Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. Signal Processing: Image Communication 28(6), 642–658 (2013)
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vision 115(3), 211–252 (2015)

30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
31. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR abs/1409.1556 (2014), http://arxiv.org/abs/1409.1556
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE International Conference on Computer Vision and Pattern Recognition (2015)
33. Tatler, B.W.: The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. Journal of Vision 7(14), 4–4 (2007)
34. Torralba, A., Oliva, A., Castelhano, M.S., Henderson, J.M.: Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. Psychological review 113(4), 766 (2006)
35. Vig, E., Dorr, M., Cox, D.: Large-scale optimization of hierarchical features for saliency prediction in natural images. In: IEEE International Conference on Computer Vision and Pattern Recognition (2014)
36. Yang, Y., Song, M., Li, N., Bu, J., Chen, C.: What is the chance of happening: a new way to predict where people look. In: European Conference on Computer Vision, pp. 631–643. Springer (2010)
37. Zhang, J., Sclaroff, S.: Saliency detection: A boolean map approach. In: IEEE International Conference on Computer Vision. pp. 153–160 (2013)