Predicting human eye fixations via an LSTM-Based saliency attentive model

Predicting Human Eye Fixations via an LSTM-based Saliency Attentive Model

Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara

***Abstract*—Data-driven saliency has recently gained a lot of attention thanks to the use of Convolutional Neural Networks for predicting gaze fixations. In this paper we go beyond standard approaches to saliency prediction, in which gaze maps are computed with a feed-forward network, and we present a novel model which can predict accurate saliency maps by incorporating neural attentive mechanisms. The core of our solution is a Convolutional LSTM that focuses on the most salient regions of the input image to iteratively refine the predicted saliency map. Additionally, to tackle the center bias present in human eye fixations, our model can learn a set of prior maps generated with Gaussian functions. We show, through an extensive evaluation, that the proposed architecture overcomes the current state of the art on two public saliency prediction datasets. We further study the contribution of each key components to demonstrate their robustness on different scenarios.**

***Index Terms*—Saliency, Human Eye Fixations, Convolutional Neural Networks, Deep Learning**

I. Introduction

VISUAL cognition science has shown that humans, when observing a scene, do not focus on each region with the same intensity. Instead, selective mechanisms guide their gazes on salient and relevant parts of the image, focusing on different elements [1]. An intensive research effort aimed to emulate such selective visual mechanisms, as computational saliency can be applied to a wide range of applications like image retargeting [2], [3], object recognition [4], video compression [5], tracking [6] and other data-dependent tasks such as image captioning [7].

Traditional saliency prediction methods have followed biological evidences by defining features that capture low-level cues such as color, contrast and texture or semantic concepts such as faces, people and text [8], [9], [10], [11]. However, these techniques have failed to capture the wide variety of causes that contribute to define visual saliency maps.

With the advent of deep neural networks, saliency prediction has achieved strong improvements both thanks to specific architectures and to large annotated datasets [12], [13], [14], [15]. Although these approaches went beyond the limitations of hand-crafted models, no one has yet investigated the incorporation of neural attentive models [16] in saliency prediction. Neural attention is a computational paradigm which aims, with neural networks, at emulating the human attentive behavior. Human attention can be task driven (e.g. during driving or reading) or non-task driven, *i.e.* spontaneously driven by the observed scene. In this last non task-driven context, neural attention has been successfully applied to image captioning [16] and machine translation [17] to selectively focus on different parts of a sentence, and to action recognition [18] to focus on the relevant parts of a spatio-temporal volume. We claim that neural attention can be effective also for saliency prediction, as a powerful way to process saliency-specific features, extracted from a Convolutional Neural Network (CNN), to obtain an enhanced prediction.

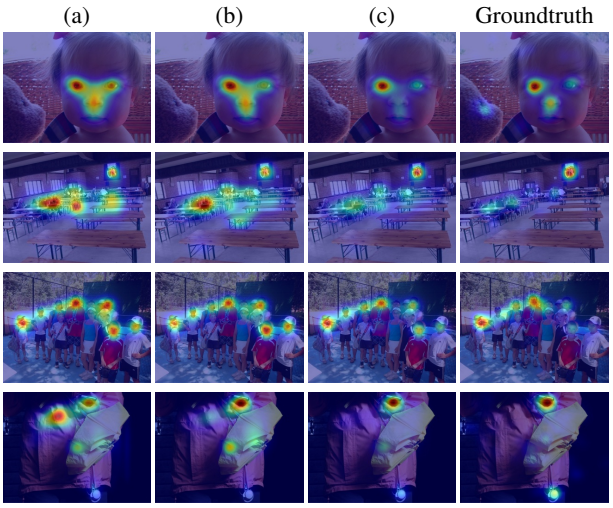


Fig. 1. Our Saliency Attentive Model (SAM) is composed of three main parts: a Dilated Convolutional Network (DCN), an Attentive Convolutional LSTM, and a set of learnable priors. We show some examples of saliency maps predicted by the DCN (a), the DCN with the Attentive ConvLSTM (b), and the DCN with the Attentive ConvLSTM and learned priors (c).

In this paper we propose a novel saliency prediction architecture that incorporates an Attentive Convolutional Long Short-Term Memory network (Attentive ConvLSTM) that can iteratively focus on relevant locations of the image to refine saliency features. This architecture is particularly original, because the LSTM model is used to achieve a refinement over an image, instead of handling a temporal sequence.

Since the rescaling caused by max-pooling and strides in convolutional layers deteriorates the performance of saliency prediction, we present an extension of two popular CNNs (namely, VGG-16 [19] and ResNet-50 [20]) which can reduce the downscaling effect and maintain spatial resolution. This expedient allows us to preserve detailed visual information and obtain improved feature extraction capabilities.

Moreover, in order to handle the tendency of humans to fix the center region of an image, we also introduce an explicit

M. Cornia, L. Baraldi and R. Cucchiara are with the Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Modena, Italy (e-mail: {marcella.cornia, lorenzo.baraldi, rita.cucchiara}@unimore.it).

G. Serra is with the Department of Computer Science, Mathematics and Physics, University of Udine, Udine, Italy (e-mail: giuseppe.serra@uniud.it).

center prior component. In fact, many studies conducted with eye-tracking devices have shown that eye fixations are biased toward the center of the scene [21], [22]. Unlike previous approaches that include handcrafted priors [10], [23], [12], [24], [25], our module keeps the architecture trainable end-to-end and can learn priors in an automatic way.

Figure 1 shows examples of saliency predictions obtained with our proposed solution, which we call Saliency Attentive Model (SAM), and with only some of its main components with respect to the groundtruth. We quantitatively validate our approach on three publicly available benchmark datasets: SALICON, MIT300 and CAT2000. Experimental results show that the proposed solution, by incorporating an attentive architecture and learned priors, significantly improves prediction.
To sum up, the contributions of this paper are threefold:

- We propose a novel Attentive ConvLSTM that sequentially enhances predicted saliency maps. To the best of our knowledge, this is the first work that incorporates attentive models in a saliency prediction architecture.
- Our network is able to learn the bias present in eye fixations, without the need to integrate this information manually.
- The proposed solution overcomes by a big margin the current state of the art on the largest dataset available for saliency prediction, the SALICON dataset. Moreover, on MIT300 and CAT2000 datasets our method achieves state of the art results showing competitive generalization properties.

We make the source code of our method and pre-trained models publicly available[1].

II. Related Work

Pioneering works on saliency prediction were based on the Feature Integration Theory proposed by Treisman *et al.* [26] in the eighties. Itti *et al.* [27] defined the first computational model to predict saliency on images: this work, inspired by Koch and Ullman [28], computed a set of individual topographical maps representing low-level cues such as color, intensity and orientation and combined them into a global saliency map. After this seminal work, a large variety of methods explored the same idea of combining complementary low-level features [29], [8], [30], [31] and often included additional center-surround cues [32], [11]. Other methods enriched predictions exploiting semantic classifiers for detecting higher level concepts such as faces, persons, cars and horizons [33], [10], [34], [9], [35].

A. Saliency and Deep Learning

Only recently, thanks to the large spread of deep learning techniques, the saliency prediction task has achieved a considerable improvement. One of the first proposals has been the *Ensemble of Deep Networks* (*e*DN) model by Vig *et al.* [25]. This model consists of three convolutional layers followed by a linear classifier that blends feature maps coming from the previous layers. After this work, Kümmerer *et al.* [12], [24] proposed two deep saliency prediction networks: the first, called *DeepGaze I*, based on the AlexNet model [36], while the second, *DeepGaze II*, built upon the VGG-19 network [19]. Liu *et al.* [37] presented a multi-resolution CNN (*Mr-CNN*) fine-tuned over image patches centered on fixation and non-fixation locations.

It is well known that deep learning approaches strongly depend on the availability of sufficiently large datasets. The publication of a large-scale eye-fixation dataset, SALICON [38], indeed contributed to a big progress of deep saliency prediction models. Huang *et al.* [13] introduced an architecture consisting of a deep neural network applied at two different image scales. They compared different standard CNN architectures such as AlexNet [36], VGG-16 [19] and GoogleNet [39] showing the effectiveness especially of the VGG network.

After this work, several deep saliency models based on the VGG network have been published [23], [40], [14], [41], [15], [42], [43], [44]. Accordingly, we proposed a new architecture, called *ML-Net* [15], which improved previous attempts by using features coming from multiple layers of a CNN and adding a learned prior map. In particular, we learned a matrix of weights which was applied to the output saliency map with a pixel-wise multiplication. The usage of centered priors has also been investigated in [23], where multiple predefined priors were fed to a convolutional layer.

In this work, instead, we model the center bias present in human gazes using multiple learned prior maps. This is completely different from the approaches of [15] and [23], since we let the network learn a set of Gaussian parameters, keeping it trainable end-to-end without predefined information.

Recently, Pan *et al.* [43] introduced *SalGAN*, a deep network for saliency prediction trained with adversarial examples. As all other Generative Adversarial Networks, it is composed by two modules, a generator and a discriminator, which combine efforts to produce saliency maps.

In this work, we also employ the ResNet [20] model to extract feature maps from the input image. The only other saliency model that exploits this network is that proposed by Liu *et al.* [45] and called *DSCLRCN*. This model simultaneously incorporates global and scene contexts to infer image saliency thanks to a deep spatial contextual LSTM which scan the image both horizontally and vertically.

B. Salient Object Detection

For the sake of clarity, salient object detection is slightly related to the topic of this work, even though it is a significantly different task. Salient object detection consists, indeed, in identifying a binary map indicating the presence of salient objects [46], [47], [48], [49]. In saliency prediction, instead, the objective is to predict a density map of eye fixations.

A saliency detection approach which is in some aspects related to our work is that of Kuen *et al.* [50], in which a recurrent (non convolutional) network provides salient object detection. At each timestep, their recurrent network outputs the parameter of a spatial transformation which is used to focus on a particular location of the image, and build the binary prediction for that location. Our recurrent network is,

[1]https://github.com/marcellacornia/sam

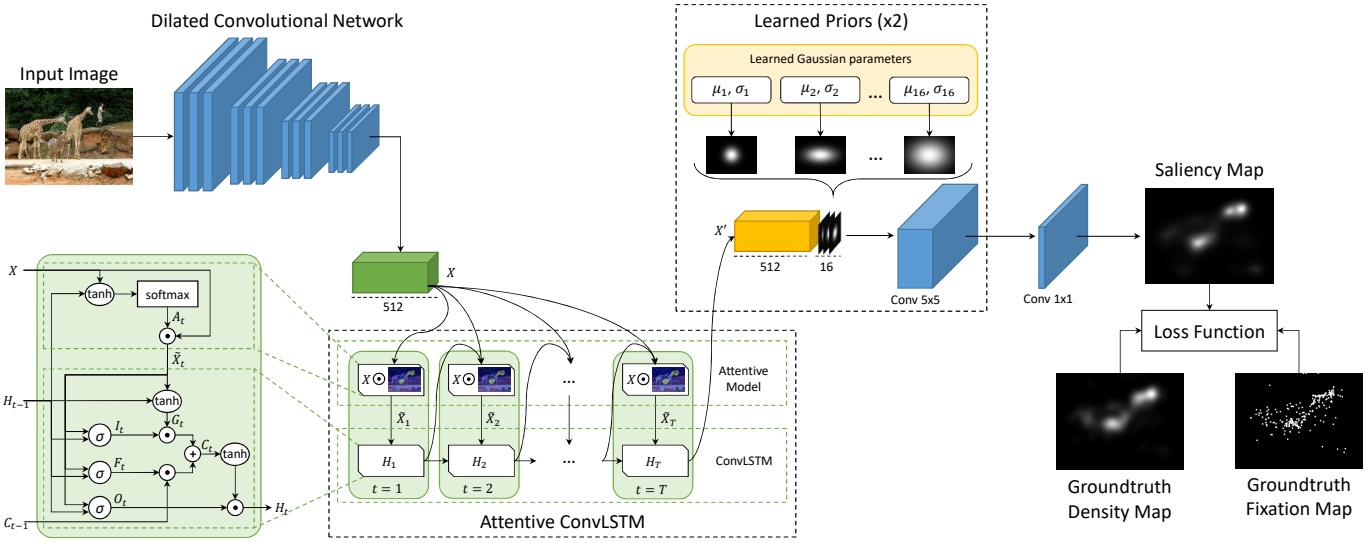


Fig. 2. Overview of our Saliency Attentive Model (SAM). After computing a set of feature maps on the input image through a new architecture called Dilated Convolutional Network, an Attentive Convolutional LSTM sequentially enhances saliency features thanks to an attentive recurrent mechanism. Predictions are then combined with multiple learned priors to model the tendency of humans to fix the center region of the image. During the training phase, we encourage the network to minimize a combination of different loss functions, thus taking into account different quality aspects that predictions should meet.

instead, convolutional, and is used to process saliency features by iteratively refining the prediction.

III. Model Architecture

In this section we present the architecture of our complete model, called SAM (Saliency Attentive Model).

The main novelty of our proposal is an Attentive Convolutional model, which recurrently process saliency features at different locations, by selectively attending to different regions of a tensor. This architecture, that for the first time uses an LSTM without the concept of time, is described in Section III-A.

Predictions are then combined with multiple learned priors which are used to model the human-gaze center bias (Section III-B). To extract feature maps from input images, we employ a Convolutional Neural Network model. Instead of using a pre-defined CNN, we propose a Dilated Convolutional Network to limit the rescaling effects which can worse saliency prediction performance (Section III-C). A new combination of different loss functions is finally used to train the whole network by simultaneously taking into account different quality aspects (Section III-D). The overall architecture of our model is shown in Figure 2.

A. Attentive Convolutional LSTM

Long Short-Term Memory networks [51] have achieved good performances on several tasks in which time dependencies are a key component [52], [53], [54], [55], but they can not be directly employed for saliency prediction, as they work on sequences of time varying vectors. We extend the traditional LSTM to work on spatial features: formally this is achieved by substituting dot products with convolutional operations in the LSTM equations. Moreover, we exploit the sequential nature of LSTM to process features in an iterative way, instead of using the model to deal with temporal dependencies in the input.

To explain our proposal of the attentive model, let's consider the LSTM scheme on the left part of Fig. 2. Here the LSTM takes as input a stack of features extracted from the input image (X in Fig. 2) and produces a refined stack of feature maps (X' in Fig. 2) entering in the learned prior module. The LSTM works by sequentially updating an internal state, according to the values of three sigmoid gates. Specifically, the update is driven by the following equations

$$I_t = \sigma(W_i * \tilde{X}_t + U_i * H_{t-1} + b_i) \quad (1)$$
$$F_t = \sigma(W_f * \tilde{X}_t + U_f * H_{t-1} + b_f) \quad (2)$$
$$O_t = \sigma(W_o * \tilde{X}_t + U_o * H_{t-1} + b_o) \quad (3)$$
$$G_t = \tanh(W_c * \tilde{X}_t + U_c * H_{t-1} + b_c) \quad (4)$$
$$C_t = F_t \odot C_{t-1} + I_t \odot G_t \quad (5)$$
$$H_t = O_t \odot \tanh(C_t) \quad (6)$$

here, the gates I_t, F_t, O_t, the candidate memory G_t, memory cell C_t, C_{t-1}, and hidden state H_t, H_{t-1} are 3-d tensors, each of them having 512 channels. $*$ represents the convolutional operator, all W and U are 2-d convolutional kernels, and all b are learned biases.

The input of the LSTM layer $\tilde{X}_t$ is computed, at each timestep (i.e. at each iteration), through an attentive mechanism which selectively focuses on different regions of the image. In particular, the LSTM computes an attention map which is generated by convolving the previous hidden state H_{t-1} and the input X, feeding the result to a $\tanh$ activation function and finally convolving with a one channel convolutional kernel

$$Z_t = V_a * \tanh(W_a * X + U_a * H_{t-1} + b_a). \quad (7)$$

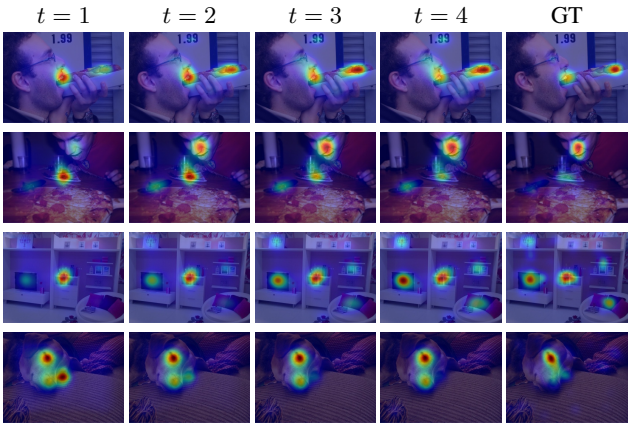


Fig. 3. Progressive refinement of predictions performed by the Attentive ConvLSTM. The first and the second row show a progressive change of focus in the saliency map, so that regions which were wrongly predicted as salient are progressively corrected, and truly salient regions are correctly identified. The third and the fourth row, instead, respectively show an increase and a reduction of saliency in regions of the image that have been (or have not been) considered as salient at the first timestep. In all cases, the result is a progressive approach of the saliency map to the groundtruth.

The output of this operations is a 2-d map from which we can compute a normalized spatial attention map through the *softmax* operator

$$A_t^{ij} = p(att_{ij}|X, H_{t-1}) = \frac{\exp(Z_t^{ij})}{\sum_i \sum_j \exp(Z_t^{ij})} \tag{8}$$

where A_t^{ij} is the element of the attention map in position (i, j). The attention map is applied to the input X with an element-wise product between each channel of the feature maps and the attention map

$$\tilde{X}_t = A_t \odot X. \tag{9}$$

Fig. 3 shows saliency predictions on four sample images, when using the output of the ConvLSTM module at different timesteps as input of the rest of the model. As can be noticed, predictions are progressively refined by modifying the initial map given by the CNN: this refinement results in an significant enhancement of the predictions.

B. Learned Priors

Psychological studies have shown that when observers look at images, their gazes are biased toward the center [21], [22]. This phenomena is mainly due to the tendency of photographers to position objects of interest at the center of the image. Also, when people repeatedly watch images with salient information placed in the center, they naturally expect to find the most informative content of the image around its center [22]. Another important reason that encourages this behavior is the interestingness of the scene [56]. Indeed, when there are not highly salient regions, humans are inclined to look at the center of the image.

We integrate a module which can learn multiple prior maps from data, keeping the architecture trainable end-to-end. We model the center bias by means of a set of Gaussian functions with diagonal covariance matrix. Means and variances are learned for each prior map, according to the following equation:

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\left(\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right)\right). \tag{10}$$

Our network learns the parameters of N Gaussian functions (in our experiments $N = 16$) and generates the relative prior maps. These maps are then concatenated with the output tensor of the Attentive ConvLSTM. Since this tensor has 512 channels, after the concatenation with learned prior maps, we obtain a tensor with 528 channels. The resulting tensor is fed through a convolutional layer with 512 filters. The entire learning prior module is replicated two times.

C. Dilated Convolutional Network

One of the main drawbacks of using CNNs to extract features for saliency prediction is that they considerably rescale the input image during the feature extraction phase, thus worsening the prediction accuracy. In the following, we describe a technique to limit the rescaling phenomena, and thus improve performance, on two recent feature extraction networks: the VGG-16 [19] and the ResNet-50 [20]. The same ideas could be applied, in principle, to any CNN architecture.

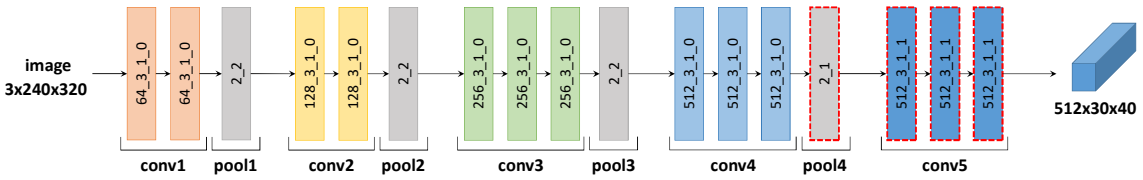
The VGG-16 network is composed by 13 convolutional layers and 3 fully connected layers. The convolutional layers are divided in five convolutional blocks where, each of them is followed by a max-pooling layer with a stride of 2.

The ResNet-50, instead of having a series of stacked layers that process the input image as in common CNNs, performs a series of residual mappings between blocks composed by a few stacked layers. This is obtained using shortcut connections that realize an identity mapping, *i.e.* the input of the block is added to its output. Residual connections help to avoid the accuracy degradation problem [57] that occurs with the increase of the network depth, and are beneficial also in the saliency prediction case, since they improve the feature extraction capabilities of the network.

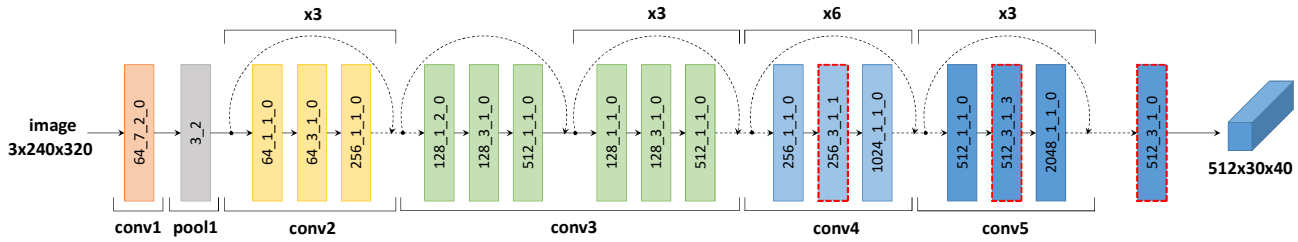
In particular, the ResNet-50 network consists of five convolutional blocks and a fully connected layer. The first block is composed by one convolutional layer followed by a max-pooling layer, both of them having a stride of 2, while the remaining four blocks are fully convolutional. All of these blocks, except the second one (`conv2`), reduce the dimension of feature maps with strides of 2.

Since the purpose of our network is to extract feature maps, we only consider convolutional layers and ignore fully connected layers which are present at the end of both networks. Moreover, it can be noticed that the downscaling factor of both of these architectures is particularly critical. For example, with an input image having a size of 240×320, the output dimension is 8×10, which is relatively small for the saliency prediction task. For this reason, we modify network structures to limit the rescaling phenomena.

For the VGG-16 model, we remove the last max-pooling layer and we set the stride to 1 in the last but one (as in Figure 4a). Besides, we introduce dilated convolutions [58] in the last convolutional block. This ensure that filters of the



(a) Dilated VGG Convolutional Network



(b) Dilated Residual Convolutional Network

Fig. 4. Overall architectures of Dilated Convolutional Networks based on the VGG-16 and ResNet-50 models. Convolutional and pooling blocks are respectively expressed in terms of `channels_kernel_stride_holes` and `kernel_stride`. On top of the ResNet model, we report the number of repetitions for each block. Red dashed edges indicate modified layers with respect to the original networks.

final block operate on the same receptive field for which they were originally conceived, and that pre-trained weights can be properly fine-tuned. In particular, we introduce holes of size 1 in the kernels of the block `conv5`.

For the ResNet-50, instead, we remove the stride and we introduce dilated convolutions in the last two blocks for the same reason (see Figure 4b). In particular, we introduce holes of size 1 in the kernels of the block `conv4` and holes of size $2^2 - 1 = 3$ in the kernels of the block `conv5`. The output of the residual network is a tensor with 2048 channels. To limit the number of feature maps, we fed this tensor into another convolutional layer with 512 filters.

The use of dilated convolutions allows the layer to have a larger receptive field without increasing the number of parameters. Thanks to these expedients, our saliency maps are rescaled by a factor of 8 instead of 32 as in the original VGG-16 and ResNet-50 models.

We include dilated convolutions also in prior layers, thus obtaining two convolutional layers with large receptive fields that allows us to capture the saliency of an object with respect to its neighborhood. We set the kernel size of these layers to 5 and the holes size to 3 achieving therefore a receptive field of 17×17.

The last layer of our model is a convolutional operation with one filter and a kernel size of 1 that extracts the final saliency map. Finally, the predicted saliency map is brought to its original size with a bilinear upsampling.

D. Loss function

Saliency predictions are usually evaluated through different metrics, in order to capture several quality factors. Inspired by this evaluation protocol, we introduce a new loss function given by a linear combination of three different saliency evaluation metrics. We define the overall loss function as follows:

$$L(\tilde{\mathbf{y}}, \mathbf{y}^{den}, \mathbf{y}^{fix}) = \alpha L_1(\tilde{\mathbf{y}}, \mathbf{y}^{fix}) + \beta L_2(\tilde{\mathbf{y}}, \mathbf{y}^{den}) + \gamma L_3(\tilde{\mathbf{y}}, \mathbf{y}^{den}) \tag{11}$$

where $\tilde{\mathbf{y}}$, $\mathbf{y}^{den}$ and $\mathbf{y}^{fix}$ are respectively the predicted saliency map, the groundtruth density distribution and the groundtruth binary fixation map, while α, β and γ are three scalars which balance the three loss functions. L_1, L_2 and L_3 are respectively the Normalized Scanpath Saliency (NSS), the Linear Correlation Coefficient (CC) and the Kullback-Leibler Divergence (KL-Div) which are commonly used to evaluate saliency prediction models.

The NSS metric was defined specifically for the evaluation of saliency models [59]. The idea is to quantify the saliency map values at the eye fixation locations and to normalize it with the saliency map variance

$$L_1\left(\tilde{\mathbf{y}}, \mathbf{y}^{fix}\right) = \frac{1}{N} \sum_i \frac{\tilde{\mathbf{y}}_i - \mu(\tilde{\mathbf{y}})}{\sigma(\tilde{\mathbf{y}})} \cdot \mathbf{y}_i^{fix} \tag{12}$$

where i indexes the i^{th} pixel, $N = \sum_i \mathbf{y}_i^{fix}$ is the total number of fixated pixels and $\tilde{\mathbf{y}}$ is normalized to have a zero mean and unit standard deviation.

The CC, instead, is the Pearson's correlation coefficient and treats the saliency and groundtruth density maps, $\tilde{\mathbf{y}}$ and $\mathbf{y}_{den}$, as random variables measuring the linear relationship between them. It is computed as

$$L_2\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right) = \frac{\sigma\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right)}{\sigma\left(\tilde{\mathbf{y}}\right) \cdot \sigma\left(\mathbf{y}^{den}\right)} \tag{13}$$

where $\sigma\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right)$ is the covariance of $\tilde{\mathbf{y}}$ and $\mathbf{y}^{den}$.

The KL-Div evaluates the loss of information when the distribution $\tilde{\mathbf{y}}$ is used to approximate the distribution $\mathbf{y}^{den}$, therefore taking a probabilistic interpretation of saliency and groundtruth density maps. Formally

$$L_3\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right) = \sum_i \mathbf{y}_i^{den} \log\left(\frac{\mathbf{y}_i^{den}}{\tilde{\mathbf{y}}_i + \epsilon} + \epsilon\right) \tag{14}$$

where i indexes the i^{th} pixel and ϵ is a regularization constant. The KL-Div is a dissimilarity metric and a lower value indicates a better approximation of the groundtruth by the predicted saliency map.

In Section V-A, we quantitatively justify the choice of our loss combination comparing our results with those obtained using single evaluation metrics as loss function.

IV. Experimental Setup

In this section we describe datasets and metrics used to evaluate the proposed model, and provide implementation details.

A. Datasets

Evaluation is carried out on four of the most popular saliency datasets which differ in terms of both image content and experimental settings.

- SALICON [38]: This is the largest available dataset for saliency prediction. It contains 10,000 training images, 5,000 validation images and 5,000 testing images, taken from the Microsoft COCO dataset [60]. Eye fixations are simulated with mouse movements: as shown in [38], there is an high degree of similarity between mouse-contingent saliency annotations and fixations recorded with eye-tracking systems. Groundtruth maps of the test set are not publicly available and predictions must be submitted to the SALICON challenge website[2] for evaluation.
- MIT1003 [10]: The MIT1003 dataset contains 1003 images coming from Flickr and LabelMe. Saliency maps have been created from eye-tracking data of 15 observers.
- MIT300 [61]: The MIT300 dataset is a collection of 300 natural images with saliency maps generated from eye-tracking data of 39 users. Saliency maps of this entire dataset are held out and we used the MIT Saliency benchmark [62] for evaluating our predictions.
- CAT2000 [56]: This dataset contains 4,000 images coming from a large variety of categories such as *Cartoons*, *Art*, *Satellite*, *Low resolution images*, *Indoor*, *Outdoor*, *Line drawings*, ect. It is composed of 20 different categories with 200 images for each of them. Saliency maps of the testing set, composed by 2,000 images, are not available and also in this case we submitted our saliency maps to the MIT Saliency benchmark [62].

B. Evaluation Metrics

There exists a large variety of metrics to evaluate saliency prediction models and the main difference between them concerns the ground-truth representation. In fact, saliency evaluation metrics can be categorized in location-based and distribution-based metrics [63], [64], [65]. The first category considers saliency maps at discrete fixation locations, while the second treats both ground-truth fixation maps and predicted saliency maps as continuous distributions.

The most widely used location-based metrics are the Area under the ROC curve, in its different variants of Judd (AUC) and shuffled (sAUC), and the Normalized Scanpath Saliency (NSS). The AUC metrics do not penalize low-valued false positives giving an high score for high-valued predictions placed at fixated locations and ignoring the others. Besides, the sAUC is designed to penalize models that take into account the center bias present in eye fixations. The NSS, instead, is sensitive in an equivalent manner to both false positives and false negatives.

For the distribution-based category, the most used evaluation metrics are the Linear Correlation Coefficient (CC), the Similarity (SIM) and the Earth Mover Distance (EMD). The CC treats both false positives and false negatives symmetrically, differently from the SIM that instead measures the intersection between two distributions and for this reason it is very sensitive to missing values. The EMD is a dissimilarity metric that penalizes false positives proportionally to the spatial distance from the groundtruth.

C. Implementation Details

We evaluate our model on SALICON, MIT300 and CAT2000 datasets. For the first dataset, we train the network on its training set and we use the 5,000 validation images to validate the model. For the second and the third dataset, we pre-train the network on SALICON and then fine-tune on MIT1003 and CAT2000 respectively as suggested by the MIT Saliency benchmark organizers. In particular, we use 903 randomly selected images of the MIT1003 to fine-tune the network and the remaining 100 as validation set. For the CAT2000 dataset, instead, we randomly choose 1,800 images of training set for the fine-tuning and we use the remaining 200 (10 for each category) as validation set.

For the SALICON, MIT1003 and MIT300 datasets, we resize input images to 240×320. Since images from MIT1003 and MIT300 have different sizes, we carry out a zero padding bringing images to have an aspect ratio of 4:3 and we then resize them to have the selected input size. Images from CAT2000 dataset, instead, have all the same input size of 1080×1920. For this reason, we resize all images of this dataset to 180×320.

Predictions of all datasets are slightly blurred with a Gaussian filter. After a validation process, we set the standard deviation of the Gaussian kernel to 7.

Weights of the Dilated Convolutional Networks are initialized with those of the VGG-16 and ResNet-50 models trained on ImageNet [66]. For the Attentive ConvLSTM, we initialize the recurrent weights matrices U_i, U_f, U_o and U_c as random orthogonal matrices. All W matrices and U_a are initialized by sampling each element from the Gaussian distribution of mean 0 and variance 0.05^2. The matrix V_a and all bias vectors are initialized to zero. Weights of all other convolutional layers of our model are initialized according to [67].

At training time, we randomly sample a minibatch containing K training saliency maps (in our experiments $K = 10$), and encourage the network to minimize the proposed loss function through the RMSprop optimizer [68].

Loss parameters α, β and γ are respectively set to -1, -2 and 10 balancing the contribution of each loss function.

[2]https://competitions.codalab.org/competitions/3791

TABLE I
ABLATION ANALYSIS OF SAM-VGG AND SAM-RESNET MODELS ON SALICON VALIDATION SET [38].

	SAM-VGG				SAM-ResNet			
	CC	sAUC	AUC	NSS	CC	sAUC	AUC	NSS
Plain CNN	0.743	0.765	0.870	2.333	0.771	0.762	0.876	2.404
Dilated Convolutional Network	0.801	**0.786**	0.876	3.122	0.823	0.774	0.879	3.187
DCN + Attentive ConvLSTM	0.809	0.784	0.878	3.142	0.841	0.786	0.885	3.256
DCN + Attentive ConvLSTM + Learned Priors	**0.830**	0.782	**0.883**	**3.219**	**0.844**	**0.787**	**0.886**	**3.260**

Differently from the KL-Div that is a dissimilarity metric and its value should be minimized, the CC and the NSS are to be maximized to predict better saliency maps. To this end, we set α and β as negative weights.

During the training phase, we set the initial learning rate to 10^{-4} and we decrease it by a factor of 10 every two epochs for the model based on the ResNet, and every three epochs for that based on the VGG network.

V. EXPERIMENTAL EVALUATION

In this section we perform analyses and experiments to validate the contribution of each component of the network. We also show quantitative and qualitative comparisons with other state of the art models.

A. Comparison between different loss functions

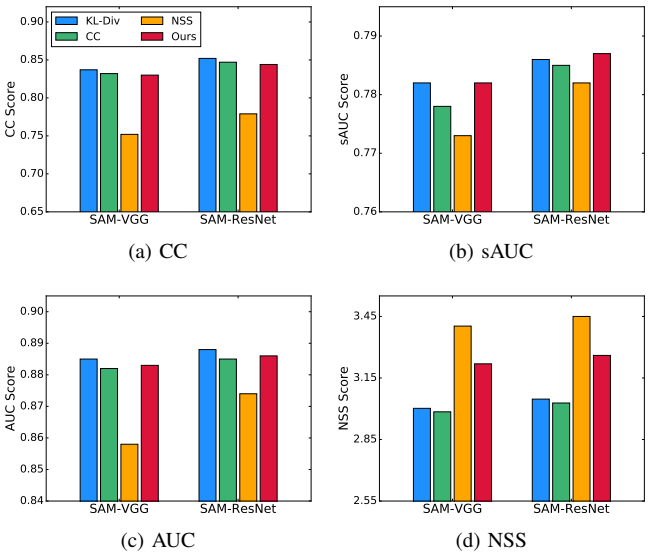
In Fig. 5 we compare results obtained by using single loss functions (KL-Div, CC, NSS) and our combination proposed in Section III-D. Results are reported for both versions of our model. We call SAM-VGG the model based on the VGG network and SAM-ResNet that based on the ResNet network.

As it can be seen, our combined loss achieves on average better results on all metrics. In particular, when the model is trained using the KL-Div or the CC metrics as loss function, the performance are good especially on the CC, while the model fails on the NSS. When the model is trained using the NSS metric, instead, it achieves better results only on the NSS and fails on all other metrics. Our combined loss reaches competitive results on all metrics differently from the other loss functions. For this reason, results of all following experiments are obtained by training the network with our combination of loss.

B. Model Ablation Analysis

We evaluate the contribution of each component of the architecture, using the SALICON validation set. To this end, we construct four different variations: the plain CNN architecture without the last fully convolutional layer (as a baseline), the Dilated Convolutional Network (DCN), the DCN with the proposed ConvLSTM model and the final version of our model with all its components.

Table I shows the results of the ablation analysis using both versions of our model. The results empathize that the overall architecture is able to predict better saliency maps in both SAM-VGG and SAM-ResNet variants and each proposed component gives an important contribution to the final performance. There is a constant improvement on all metrics. For example, the VGG baseline achieves a result of 0.743 in terms of CC, while the DCN achieves a relative improvement of $\frac{0.801-0.743}{0.743} = 7.8\%$. This result is further improved by a 1% when adding the Attentive ConvLSTM; finally, the learned priors add another important improvement of 2.6%. The ResNet baseline, instead, achieves a CC result of 0.771 that is improved by a 6.7% when adding the dilated convolutions. The Attentive ConvLSTM add an improvement of 2.2%, while learned priors slightly improve predictions by a 0.4%.



(a) CC (b) sAUC (c) AUC (d) NSS

Fig. 5. Comparison between different loss functions on SALICON validation set [38]. Each plot corresponds to a different evaluation metric (CC, sAUC, AUC and NSS). The four color bars represent the performance of our model trained with the considered loss functions. We report results of both SAM-VGG and SAM-ResNet models.

It is also noteworthy that, with our pipeline, a VGG-based network and a ResNet-based network achieve almost the same performance, so one of the two model can be equally chosen according to speed and memory allocation needs, without considerably affecting prediction performance.

C. Contribution of the attentive model and learned priors

Table II reports the performance of our model when using the output of the Attentive ConvLSTM module at different timesteps as input for the rest of the model. Results clearly

TABLE II
RESULTS ON SALICON VALIDATION SET [38] WHEN USING THE OUTPUT OF THE ATTENTIVE CONVLSTM MODULE AT DIFFERENT TIMESTEPS AS INPUT OF THE REST OF THE MODEL.

	T	CC	sAUC	AUC	NSS
SAM-VGG	1	0.821	0.777	**0.884**	3.168
SAM-VGG	2	0.827	0.777	0.883	3.224
SAM-VGG	3	0.828	0.781	0.883	**3.226**
SAM-VGG	4	**0.830**	**0.782**	0.883	3.219
SAM-ResNet	1	0.785	0.737	0.879	3.050
SAM-ResNet	2	0.829	0.764	**0.886**	3.214
SAM-ResNet	3	0.842	0.779	**0.886**	3.256
SAM-ResNet	4	**0.844**	**0.787**	**0.886**	**3.260**

TABLE III
COMPARISON RESULTS BETWEEN OUR LEARNED PRIORS AND THAT PROPOSED IN [15] ON SALICON VALIDATION SET [38].

	CC	sAUC	AUC	NSS
SAM-VGG (prior of [15])	0.811	**0.783**	0.878	3.150
SAM-VGG (learned priors)	**0.830**	0.782	**0.883**	**3.219**
SAM-ResNet (prior of [15])	0.840	0.785	0.884	3.249
SAM-ResNet (learned priors)	**0.844**	**0.787**	**0.886**	**3.260**

show that the refinement carried out by the Attentive model results in better performance. No further significant improvements were observed for $t > 4$.

To assess the effectiveness of our prior learning strategy, we compare it with the approach in [15], in which a low resolution prior map is learned and applied element-wise to the predicted saliency map, after performing a bilinear upsampling. We chose to compare our solution to that in [15] because it is the only other attempt to incorporate the center bias in a deep learning model without the use of hand-crafted prior maps. Results are reported in Table III: using multiple and Gaussian learned priors, instead of learning an entire prior map, with no pre-defined structure, shows to be beneficial according to all metrics.

D. Comparison with state of the art

We quantitatively compare our method with state of the art models on SALICON, MIT300 and CAT2000 test sets. We decide to sort model performances by the NSS metric as suggested by the MIT Saliency Benchmark [62], [64], [65].

Table IV shows the results on the SALICON dataset in terms of CC, sAUC, AUC and NSS. As it can be observed, our SAM-ResNet solution outperforms all competitors by a big margin especially on CC and NSS metrics and obtains the best result also on the sAUC. In particular, our method overcomes the other ResNet-based model [45] with an improvement of 1.5% according to NSS metric, 1.3% and 0.4% according to CC and sAUC. For a fair comparison with other methods, we include also the results achieved by our SAM-VGG model. The improvement with respect all other VGG-based methods is even more important than that obtained by the SAM-ResNet model. In detail, our SAM-VGG overcomes all other VGG-based methods with an improvement of 12.7% and 5.6% according to NSS and CC metrics.

TABLE IV
COMPARISON RESULTS ON SALICON TEST SET [38]. THE RESULTS IN BOLD INDICATE THE BEST PERFORMING METHOD ON EACH EVALUATION METRIC. (*) INDICATES CITATIONS TO NON-PEER REVIEWED TEXTS. METHODS ARE SORTED BY THE NSS METRIC.

	CC	sAUC	AUC	NSS
SAM-ResNet	**0.842**	**0.779**	0.883	**3.204**
DSCLRCN [45] (*)	0.831	0.776	0.884	3.157
SAM-VGG	0.825	0.774	0.881	3.143
ML-Net [15]	0.743	0.768	0.866	2.789
MixNet [44] (*)	0.730	0.771	0.861	2.767
SU [41]	0.780	0.760	0.880	2.610
SalGAN [43] (*)	0.781	0.772	0.781	2.459
SalNet [40]	0.622	0.724	0.858	1.859
DeepGazeII [24] (*)	0.509	0.761	**0.885**	1.336

The results on MIT300 and CAT2000 datasets are respectively reported in Tables V and VI. Our method achieves state of the art results on all metrics, except for the sAUC, on the CAT2000 dataset surpassing other methods by an important margin especially on SIM, CC, NSS and EMD metrics. On the MIT300 dataset, instead, we obtain results very close to the best ones.

Our model does not obtain a big gain in performance especially on the AUC metrics. This can be explained considering that the AUC metrics are primarily based on true positives without significantly penalizing false positives. For this reason, hazy or blurred saliency maps like the ones predicted by [24] achieve high AUC values [69], [34], despite being visually very different from the groundtruth annotations, as we will show in the following.

Qualitative results obtained by our models on SALICON and MIT1003 validations sets, together with those of other state of the art models, are shown in Figure 6. As it can be noticed, our network is able to predict high saliency values on people, faces, objects and other predominant cues. It also produces good saliency maps when images do not contain strong saliency regions, such as when saliency is concentrated in the center of the scene or when images portray a landscape. Moreover the proposed solution is able to infer the importance of different people present in the scene. In fact, people and faces are normally considered to be highly salient but, when there is more than one person, not all people have the same importance [70].

VI. CONCLUSION

We described a novel Saliency Attentive Model which can predict human eye fixations on natural images. The main novelty of the proposal is an Attentive Convolutional LSTM specifically designed to sequentially enhance saliency prediction. The same idea could be potentially employed in other tasks in which an image refinement is profitable. Furthermore, we captured an important property of human

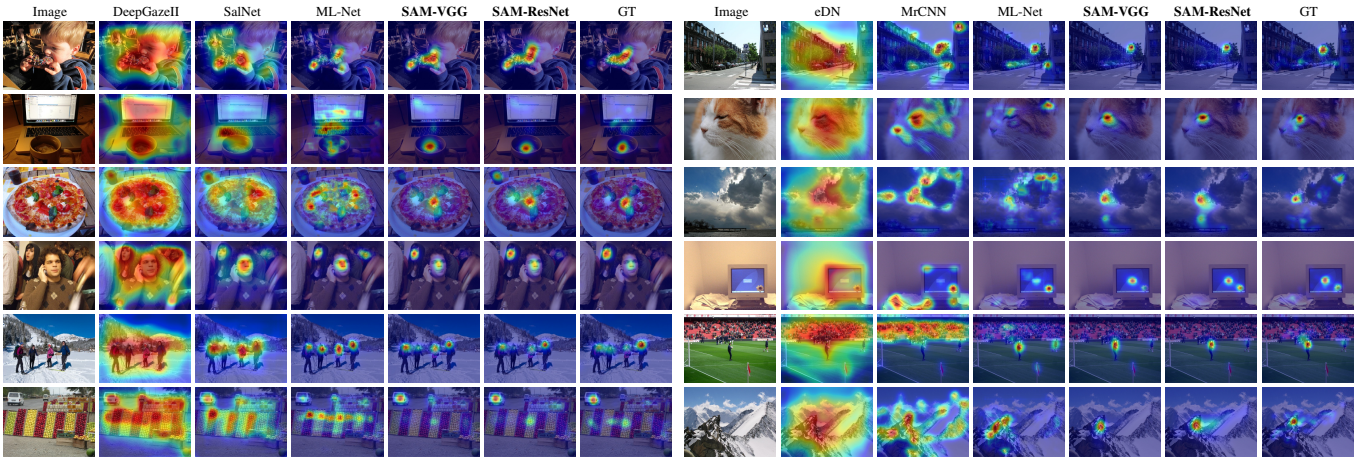


Fig. 6. Qualitative results and comparison with other state of the art models. Left images are from SALICON validation set [38], while right images are from MIT1003 validation set [10].

TABLE V
COMPARISON RESULTS ON MIT300 DATASET [61]. THE RESULTS IN BOLD INDICATE THE BEST PERFORMING METHOD ON EACH EVALUATION METRIC. (*) INDICATES CITATIONS TO NON-PEER REVIEWED TEXTS. METHODS ARE SORTED BY THE NSS METRIC.

	SIM	CC	sAUC	AUC	NSS	EMD
DSCLRCN [45] (*)	**0.68**	**0.80**	0.72	0.87	**2.35**	2.17
SAM-ResNet	**0.68**	0.78	0.70	0.87	2.34	2.15
SAM-VGG	0.67	0.77	0.71	0.87	2.30	2.14
DeepFix [23] (*)	0.67	0.78	0.71	0.87	2.26	**2.04**
SALICON [13]	0.60	0.74	**0.74**	0.87	2.12	2.62
PDP [14]	0.60	0.70	0.73	0.85	2.05	2.58
ML-Net [15]	0.59	0.67	0.70	0.85	2.05	2.63
SalGAN [43] (*)	0.63	0.73	0.72	0.86	2.04	2.29
iSEEL [42] (*)	0.57	0.65	0.68	0.84	1.78	2.72
SalNet [40]	0.52	0.58	0.69	0.83	1.51	3.31
BMS [11]	0.51	0.55	0.65	0.83	1.41	3.35
Mr-CNN [37]	0.48	0.48	0.69	0.79	1.37	3.71
DeepGazeII [24] (*)	0.46	0.52	0.72	**0.88**	1.29	3.98
GBVS [8]	0.48	0.48	0.63	0.81	1.24	3.51
eDN [25]	0.41	0.45	0.62	0.82	1.14	4.56

TABLE VI
COMPARISON RESULTS ON CAT2000 TEST SET [56]. THE RESULTS IN BOLD INDICATE THE BEST PERFORMING METHOD ON EACH EVALUATION METRIC. (*) INDICATES CITATIONS TO NON-PEER REVIEWED TEXTS. METHODS ARE SORTED BY THE NSS METRIC.

	SIM	CC	sAUC	AUC	NSS	EMD
SAM-ResNet	**0.77**	**0.89**	0.58	**0.88**	**2.38**	**1.04**
SAM-VGG	0.76	**0.89**	0.58	**0.88**	**2.38**	1.07
DeepFix [23] (*)	0.74	0.87	0.58	0.87	2.28	1.15
MixNet [44] (*)	0.66	0.76	0.58	0.86	1.92	1.63
iSEEL [42] (*)	0.62	0.66	**0.59**	0.84	1.67	1.78
BMS [11]	0.61	0.67	**0.59**	0.85	1.67	1.95
eDN [25]	0.52	0.54	0.55	0.85	1.30	2.64
GBVS [8]	0.51	0.50	0.58	0.80	1.23	2.99

gazes by optimally combining multiple learned priors, and effectively addressed the downscaling effect of CNNs. The effectiveness of each component has been validated through extensive evaluations, and we showed that our model achieves state of the art results on two of the most important datasets for saliency prediction. Finally, we contribute to further research efforts by releasing the source code and pre-trained models of our architecture.

ACKNOWLEDGMENT

This work was partially supported by the Fondazione Cassa di Risparmio di Modena project "Vision for Augmented Experience" (VAEX) and by the Emilia Romagna region project "Smart Architecture for Cultural Heritage in Emilia Romagna" (SACHER). We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

REFERENCES

[1] R. A. Rensink, "The Dynamic Representation of Scenes," *Visual Cognition*, vol. 7, no. 1-3, pp. 17–42, 2000.
[2] L.-Q. Chen, X. Xie, X. Fan, W.-Y. Ma, H.-J. Zhang, and H.-Q. Zhou, "A Visual Attention Model for Adapting Images on Small Displays," *Multimedia Syst.*, vol. 9, no. 4, pp. 353–364, 2003.
[3] V. Setlur, S. Takagi, R. Raskar, M. Gleicher, and B. Gooch, "Automatic Image Retargeting," in *Int. Conf. on Mobile and Ubiquitous Multimedia*, 2005.
[4] D. Walther, L. Itti, M. Riesenhuber, T. Poggio, and C. Koch, "Attentional selection for object recognition - a gentle way," in *International Workshop on Biologically Motivated Computer Vision*, 2002, pp. 472–479.
[5] H. Hadizadeh and I. V. Baji, "Saliency-Aware Video Compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19–33, 2014.
[6] V. Mahadevan and N. Vasconcelos, "Biologically Inspired Object Tracking Using Center-Surround Saliency Mechanisms," *IEEE TPAMI*, vol. 35, no. 3, pp. 541–554, 2013.
[7] Y. Sugano and A. Bulling, "Seeing with humans: Gaze-assisted neural image captioning," *arXiv preprint arXiv:1608.05203*, 2016.
[8] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *ANIPS*, 2006.
[9] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE TPAMI*, vol. 34, no. 10, pp. 1915–1926, 2012.

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

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

[12] M. Kümmerer, L. Theis, and M. Bethge, "DeepGaze I: Boosting saliency prediction with feature maps trained on ImageNet," in *ICLR Workshop*, 2015.

[13] 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.

[14] S. Jetley, N. Murray, and E. Vig, "End-to-End Saliency Mapping via Probability Distribution Prediction," in *CVPR*, 2016.

[15] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "A Deep Multi-Level Network for Saliency Prediction," in *ICPR*, 2016.

[16] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in *ICML*, 2015.

[17] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based Models for Speech Recognition," in *NIPS*, 2015.

[18] Z. Li, E. Gavves, M. Jain, and C. G. Snoek, "VideoLSTM Convolves, Attends and Flows for Action Recognition," *arXiv preprint arXiv:1607.01794*, 2016.

[19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[21] 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.

[22] P.-H. Tseng, R. Carmi, I. G. Cameron, D. P. Munoz, and L. Itti, "Quantifying center bias of observers in free viewing of dynamic natural scenes," *Journal of Vision*, vol. 9, no. 7, pp. 4–4, 2009.

[23] 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.

[24] M. Kümmerer, T. S. Wallis, and M. Bethge, "DeepGaze II: Reading fixations from deep features trained on object recognition," *arXiv preprint arXiv:1610.01563*, 2016.

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

[26] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[27] L. Itti, C. Koch, E. Niebur *et al.*, "A model of saliency-based visual attention for rapid scene analysis," *IEEE TPAMI*, vol. 20, no. 11, pp. 1254–1259, 1998.

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

[29] N. Bruce and J. Tsotsos, "Saliency based on information maximization," in *ANIPS*, 2005.

[30] R. Valenti, N. Sebe, and T. Gevers, "Image saliency by isocentric curvedness and color," in *ICCV*, 2009.

[31] E. Erdem and A. Erdem, "Visual saliency estimation by nonlinearly integrating features using region covariances," *Journal of Vision*, vol. 13, no. 4, pp. 11–11, 2013.

[32] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, "Saliency estimation using a non-parametric low-level vision model," in *CVPR*, 2011.

[33] M. Cerf, J. Harel, W. Einhäuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *ANIPS*, 2008.

[34] Q. Zhao and C. Koch, "Learning a Saliency Map using Fixated Locations in Natural Scenes," *Journal of Vision*, vol. 11, no. 3, pp. 9–9, 2011.

[35] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation," in *CVPR*, 2012.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *ANIPS*, 2012.

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

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

[39] 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.

[40] 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.

[41] S. S. Kruthiventi, V. Gudisa, J. H. Dholakiya, and R. Venkatesh Babu, "Saliency Unified: A Deep Architecture for Simultaneous Eye Fixation Prediction and Salient Object Segmentation," in *CVPR*, 2016.

[42] H. R. Tavakoli, A. Borji, J. Laaksonen, and E. Rahtu, "Exploiting inter-image similarity and ensemble of extreme learners for fixation prediction using deep features," *arXiv preprint arXiv:1610.06449*, 2016.

[43] J. Pan, C. Canton, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, "SalGAN: Visual Saliency Prediction with Generative Adversarial Networks," *arXiv preprint arXiv:1701.01081*, 2017.

[44] S. Dodge and L. Karam, "Visual Saliency Prediction Using a Mixture of Deep Neural Networks," *arXiv preprint arXiv:1702.00372*, 2017.

[45] N. Liu and J. Han, "A Deep Spatial Contextual Long-term Recurrent Convolutional Network for Saliency Detection," *arXiv preprint arXiv:1610.01708*, 2016.

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

[47] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *CVPR*, 2015.

[48] D. Zhang, J. Han, C. Li, and J. Wang, "Co-saliency detection via looking deep and wide," in *CVPR*, 2015.

[49] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *CVPR*, 2015.

[50] J. Kuen, Z. Wang, and G. Wang, "Recurrent Attentional Networks for Saliency Detection," in *CVPR*, 2016.

[51] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[52] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," in *CVPR*, 2015.

[53] A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions," in *CVPR*, 2015.

[54] Q. Wu, P. Wang, C. Shen, A. v. d. Hengel, and A. Dick, "Ask Me Anything: Free-form Visual Question Answering Based on Knowledge from External Sources," in *CVPR*, 2016.

[55] L. Baraldi, C. Grana, and R. Cucchiara, "Hierarchical Boundary-Aware Neural Encoder for Video Captioning," in *CVPR*, 2017.

[56] A. Borji and L. Itti, "CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research," in *CVPR Workshops*, 2015.

[57] K. He and J. Sun, "Convolutional Neural Networks at Constrained Time Cost," in *CVPR*, 2015.

[58] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[59] 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.

[60] 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.

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

[62] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba, "MIT Saliency Benchmark," http://saliency.mit.edu/.

[63] 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.

[64] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *arXiv preprint arXiv:1604.03605*, 2016.

[65] M. Kümmerer, T. S. Wallis, and M. Bethge, "Information-Theoretic Model Comparison Unifies Saliency Metrics," *National Academy of Sciences*, vol. 112, no. 52, pp. 16 054–16 059, 2015.

[66] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[67] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Int. Conf. on Artificial Intelligence and Statistics*, 2010.

[68] T. Tieleman and G. Hinton, "RMSProp: Divide the gradient by a running average of its recent magnitude," *Coursera Course: Neural Networks for Machine Learning*, 2012.

[69] A. Borji, H. R. Tavakoli, D. N. Sihite, and L. Itti, "Analysis of Scores, Datasets, and Models in Visual Saliency Prediction," in *ICCV*, 2013.

[70] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand, "Where should saliency models look next?" in *ECCV*, 2016.