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 crowdsourcing 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 can improve the prediction of observers' eye movements in real-world scenes.

Similarly, Goferman *et al.* [18] present a technique which aims at identifying salient regions that are distinctive with respect to both their local and global surroundings. Similarly to Cerf *et al.* [7], Judd *et al.* [19] propose an approach that combines low-level features (color, orientation and intensity) with high-level semantic information (i.e. the location of faces, cars and text) and show that this solution significantly improves the ability to predict eye fixations. In general, these approaches presented hand-tuned features or trained specific higher-level classifiers.

A first attempt to model saliency with deep convolutional networks (DCNs) has been recently proposed by Vig *et al.* [10]. They propose Ensembles of Deep Networks (eDN),

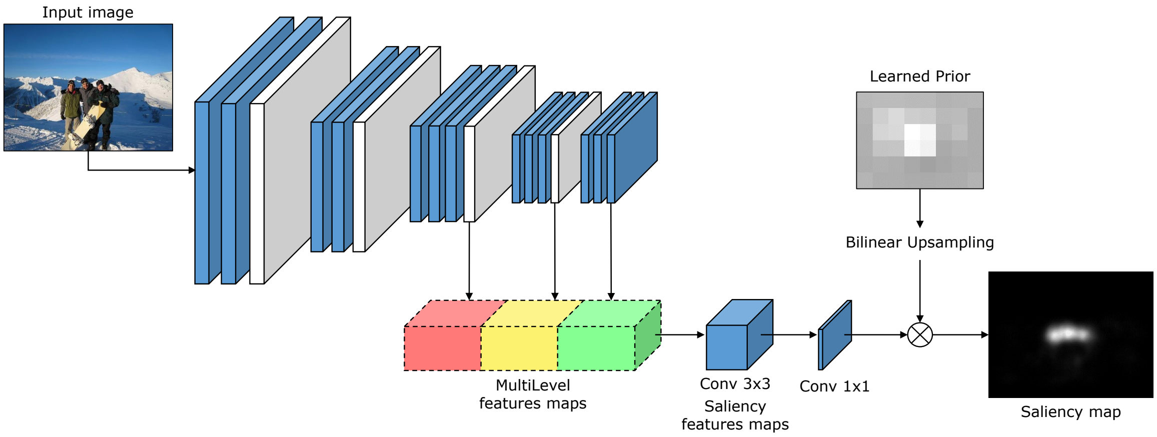


Fig. 1. Overview of our model (see Section III for details). A CNN is used to compute low and high level features from the input image. Extracted features maps are then fed to an Encoding network, which learns a feature weighting function to generate saliency-specific feature maps. A prior image is also 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 $k \times \left\lfloor \frac{H}{f} \right\rfloor \times \left\lfloor \frac{W}{f} \right\rfloor$, 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 $\left\lfloor \frac{W}{2^5} \right\rfloor \times \left\lfloor \frac{H}{2^5} \right\rfloor$, thus a Fully Convolutional model built upon the VGG-16 would output a saliency map rescaled by a factor of 32.

To limit this rescaling phenomenon, we remove the last pooling stage and decrease the stride of the last but one, while keeping unchanged its stride. This way, the output feature map of our feature extraction network are rescaled by a factor of 8 with respect to the input image. In the following, we will refer to the output size of the feature extraction network as $w \times h$, with $w = \left\lfloor \frac{W}{8} \right\rfloor$ and $h = \left\lfloor \frac{H}{8} \right\rfloor$. For reference, a complete description of the network is reported in Figure 2.

Encoding network We take feature maps at three different locations: the output of the third pooling layer (which contains 256 feature maps), that of the last pooling layer (512 feature maps), and the output of the last convolutional layer (512 feature maps). In the following, we will call these maps, respectively, 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 3×3 kernel. A final 1×1 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

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

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.

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

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

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. Weights for the encoding network are initialized according to [23], 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.

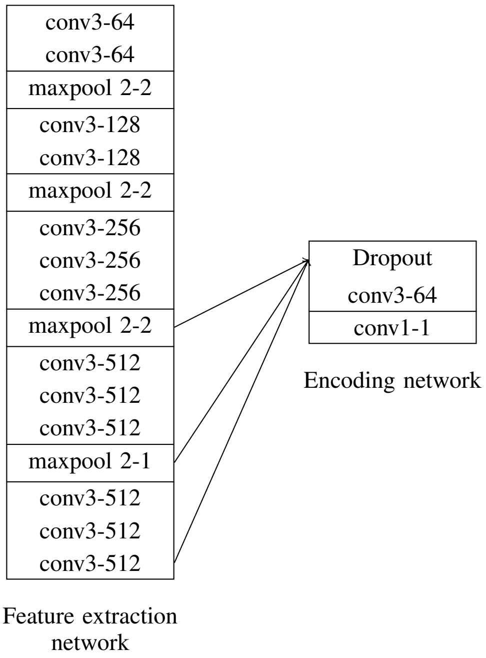


Fig. 2. Architecture of the proposed networks. The convolutional layer parameters are denoted as "conv<receptive field size>-<number of channels>". The ReLU activation function is not shown for brevity.

IV. Experimental Evaluation

A. Experimental setup

We evaluate the effectiveness of our proposal on two publicly available datasets: SALICON [24] and MIT300 [28]. SALICON is currently the largest public dataset for saliency prediction, and contains 10,000 training images, 5,000 validation images and 5,000 testing images, taken from the Microsoft CoCo dataset [29]. Saliency maps were generated by collecting mouse movements, as a replacement of eye-tracking systems, and authors show an high degree of similarity between their maps and those created from eye-tracking data.

MIT300 is one of the most commonly used datasets for saliency prediction, in spite of its limited size. It consists of 300 natural images, in which Saliency maps have been created from eye-tracking data of 39 observers. Saliency maps are not public available, and predictions must be submitted to the MIT saliency benchmark [30] for evaluation. Organizers of the benchmark suggest to use the MIT1003 [19] dataset for fine-tuning the model. This includes 1003 images taken from Flickr and LabelMe, generated through eye-tracking data of 15 participants.

Evaluation metrics Saliency prediction results are usually evaluated with a large variety of metrics: Similarity, Linear Correlation Coefficient (CC), AUC Shuffled, AUC Borji, AUC Judd, Normalized Scanpath Saliency (NSS) and Earth Mover's Distance (EMD). Some of these metrics compare the predicted saliency map with the ground truth map generated from

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

	CC	AUC shuffled	AUC Judd
Our method	**0.7430**	**0.7680**	**0.8660**
Deep Convnet - Pan *et al.* [14]	0.6220	0.7240	0.8580
Shallow Convnet - Pan *et al.* [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 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.

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 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.

B. Feature importance analysis

Our method relies on a non-linear combination of features extracted at different levels of a CNN. To validate our multilevel approach, we first evaluate the relative importance of features coming from each level. In the following, we define the importance of a feature as the extent to which a variation of the feature can affect the predicted map. Let us start by considering a linear model where different levels of a CNN are combined to obtain a pixel of the saliency map $\phi_i(\mathbf{x})$

$$\phi_i(\mathbf{x}) = w_i^T \mathbf{x} + \theta_i \tag{4}$$

where w_i and θ_i are the weight vector and the bias relative to pixel i, while $\mathbf{x}$ represents the activation coming from different levels of the feature extraction CNN, and $\phi_i(\mathbf{x})$ is the predicted saliency pixel. It is easy to see that the magnitude of elements in w_i defines the importance of the corresponding features. In the extreme case of a pixel of the feature map which is always multiplied by 0, it is straightforward to see that part of the feature map is ignored by the model, and has therefore no importance, while a pixel with high absolute values in w_i will have a considerable effect on the predicted saliency pixel.

In our model, $\phi_i(\cdot)$ is a highly non-linear function of the input, due to the presence of the encoding network and of the prior, thus the above reasoning is not directly applicable. Instead, given an image $\mathbf{x}_j$, we can approximate $\phi_i(\mathbf{x}_j)$ in the neighborhood of $\mathbf{x}_j$ as follows

$$\phi_i(\mathbf{x}_j) \approx \nabla\phi_i(\mathbf{x}_j)^T \mathbf{x} + \theta \tag{5}$$

An intuitive explanation of this approximation is that the magnitude of the partial derivatives indicates which features need to be changed to affect the model. Also notice that Eq. 5 is equivalent to a first order Taylor expansion.

To get an estimation of the importance of each pixel in an activation map regardless of the choice of $\mathbf{x}_j$, we can average the element-wise absolute values of the gradient computed in the neighborhood of each validation sample.

$$w_i = \frac{1}{N}\sum_{j=1}^{N}\left[\left|\frac{\partial\phi_i}{\partial x^1}(\mathbf{x}_j)\right|, \left|\frac{\partial\phi_i}{\partial x^2}(\mathbf{x}_j)\right|, \cdots, \left|\frac{\partial\phi_i}{\partial x^d}(\mathbf{x}_j)\right|\right] \tag{6}$$

where d is the dimensionality of $\mathbf{x}_j$. Then, to get the relative importance of each activation map, we average the values of w_i corresponding to that map, and L_1 normalize the resulting importances.

To get an estimate of the importance of feature maps extracted from the CNN, we should compute $\phi_i(\mathbf{x}_j)$ for every

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

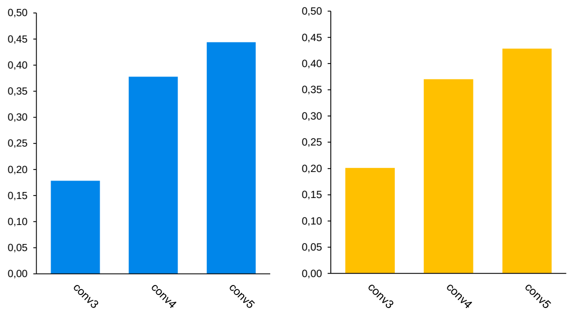
	Similarity	CC	AUC shuffled	AUC Borji	AUC Judd	NSS	EMD
Infinite humans	1.00	1.00	0.80	0.87	0.91	3.18	0.00
DeepFix [13]	**0.67**	**0.78**	0.71	0.80	**0.87**	**2.26**	**2.04**
SALICON [12]	0.60	0.74	0.74	0.85	**0.87**	2.12	2.62
Our method	0.59	0.67	0.70	0.75	0.85	2.05	2.63
Pan *et al.* - Deep Convnet [14]	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	**0.76**	**0.86**	**0.87**	1.29	4.00
Mr-CNN [22]	0.48	0.48	0.69	0.75	0.79	1.37	3.71
Pan *et al.* - 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 relative importance of activation maps coming from `conv3`, `conv4` and `conv5` on the model trained on SALICON. It is easy to notice that all features give a valuable contribution to the final result, and that while high level features are still the most relevant ones, medium level features have a considerable role in the prediction of the saliency map. This confirms our strategy to incorporate activations coming from different levels.

C. Comparison with state of the art

We evaluate our model on the SALICON dataset and on the MIT300 benchmark. In the first case, the network is trained on training images from the SALICON dataset, in the latter after training on SALICON we finetune on the MIT1003 dataset, as suggested by the MIT300 benchmark organizers. Images from all datasets were zero-padded to fit a $4:3$ aspect ratio, and then resized to 640×480.



(a) Contribution to mean (b) Contribution to variance

Fig. 3. Contribution of features extracted from `conv3`, `conv4` and `conv5` to prediction mean and variance.

Table I compares the performance of our model on SALICON in terms of CC, AUC shuffled and AUC Judd. As it can be observed, our solution outperforms all competitors by a margin of 12% according to CC metric, 4% and 1% according to AUC shuffled and AUC Judd.

For reference, the proposed solution is also evaluated on the MIT300 saliency benchmark which contains results of almost 60 different methods. Table II presents a comparison between our approach and top performers in this benchmark. Our method outperforms the vast majority of the approaches in the leaderboard of the benchmark, and achieves competitive results when compared to the top ranked approaches.

Figure 4 presents instead a qualitative comparison showing eight randomly chosen input images from SALICON and MIT1003 datasets, their corresponding ground truth annotations and predicted saliency maps. These examples clearly show how our approach is able to predict saliency maps that are very similar to the ground truth, while saliency maps generated by other methods are far less consistent with the ground truth.

Finally, we present some failure cases in Figure 5. As shown, when there is no a clear and explicit object in the image, eye fixations tend to be biased toward image center, which our model fails to predict.

V. CONCLUSIONS

This paper presented a novel deep learning architecture for saliency prediction. Our model learns a non-linear combination of medium and high level features extracted from a CNN, and a prior to apply to predicted saliency maps, while still being trainable end-to-end. Qualitative and quantitative comparisons with state of the art approaches demonstrate the effectiveness of the proposal on the biggest dataset and on the most popular public benchmark for saliency prediction.

ACKNOWLEDGMENT

We acknowledge the support of NVIDIA Corporation with the donation of the GPUs used in this work. This work was partially supported by the Fondazione Cassa di Risparmio di Modena project: “Vision for Augmented Experience” and the PON R&C project DICET-INMOTO (Cod. PON04a2 D).

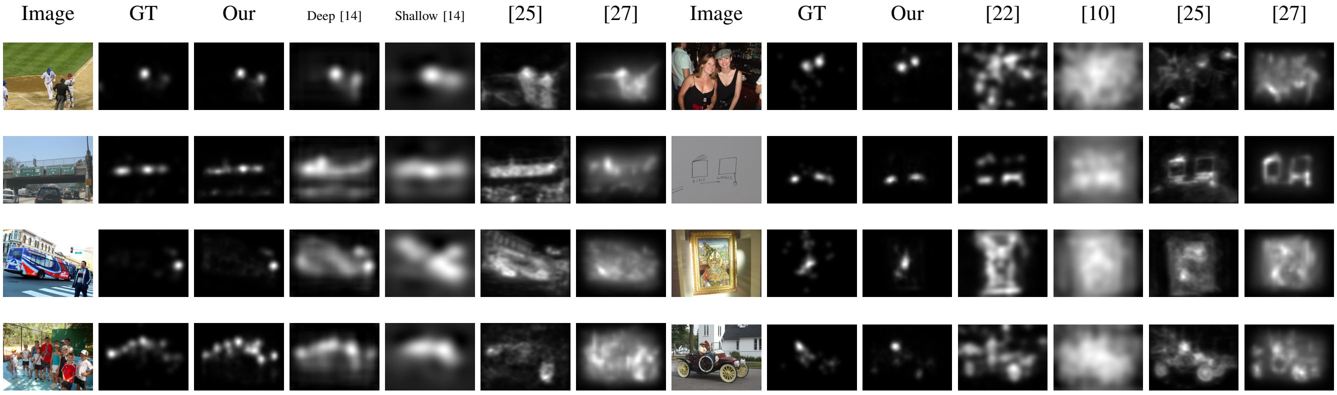


Fig. 4. Qualitative results and comparison to the state of the art. Left: validation images from SALICON dataset [24]. Right: validation images from MIT1003 dataset [19]. Best viewed in color.

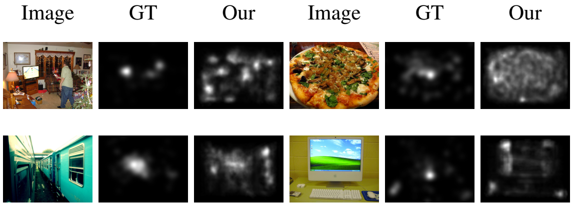


Fig. 5. Example of failure cases on validation images from SALICON dataset [24]. Best viewed in color.

References

[1] G. T. Buswell, "How people look at pictures: a study of the psychology and perception in art.," *University of Chicago Press*, 1935.

[2] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE TPAMI*, , no. 11, pp. 1254–1259, 1998.

[3] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *CVPR*, 2015.

[4] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *CVPR*, 2013.

[5] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *CVPR*, 2013.

[6] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *CVPR*, 2007.

[7] M. Cerf, E. P. Frady, and C. Koch, "Faces and text attract gaze independent of the task: Experimental data and computer model," *Journal of vision*, vol. 9, no. 12, pp. 10–10, 2009.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *ANIPS*, 2012, pp. 1097–1105.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[10] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *CVPR*, 2014.

[11] M. Kümmerer, L. Theis, and M. Bethge, "Deep Gaze I: Boosting saliency prediction with feature maps trained on ImageNet," *arXiv preprint arXiv:1411.1045*, 2014.

[12] X. Huang, C. Shen, X. Boix, and Q. Zhao, "SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks," in *ICCV*, 2015.

[13] S. S. Kruthiventi, K. Ayush, and R. V. Babu, "DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations," *arXiv preprint arXiv:1510.02927*, 2015.

[14] J. Pan, K. McGuinness, S. E., N. O'Connor, and X. Giró-i Nieto, "Shallow and Deep Convolutional Networks for Saliency Prediction," in *CVPR*, 2016.

[15] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *Journal of Vision*, vol. 7, no. 14, pp. 4–4, 2007.

[16] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of intelligence*, pp. 115–141. Springer, 1987.

[17] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search," *Psychological review*, vol. 113, no. 4, pp. 766, 2006.

[18] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE TPAMI*, vol. 34, no. 10, pp. 1915–1926, 2012.

[19] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *ICCV*, 2009.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[22] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, "Predicting eye fixations using convolutional neural networks," in *CVPR*, 2015.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[24] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *CVPR*, 2015.

[25] N. Riche, M. Mancas, M. Duvinage, M. Mibulumukini, B. Gosselin, and T. Dutoit, "Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis," *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 642–658, 2013.

[26] J. Zhang and S. Sclaroff, "Saliency detection: A boolean map approach," in *ICCV*, 2013.

[27] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *ANIPS*, 2006, pp. 545–552.

[28] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," in *MIT Technical Report*, 2012.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[30] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba, "Mit saliency benchmark," http://saliency.mit.edu/.

[31] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit, "Saliency and human fixations: state-of-the-art and study of comparison metrics," in *ICCV*, 2013.

[32] R. J. Peters, A. Iyer, L. Itti, and C. Koch, "Components of bottom-up gaze allocation in natural images," *Vision research*, vol. 45, no. 18, pp. 2397–2416, 2005.