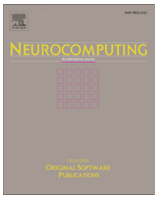
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Deep saliency models : The quest for the loss function

Alexandre Bruckert [a,*], Hamed R. Tavakoli [b], Zhi Liu [c], Marc Christie [a], Olivier Le Meur [a]

[a] Univ. Rennes, IRISA, CNRS, France
[b] Nokia Technologies, Finland
[c] Shanghai University, China

ARTICLE INFO

Article history:
Received 7 January 2020
Revised 18 May 2020
Accepted 8 June 2020
Available online xxxx
Communicated by Wenguan Wang

Keywords:
Deep saliency models
Loss functions
Human visual saliency

ABSTRACT

Deep learning techniques are widely used to model human visual saliency, to such a point that state-of-the-art performances are now only attained by deep neural networks. However, one key part of a typical deep learning model is often neglected when it comes to modeling visual saliency: the choice of the loss function.

In this work, we explore some of the most popular loss functions that are used in deep saliency models. We demonstrate that on a fixed network architecture, modifying the loss function can significantly improve (or depreciate) the results, hence emphasizing the importance of the choice of the loss function when designing a model. We also evaluate the relevance of new loss functions for saliency prediction inspired by metrics used in style-transfer tasks. Finally, we show that a linear combination of several well-chosen loss functions leads to significant improvements in performance on different datasets as well as on a different network architecture, thus demonstrating the robustness of a combined metric.

1. Introduction

Despite decades of research, visual attention mechanisms of humans remain complex to understand and even more complex to model. With the availability of large databases of eye-tracking and mouse movements recorded on images [9,24], there is now a far better understanding of the perceptual mechanisms. Significant progress has been made in trying to predict visual saliency, *i.e.* computing the topographic representation of visual stimulus strengths across an image. *Deep saliency models* have strongly contributed to this progress.

A challenge in the design of a deep saliency model is the choice of an appropriate loss function. An effort towards understanding loss function was carried out by Jetley et al. [22]. They proposed a probabilistic end-to-end framework and five relevant loss functions were studied. Despite this early effort, the choice of appropriate loss function remains an open challenge [2]. Yet, to the best of our knowledge, no one has investigated the role of loss functions in saliency models properly, despite its substantial influence on the quality of the model. Important questions therefore arise: how do different loss functions affect performance of deep saliency networks? Which loss functions perform better than others and on which metrics? Is there actually substantial benefits in combining loss functions? And how does the combination of loss functions perform with respect to individual loss functions? In this work, we investigate such questions by conducting a series of extensive experiments with both well-known and newly designed loss functions.

For this purpose, we first categorize loss functions into 4 categories: (i) pixel-based comparisons *e.g.* Mean Square Error, Mean Absolute Errors, Mean Exponential Absolute Difference, (ii) distribution-based metrics *e.g.* Kullback–Leibler divergence, Bhattacharya loss, binary cross-entropy, (iii) saliency-inspired metrics such as Normalized Scanpath Sali-ency or Pearson's correlation coefficient, or (iv) *style-transfer inspired losses*. We propose gathering two novel metrics in this paper, which are inspired from image style transfer, and measure the aggregation of distances computed at each convolutional layer, between the convoluted reference image and the generated saliency map.

We design a deep saliency model to provide a fixed and well controlled network architecture as a reference on which all the loss functions will be evaluated. Our evaluation strategy then consists in evaluating the impact of all the loss functions taken individually, on our fixed network with a fixed image dataset (MIT). By building on the common agreement that different metrics favor different perceptual characteristics of the image [2], we further explore how the combination of loss functions, typically aggregating pixel-based, distribution-based, saliency-based and perceptual-based functions, can influence the quality of the training. To

[*] Corresponding author.
E-mail address: alexandre.bruckert@irisa.fr (A. Bruckert).

https://doi.org/10.1016/j.neucom.2020.06.131
0925-2312/© 2020 Elsevier B.V. All rights reserved.

demonstrate the generalization capacity of our combined metric, we measure its impact on different datasets (CAT2000 and FiWi) and also with different network architectures (SAM-VGG and SAM-ResNet).

The contributions of this paper are therefore: (i) to demonstrate how the choice of the loss function can strongly improve (or depreciate) the quality of a deep saliency model without the need to increase the number of trained parameters; (ii) a proof of how an aggregation of carefully selected loss functions can lead to significant improvements, both on the fixed network architecture we propose, but also on some other architectures and datasets; (iii) an evaluation of the relevance of losses based on the measure of distance between representations of saliency maps in several feature spaces; (iv) a new deep saliency network, characterized by an excellent trade-off between number of trainable parameters and performances.

The paper is organized as follows. Section 2 presents the related works. The loss functions for training a deep architecture aiming to predict saliency map are described in Section 3. Section 4 presents the comprehensive analysis of loss functions and their combinations. Conclusions are drawn in the last section.

2. Related works

Computational models of saliency prediction, a long standing problem in computer vision, have been studied from so many perspectives that going through all is beyond the scope of this manuscript. We, thus, provide a brief account of relevant works and summarize them in this section. We refer the readers to [3,5] for an overview.

To date, from a computer vision perspective, we can divide the research on computational models of saliency prediction into two era (1) pre-deep learning, and (2) deep learning. During the pre-deep learning period, significant number of saliency models were introduced, e.g. [21,8,17,19,55,38,35,37,36], and numerous survey papers looked into these models and their properties, e.g. [3,56]. During this period the community converged into adopting eye tracking as a medium for obtaining ground truth and dealt with challenges regarding the evaluation and the models, *e.g.* [44,6]. This era was then replaced by saliency models based on deep learning techniques [2], which will be the main focus of this paper.

We therefore outline the recent research developments of deep saliency model era from two perspectives, (1) challenges of deep models and works that addressed them, and (2) the deep saliency models. We, then, stress the importance of task specific loss functions in computer vision.

2.1. Challenges of deep saliency models

The use of deep learning introduced new challenges to the community. The characteristics of most of the models shifted towards data intensive models based on deep convolutional neural networks (CNNs). To train a model, a huge amount of data is required, motivating the search for alternatives to eye tracking databases like mouse tracking [24], or pooling all the existing eye tracking databases into one [7].

To improve the training, Bruce et al. [7] investigated the factors required to take into account when relying on deep models, *e.g.*, pre-processing steps, tricks for pooling all the eye tracking databases together and other nuances of training a deep model. Authors, however, considered only one loss function in their study.

Tavakoli et al. [47] looked into the correlation between mouse tracking and eye tracking at finer details, showing the data from the two modalities are not exactly the same. They demonstrated that, while mouse tracking is useful for training a deep model, it is less reliable for model selection and evaluation in particular when the evaluation standards are based on eye tracking.

Given the sudden boost in overall performance by saliency models using deep learning techniques, Bylinskii et al. [10] reevaluated the existing benchmarks and looked into the factors influencing the performance of models in a finer detail. They quantified the remaining gap between models and human. They argued that pushing performance further will require high-level image understanding.

Recently Sen et al. [18] investigated the effect of model training on neuron representations inside a deep saliency model. They demonstrated that (1) some visual regions are more salient than others, and (2) the change in inner-representations is due to the task that original model is trained on prior to being fine-tuned for saliency.

2.2. Deep saliency models

The deep saliency models fall into two categories, (1) those using CNNs as a fixed feature extractors and learn a regression from feature space into saliency space using a none-neural technique, and (2) those that train a deep saliency model end-to-end. The number of models belonging to the first category is limited. They are not comparable within the context of this research because the regression is often carried out such that the error can not be back-propagated, *e.g.*, [50] employs support vector machines and [48] uses extreme learning machines. Our focus is, however, the second group.

Within end-to-end deep learning techniques, the main research has been on architecture design. Many of the models borrow the pre-trained weights of an image recognition network and experiment combining different layers in various ways. In other words, they engineer an encoder-decoder network that combines a selected set of features from different layers of a recognition network. In the following we discuss some of the most well-known models.

Huang et al. [20] proposed a multi-scale encoder based on VGG networks and learn a linear combination from responses of two scales (fine and coarse). Wang et al. [51] also use multi-sale saliency estimates, at three different levels, using a VGG encoding architecture, before learning a fusion of these estimates to create the final saliency map. Kümmerer et al. [29] use a single scale model using features from multiple layers of AlexNet. Similarly, Kümmerer et al. [33] and Cornia et al. [13] employed single scale models with features from multiple layers of a VGG architecture.

There has been also a wave of models incorporating recurrent neural architectures. Han and Liu [40] proposed a multi-scale architecture using convolutional long-short-term memory (ConvLSTM). It is followed by [14] using a slight modified architecture using multiple layers in the encoder and a different loss function. Recurrent models of saliency prediction are more complex than feed-forward models and more difficult to train. Moreover, their performance is not yet significantly better than some recent feed-forward networks such as EML-NET [23].

In the literature of deep saliency models, a loss function or a combination of several ones is chosen based on intuition, expertise of the authors or sometimes mathematical formulation of a model. Kümmerer et al. [30] introduces the idea that information-theory can be a good inspiration for saliency metrics. They use the information gain to explain how well a model performs compared to a gold-standard baseline. Consequently, they use the log-likelihood for a loss function in [31], achieving state-of-the-art results in saliency prediction. Wang et al. [54] propose to use a linear combination of functions derived from metrics commonly used for evaluating saliency models. Such combination proves very efficient to predict visual saliency on video stimuli. Jetley et al. [22] are part of the very few who specifically focused on the design of a loss functions for saliency models. They proposed the use of Bhattacharyya distance and compared it to 4 other probability distances. In this paper, in contrast to [22], we (1) adopt a principled approach to compare existing loss functions and their combinations and (2) investigate their convergence properties over different datasets and network architectures.

2.3. Deep learning, loss functions and computer vision

With the application of deep learning techniques to computer vision domain, the choice of appropriate loss function for a task has become a critical aspect of the model training. The computer vision community has been successful in developing task-tailored loss functions to improve a model, *e.g.* encoding various geometric properties for pose estimation [27], curating loss functions enforcing perceptual properties of vision for various generative deep models [25], exploiting the sparsity within the structure of problem, *e.g.* class imbalanced between background and foreground in detection problem, for reshaping standard loss functions and form a new effective loss functions [39]. Our efforts follows the same path to identify the effectiveness of a range of loss functions in saliency prediction.

3. Loss functions for deep saliency network

Before delving into the description of loss functions, we present the architecture of the convolutional neural network that will be used throughout this paper. After this presentation, we elaborate on the tested loss functions. The purpose of designing a new architecture is to master the whole training process, and to constraint the number of trainable parameters.

3.1. Proposed baseline architecture

Fig. 1 presents the overall architecture of the proposed model. Our architecture is based on the deep gaze network of [31] and on the multi-level deep network of [13]. Similar architectures, like [28] have proven very successful for saliency prediction, using only a small number of parameters. The pre-trained VGG-16 network [46] is used for extracting deep features of an input image (400×300) from layers conv3_pool, conv4_pool, conv5_conv3. Feature maps of layers conv4_pool and conv5_conv3 are rescaled to get feature maps with a similar spatial resolution.

These feature maps representing 1280 channels are then fed into a shallow network composed of the following layers: a first convolutional layer allows us to reduce by a factor ten the number of channels, which are then processed by an ASPP (an atrous spatial pyramid pooling [12]) of 4 levels. Each level has a convolution kernel of 3×3, a stride equal to 1 and a depth of 32. The dilatation rates are 1, 3, 6, and 12. The ASPP benefit is to catch information in a coarse-to-fine approach while keeping the resolution of the input feature maps. The output of the four pyramid levels are then merged together, *i.e.* leading to 4×32 maps. The last 1×1 convolutional layer reduces the data dimensionality to 1 feature map. This map is then smoothed by a Gaussian filter 5×5 with a standard deviation of 1. The activation function of these layers is the ReLU activation.

To evaluate each loss, we first trained the network over the MIT dataset composed of more than 1000 images [26]. We split this dataset into 500 images for the training, 200 images for the validation and the rest for the test. We use a batch size of 60, and the stochastic gradient descent. To prevent over-fitting, a dropout layer, with a rate of 0.25, is added on top the network. The learning rate is set to 0.001. During the training, the network was validated against the validation test to monitor the convergence and to prevent over-fitting. The number of trainable parameters is approximately 1,62 millions.

In the following section, we present the different tested loss functions used during the training phase.

3.2. Loss functions

Let $\mathcal{I} : \Omega \subset \mathcal{R}^2 \mapsto \mathcal{R}^3$ an input image of resolution $N \times M$. We suppose S and $\hat{S}$ the vectorized human and the predicted saliency maps, *i.e.* S and $\hat{S}$ are in $\mathcal{R}^{N \times M}$. Let also S^{fix} be a binary human eye fixations map, *i.e.* a $N \times M$ vectorized image with values equal to 1 where a fixation occurs, and 0 otherwise. In the following, we present 14 loss functions $\mathcal{L}$ tested in this study, declined into 30 different settings. We chose these loss functions among the most commonly used for deep visual saliency models. To the best of our knowledge, this selection is representative for this kind of task. Loss functions are classified into four categories according to their characteristics: pixel-based, distribution-based, saliency-inspired and perceptual-based.

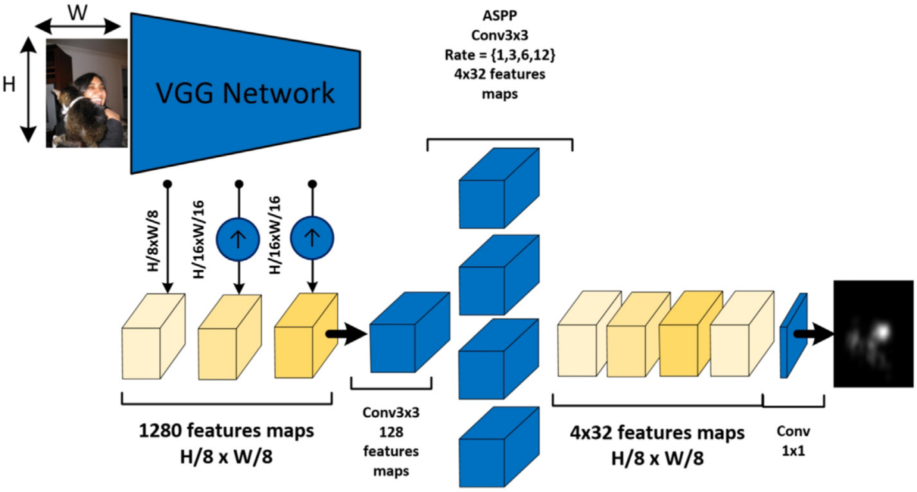


Fig. 1. Architecture of the proposed deep network.

3.2.1. Pixel-based loss functions

For pixel-based loss functions $\mathcal{L}$, we assume that S and $\hat{S}$ are in $[0,1]$. We evaluate the following loss functions:

- Mean Squared Error (MSE) measures the averaged squared error between prediction and ground truth:

$$\mathcal{L}(S,\hat{S}) = \frac{1}{N\times M}\sum_{i=1}^{N\times M}(S_i - \hat{S}_i)^2 \tag{1}$$

- Exponential Absolute Eifference (EAD):

$$\mathcal{L}(S,\hat{S}) = \frac{1}{N\times M}\sum_{i=1}^{N\times M}\Big(exp(|S_i - \hat{S}_i|) - 1\Big) \tag{2}$$

- Absolute Error (AE):

$$\mathcal{L}(S,\hat{S}) = \frac{1}{N\times M}\sum_{i=1}^{N\times M}|S_i - \hat{S}_i| \tag{3}$$

- Weighted MSE Loss (W-MSE):

$$\mathcal{L}(S,\hat{S}) = \frac{1}{N\times M}\sum_{i=1}^{N\times M}w_i\Big(S_i - \hat{S}_i\Big)^2 \tag{4}$$

The weight w_i allows to put more emphasis on errors occurring on salient pixels of the ground truth S. Two functions are tested in this paper. In [13], authors defined the loss function (MLNET): $w_i = \frac{1}{\alpha - S_i}$ and $\alpha = 1.1$. Therefore, when $S_i = 1$, the error is multiplied by a factor 10, whereas when $S_i = 0$, the multiplying factor is equal to 0.90. We also consider a weighting function based on a parametric sigmoid function (SIG-MSE): $w_i = \frac{k}{1+\exp(-k\times(S_i-\lambda))}$, where $k = 10$ and λ varies between 0 and 1.

3.2.2. Distribution-based loss functions

For the distribution-based loss functions, we consider the vectorized human map and the predicted saliency map as probability distributions. For that the network presented in Section 3.1 is modified in order to output pixel-wise predictions that can be considered as probabilities for independent binary random variables [41]. An element-wise sigmoid activation function is then added as being the last layer, replacing the ReLU activation. The following loss functions are investigated:

- Kullback–Leibler divergence (KLD) measures the divergence between the distribution S and $\hat{S}$:

$$\mathcal{L}(S,\hat{S}) = \sum_{i=1}^{N\times M}\hat{S}_i \log\frac{\hat{S}_i}{S_i} \tag{5}$$

- Bhattacharya loss (BHAT) measures the similarity between the distribution S and $\hat{S}$:

$$\mathcal{L}(S,\hat{S}) = \sum_{i=1}^{N\times M}\sqrt{S_i\hat{S}_i} \tag{6}$$

- Binary Cross Entropy (BCE) assumes that the saliency prediction $\hat{S}$ as well as the ground truth saliency map S are composed of independent binary random variables:

$$\mathcal{L}(S,\hat{S}) = -\sum_{i=1}^{N\times M}\Big(S_i\log\hat{S}_i + (1-S_i)\log(1-\hat{S}_i)\Big) \tag{7}$$

- Weighted Binary Cross Entropy (W-BCE): compared to the BCE loss, a global weight w is introduced to consider that there are much more non salient areas than salient areas [53]. It allows to put more emphasis on errors occurring when $S\rightarrow 1$ and $\hat{S}\rightarrow 0$ $(w \gg 0.5)$ or when $S\rightarrow 0$ and $\hat{S}\rightarrow 1$ $(w \ll 0.5)$:

$$\mathcal{L}(S,\hat{S}) = -\sum_{i=1}^{N\times M}\Big(w\times S_i\log\hat{S}_i + (1-w)(1-S_i)\log(1-\hat{S}_i)\Big) \tag{8}$$

- Focal Loss (FL): In order to deal with the large foreground–background class imbalance encountered during the training of dense detectors, Lin et al. [39] modified the binary cross entropy loss function. Such class imbalance is also relevant in the context of saliency prediction, for which the ground truth saliency map mainly consists of null or close to zero, creating a similar phenomenon. The approach is quite similar to W-BCE, except that the weight is locally adjusted and based on a tunable γ power of the predicted saliency. As in [39], we set by default the γ value equal to 2:

$$\mathcal{L}(S,\hat{S}) = -\sum_{i=1}^{N\times M}\Big((1-\hat{S}_i^{\gamma})\times S_i\log\hat{S}_i + \hat{S_i}^{\gamma}(1-S_i)\log(1-\hat{S}_i)\Big) \tag{9}$$

- Negative Logarithmic Likelihood (NLL): As shown by Kümmerer et al. [30], information theory provides strong insights when it comes to saliency models. For instance, in [31], they use maximum likelihood learning. Let N_{fix} be the number of fixations in an image. The logarithm of the prediction at the coordinates of each fixation is then computed:

$$\mathcal{L}(\hat{S}) = -\frac{1}{N_{fix}}\sum_{i=1}^{N\times M}\Big(S_i^{fix}\log\hat{S}_i\Big) \tag{10}$$

3.2.3. Saliency-inspired loss functions

Saliency predictions are usually evaluated using several metrics [34]. Those metrics are good candidates to use as loss functions, since they capture several properties that are specific to saliency maps.

- Normalized Scanpath Saliency (NSS): This metric was introduced in [43], to evaluate the degree of congruency between human eye fixations and a predicted saliency map. Instead of relying on a saliency map as ground truth, the predictions are evaluated against the true fixations map. The value of the saliency map at each fixation point is normalized with the whole saliency map variance:

$$\mathcal{L}(S^{fix},\hat{S}) = \frac{1}{N\times M}\sum_{i=1}^{N\times M}\left[\frac{\hat{S}_i - \mu(\hat{S}_i)}{\sigma(\hat{S}_i)}\right]S_i^{\mathrm{fix}} \tag{11}$$

- Pearson's Correlation Coefficient (CC) measures the linear correlation between the ground truth saliency map and the predicted saliency map:

$$\mathcal{L}(S,\hat{S}) = \frac{\sigma(S,\hat{S})}{\sigma(S)\sigma(\hat{S})} \tag{12}$$

where $\sigma(S,\hat{S})$ is the covariance of S and $\hat{S}$.

3.3. Style-transfer loss functions

3.3.1. VGG-16 encoder

We propose two new loss functions for deep saliency, that have been applied with success in image style-transfer problems [25,16]. The objective is to compare the representations of the ground truth and predicted saliency maps that are extracted from different layers of a fixed pre-trained convolutional neural network. The idea behind those losses is to take into account not only the saliency map, but also the deep hidden patterns that could exist, as well as the potential relationship between such patterns. Let $\phi_j(S)$ be the activation at the j^{th} layer of the VGG network when fed a saliency map S. $\phi_j(S)$ is then of size $C_j \times H_j \times W_j$, where C_j represents the number of filters, $H_j$and W_j represent the height and width of the feature maps at the layer j, respectively. We also denote J the set of layers from which we extract the representations. In this work, we extracted the outputs of the 5 pooling layers of a fixed VGG-16 network [46] pre-trained on the ImageNet dataset, representing a total of 1920 filters:

- Deep Features loss (DF) measures the Euclidean distance between the feature representations:

$$\mathcal{L}(S,\hat{S}) = \sum_{j\in J} \frac{1}{C_j \times H_j \times W_j} \|\phi_j(S) - \phi_j(\hat{S})\|^2 \tag{13}$$

- Gram Matrices of Deep Features loss (GM): In order to leverage the potential statistical dependency between features maps, we propose a new loss relying on Gram matrices. For this purpose, we reshape the output $\phi_j(S)$ into a matrix ψ of size $C_j \times (H_j W_j)$. Then, for each layer j, the Gram matrix $G_j^{\phi}(S)$ of size $C_j \times C_j$ is defined as follows:

$$G_j^{\phi}(S) = \frac{1}{C_j \times H_j \times W_j} \psi\psi^{\top} \tag{14}$$

The loss function is then the sum of the squared Frobenius norm of the difference between the Gram matrices $G_j^{\phi}, j \in J$:

$$\mathcal{L}(S,\hat{S}) = \sum_{j\in J} \|G_j^{\phi}(S) - G_j^{\phi}(\hat{S})\|_F^2 \tag{15}$$

3.3.2. Convolutional auto-encoder

The main flaw of style-transfer loss functions is the use of VGG network for extracting the deep feature maps. Indeed, VGG network has been trained on natural image sets, such as ImageNet. Therefore, the deep feature maps may not represent well the saliency map features. To make this point clear, we implemented a shallow convolutional auto-encoder which is trained only with saliency maps; the extracted deep feature maps are expected to be much more relevant for our problem.

Fig. 2 presents the architecture of the convolutional auto-encoder. Ground truth saliency maps from the CAT2000 database [4] are first downsampled to (128 × 128), and passed through three convolutional layers, each followed by a maxpooling layer. Those convolutional layers are respectively of depth 32, 64 and 128, and each level has a convolution kernel of size 3 × 3. After being encoded this way, the image is decoded through a symmetric network, three convolutional layers of respective depths 128, 64 and 32, with a convolution kernel of size 3 × 3, each followed by an upsampling layer. A last convolutional layer of depth 1 is applied to reconstruct the image. This network is trained using the binary crossentropy loss function.

We trained this auto-encoder on a subset of 1500 saliency maps from the CAT2000 database, holding out 500 for test. On those 500 test saliency maps, the Pearson's correlation coefficient between ground truth and the reconstructed saliency maps is 0.9874 ($p \ll 10^{-5}$), suggesting that the proposed network is efficient to reconstruct saliency maps. We can then assume that the features extracted by the encoder are relevant for representing saliency maps. Fig. 3 illustrates the reconstructed saliency maps. On top, the ground saliency maps are used as input of the auto-encoder. On bottom, the output of the network, *i.e.* the reconstructed saliency maps, is illustrated.

A set of deep features, called $\phi_{j(x)}$, are then extracted and used to compute the DF and GM losses presented in the previous section.

3.4. Center-bias regularization

Since our model does not take into account the center-bias with a learned prior, like in [14,13], we add a regularization term. We compute a center-bias map B as the mean of the ground truth maps from the training part of the MIT dataset, and add to the loss function the regularization term R_i for each pixel i:

$$R_i = \alpha(\hat{S}_i - B_i)^2 \tag{16}$$

We empirically set the parameter α to 0.1, even though it could be optimized to improve final results. This regularization will later be referred as R in Table 1. The center-bias map B is illustrated in Fig. 4.

3.5. Linear combinations

All presented loss functions evaluate different characteristics of the predicted saliency maps. We can then hypothesize that a linear combination of some of those loss functions could lead to better results, as it would aggregate the particularities of all measures (a strategy already adopted by [14]). Such approaches have shown good performances in problems for which, like saliency prediction, no single metric is fully representative of the performances of a model. For instance, in [52], authors combine five loss functions (weighted crossentropy, precision, recall, F-measure and mean absolute error) to predict object-level saliency maps. We decided to evaluate six linear combinations:

- LC-1: SIG-MSE ($\lambda = 0.55$) + R
- LC-2: KLD + CC + NSS
- LC-3: KLD + CC + NSS + R
- LC-4: KLD + CC + NSS + DF-VGG + GM-VGG
- LC-5: KLD + CC + NSS + DF-VGG + GM-VGG + SIG-MSE ($\lambda = 0.55$) + R

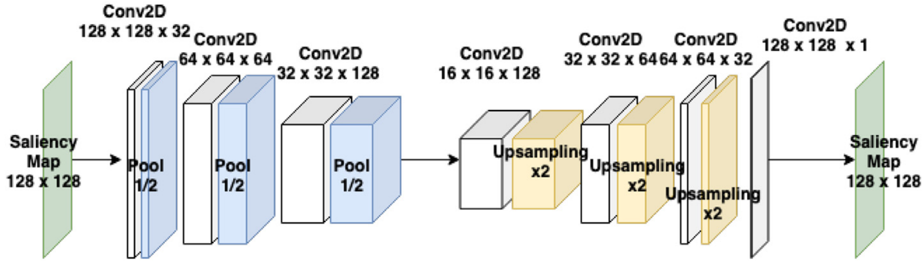


Fig. 2. Architecture of the feature-extractor auto-encoder. From an input saliency map of size 128 × 128, the encoder part extracts relevant features and the decoder part reconstruct a saliency map of the same resolution.

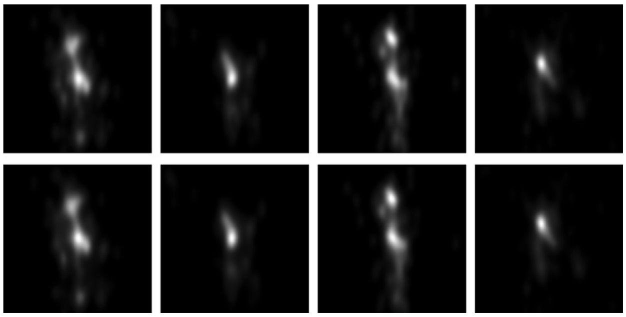


Fig. 3. Ground truth saliency maps (Top) and saliency maps reconstructed by the convolutional auto-encoder (Bottom).

- LC-6: KLD + CC + NSS + DF-AE + GM-AE + SIG-MSE $(\lambda = 0.55)$ + R

We followed the work of [14] to set the coefficients for the combination of KLD, CC and NSS: -1 for the NSS, -2 for the CC, and 10 for the KLD; other coefficients were set to 1.

4. Experiments

4.1. Testing protocols

To carry out the evaluation, we use seven quality metrics applied on the MIT benchmark [9,34]: CC (correlation coefficient, $CC \in [-1, 1]$), SIM (similarity, intersection between histograms of saliency, $SIM \in [0, 1]$), AUC (Area Under Curve, $AUC \in [0, 1]$), NSS (Normalized Scanpath Saliency, $NSS \in]-\infty, +\infty[$), EMD (Earth Mover Distance, $EMD \in [0, +\infty[$) and KL (Kullback Leibler divergence, $KL \in [0, +\infty[$).

The similarity degree between prediction and ground truth is computed over 299 images.

4.2. Loss function performance

Table 1 presents the performance obtained with the different loss functions when trained on the MIT dataset.

4.2.1. Which category of loss functions provide the best performances?

Results suggest that the pixel-based, the distribution-based and the saliency-inspired loss functions perform similarly. However, the perceptual-based loss functions we introduced, namely DF and GM, do not perform well individually compared to the aforementioned losses. The use of an autoencoder to select more relevant features than a pre-trained VGG-network also does not significantly improve the scores; however, those losses prove their interest when combined with other kind of losses.

4.2.2. Designing a stronger loss from weaker losses

Results suggest that a simple linear combination of well known loss functions increases the ability of the network to predict saliency maps. While keeping the number of trainable parameters unchanged, we succeed in improving up to 14% the correlation coefficient when we compared the best linear combination

Table 1
Performance of the loss functions after training on the MIT dataset. Best performances are in bold and the second and third best performances are in italic. (AUC-B = AUC-Borji; AUC-J = AUC-Judd)

	CC ↑	SIM ↑	AUC-J ↑	AUC-B ↑	NSS ↑	EMD ↓	KL ↓
Pixel-based loss functions							
MSE	0.6388	0.4492	0.8118	0.8363	2.0580	2.3279	0.9472
EAD	0.6725	0.4790	0.8326	0.8428	2.2133	2.1744	0.8592
AE	0.6426	0.4443	0.8145	0.8322	2.1388	2.3782	0.9616
MLNET-MSE	0.6904	**0.5962**	0.8416	0.8245	*2.2468*	**1.3581**	1.6042
SIG-MSE $(\lambda = 0.25)$	0.6929	*0.5896*	*0.8512*	0.8438	2.1897	*1.4120*	0.9876
SIG-MSE $(\lambda = 0.55)$	0.6725	0.5646	0.8505	0.8542	2.0799	*1.5137*	0.8478
SIG-MSE $(\lambda = 0.75)$	0.6440	0.5280	*0.8512*	**0.8637**	1.9480	1.7100	0.7686
Distribution-based loss functions							
BCE	0.6616	0.4712	0.8231	0.8380	2.1229	2.2600	0.8899
W-BCE $w = 0.9$	0.6333	0.4287	0.8308	0.8519	1.8734	2.4793	1.0003
W-BCE $w = 0.8$	0.6363	0.4273	0.8305	0.8531	1.8909	2.5026	1.0067
W-BCE $w = 0.7$	0.6409	0.4308	0.8179	0.8396	1.9636	2.4824	0.9976
W-BCE $w = 0.6$	0.6478	0.4335	0.8182	0.8412	2.0135	2.4800	0.9862
W-BCE $w = 0.5$	0.6739	0.4305	0.8420	*0.8582*	2.0911	2.4713	0.9871
W-BCE $w = 0.4$	0.6443	0.3992	0.8301	0.8468	2.0166	2.6625	1.0949
Focal Loss	0.6530	0.4294	0.8197	0.8403	1.9552	2.4839	0.9738
KLD	0.6326	0.4893	0.8356	0.8541	1.7913	2.0609	0.8336
Bhat	0.6203	0.5029	0.8429	0.8567	1.7321	1.9209	**0.7909**
NLL	0.6251	0.4973	0.8407	*0.8559*	1.7856	1.8734	*0.7955*
Saliency inspired loss functions							
CC	0.6943	0.4994	0.8411	0.8386	1.8201	2.1378	0.9157
NSS	0.6740	0.4325	0.8397	0.8216	**2.3142**	2.9964	1.3498
Style-transfer inspired loss functions							
Deep Features - VGG (DF-VGG)	0.6065	0.4772	0.8308	0.8259	1.9731	3.7546	0.9675
Gram Matrices - VGG (GM-VGG)	0.5911	0.4964	0.8371	0.8312	1.8357	2.0455	1.1993
Deep Features - Autoencoder (DF-AE)	0.6155	0.4842	0.8330	0.8405	1.9658	2.294	0.9501
Gram Matrices - Autoencoder (GM-AE)	0.5986	0.4899	0.8232	0.8391	1.9307	2.173	0.9322
Linear combinations							
LC-1	0.6813	0.5611	0.8507	0.8373	1.9734	3.1471	0.8349
LC-2	*0.7288*	0.5754	*0.8512*	0.8487	*2.2464*	2.1340	0.9571
LC-3	0.7176	0.5683	*0.8579*	0.8520	2.2147	2.8808	0.8912
LC-4	0.7192	0.5790	0.8492	0.8536	1.9652	2.3101	0.9387
LC-5	*0.7291*	0.5817	**0.8585**	*0.8563*	2.2094	2.5517	*0.8010*
LC-6	**0.7312**	*0.5885*	*0.8509*	*0.8597*	*2.3039*	2.6125	*0.7920*

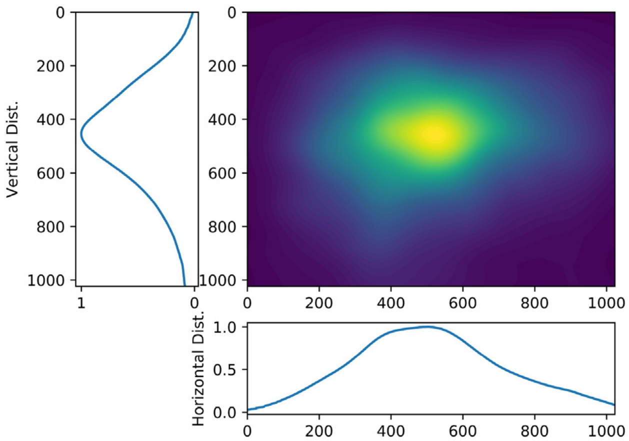


Fig. 4. Averaged colored saliency map of the training part of MIT dataset. Horizontal and vertical marginal distributions are also plotted, illustrating the center bias.

($CC = 0.7291$) to the classical MSE ($CC = 0.6388$). In a more general way, linear combinations of the loss functions systematically improve the results on most of the metrics. Such fluctuations between the performances of the different loss functions confirm our hypothesis that the choice of the loss function is a critical part of designing a deep saliency model. Moreover, the aggregated loss of LC-4 improves the SIM, AUC-B and KL scores compared to the classical aggregation LC-2. This reveals the influence of perceptual-based losses (DF-VGG + GM-VGG) in deep saliency models, and probably calls for future work in this direction. As shown in [32], the metrics used to evaluate visual saliency models are inconsistent, and it is hard for a model to perform well under all of these metrics. In that light, the use of a good combination of loss functions, especially mixing different categories, prove really efficient: despite being ranked first for only two metrics, the combination LC-6 ranks in the top-3 for each other metric, except for EMD. Beyond this quantitative analysis, Fig. 5 illustrates predicted saliency maps obtained for some of the tested loss functions. Qualitatively speaking, the saliency maps obtained when using the combined loss LC-6 look very similar to the ground truth maps. For instance, they are very condensed around the salient regions, with little noise.

4.3. Do the best losses generalize well over different datasets and a different architecture?

In this section, we test how well the best losses generalize over two additional datasets, *i.e.* CAT2000 and FiWi. To go further, we also investigate whether the best losses allow to increase the performance of another deep saliency model, having an architecture significantly different, *i.e.* SAM-VGG.

4.3.1. CAT2000 and FiWi datasets

CAT2000 eye tracking dataset [4] is composed of 2000 images belonging to 20 different categories whereas FiWi dataset [45] is composed of more than 140 screen shots of webpages. Performances of the model trained on the MIT dataset are given in Table 2. Results indicate that the loss function based on the linear combination of KLD, CC, NSS, DF-VGG, GM-VGG, SGI-MSE and R allows to significantly increase the ability to predict visual saliency. Compared to the MLNET-MSE loss function, the gain in terms of CC is 16.3% and 7.1% for CAT2000 and FiWi datasets, respectively.

4.3.2. SAM-VGG with linear combinations of loss functions

We also retrain SAM-VGG network over the MIT training dataset, as described in Section 3.1, by considering two linear combinations (LC-1 and LC-2).

Table 3 presents the results. They confirm that the linear combination approach improves the prediction (~5% in term of CC for the LC-2 loss). Even if the performances on the test datasets do not reach state-of-the-art techniques (see for instance [11] for FiWi), due to the fact that our model was only trained on natural images in MIT, the hierarchy of the loss functions we tested remains consistent, emphasizing the benefit of the linear combination. Fig. 6 illustrates a predicted saliency map generated when the SAM-VGG network is trained with a *stand-alone* loss function, *i.e.* MLNET-MSE, and with a combination of loss functions.

All these observations emphasize that the proposed loss function generalizes well independently of the dataset and the network architecture.

4.4. Analysis of the distribution of the results

The main reason that motivated the study of linear combinations of losses was to evaluate whether or not such an aggregation could compensate for the weaknesses of some of the losses. In order to validate that hypothesis, we propose in Fig. 7 a t-SNE representation [49] of the images from the testing holdout of the MIT database based on their scores with five different losses (one for each category): MLNET-MSE (b), W-BCE ($w = 0.5$) (c), CC (d), DF-AE (e) and LC-6 (f).

Fig. 7(a) illustrates a multilayer representation, in which each layer is composed of the 300 tested images, distributed according to the similarity scores between the ground truth and the prediction; in total it then represents 300 × 5, since we proposed to test five different models. More specifically, the blue layer gives the scores-based distribution when the model is trained with the MLNET loss function. This layer is also represented in Fig. 7(b), alone. The yellow layer corresponds to the best model, trained when the LC-6 loss function is used. Fig. 7(f) represents this layer alone.

Overall, this shows a relatively homogeneous distribution between the scores of the different losses, except for LC-6. Fig. 7 (f) showing the score-based distribution when using the LC-6 loss function, provides a representation that shows two main clusters, meaning that this model acts in a more even way. It indicates that combining the losses leads to improvement, even on picture for which saliency prediction was a hard task, *i.e.* pictures with lower scores.

Fig. 8 shows a zoom (top-left of Fig. 7(a)) on the same part of the t-SNE representation for each of the losses (in this case, images with the lowed scores). It appears that those images are the same for each loss, highlighting the fact that models often fail to predict accurate saliency maps on the same stimuli. However, subfigure (e), representing the same area for the LC-6 loss shows that a relatively high number of those “difficult” images are not in this score zone, meaning that re-training the model using the aggregation allows to correct, or at least attenuate the mistakes made by each individual loss.

4.5. Generalization using the SALICON dataset

In this study, we used a model based on pre-trained features. However, since the large-scale SALICON dataset was released, a lot of deep saliency models have been trained end-to-end, in order to learn saliency-specific features. Thus, in order to expand the scope and the representation power of this study, we trained the network on the SALICON dataset [24]. We use the best loss function, as well as one loss function from each category. SALICON

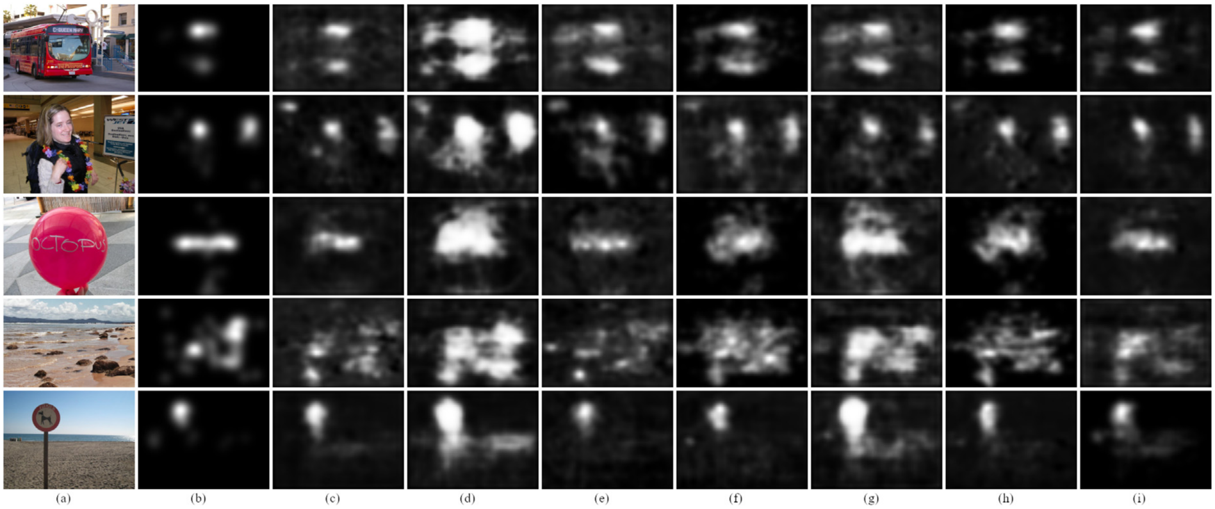


Fig. 5. (a) Visual stimulus; (b) Ground truth saliency map; (c) LC-6; (d) NSS; (e) LC-3; (f) MLNET-MSE; (g) KLD; (h) SIG-MSE ($\lambda = 0.55$); (i) CC.

Table 2
Performance of proposed model over CAT2000 and FiWi datasets with MLNET-MSE (W-MSE), LC-2 and LC-5. Best performances are in bold. (AUC-B = AUC-Borji; AUC-J = AUC-Judd)

	CC ↑	SIM ↑	AUC-J ↑	AUC-B ↑	NSS ↑
CAT2000					
W-MSE	0.5080	0.4017	0.8221	0.8016	**1.9486**
LC 2	0.5535	**0.4261**	0.8273	0.8187	1.9332
LC 5	**0.5937**	0.4203	**0.8309**	**0.8372**	1.9375
FiWi					
W-MSE	0.3954	0.3872	0.7312	0.7114	0.8050
LC 2	0.4118	0.4157	0.7621	0.7390	**0.8214**
LC 5	**0.4236**	**0.4183**	**0.7636**	**0.7681**	0.8205

Table 3
Performance of SAM-VGG over MIT dataset with MLNET-MSE (W-MSE), LC-2 and LC-5. Best performances are in bold. (AUC-B = AUC-Borji; AUC-J = AUC-Judd)

	CC ↑	SIM ↑	AUC-J ↑	AUC-B ↑	NSS ↑
MIT					
W-MSE	0.7351	**0.6769**	0.8521	0.7884	2.0037
LC 2	0.7499	0.6502	0.8635	0.7912	**2.1694**
LC 5	**0.7511**	0.6472	**0.8712**	**0.8017**	2.0741

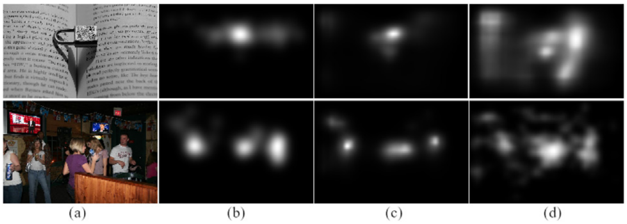


Fig. 6. Example of good predictions by the combination loss while a single loss makes bad predictions (for SAM-VGG model). (a) original image; (b) Ground truth saliency map; (c) LC-5 combination ($CC = 0.8681$ and 0.7967); (d) MLNET-MSE ($CC = 0.4320$ and 0.4491).

dataset is composed of 15,000 images (10,000 for training and 5000 for validation) along with mouse-tracking data. In [24], authors show that mouse-tracking data are highly correlated with actual eye-tracking data. We trained the network end-to-end, initializing the VGG-16 network with ImageNet weights. However, in order to lessen the potential bias related to the differences between mouse-tracking and eye-tracking, we then fine-tuned the network using 100 images from the training split of the MIT dataset, and 100 images for validation. We tested this new setting on the same test holdout of the MIT dataset as for Table 1.

Results exposed in Table 4 clearly highlight the interest of a combined loss functions over single losses. The LC-6 combination outperforms all other single losses on every metric, except the EMD.

We also compare our model and its performances to several other saliency models. Table 5 presents the performance of the proposed model when trained on the SALICON dataset with the LC-6 loss function, compared to several other existing models, *i.e.* Itti [21], Rare2012 [44], GBVS [17], AWS [15], Sam-ResNet & Sam-VGG [14], SalGan [41], ShallowNet & DeepConvNet [42]. All models are evaluated on the test dataset as defined in Section 3.1. According to the evaluation, the proposed model achieves state-of-the-art performances, as it is in the top-4 for each metric. It is also worth recalling that the number of trainable parameters is very low compared to other state-of-the-art networks, such as Sam-ResNet. The best performing models are Sam-ResNet and SalGan. We also retrained Sam-ResNet model with the LC-6 loss functions,

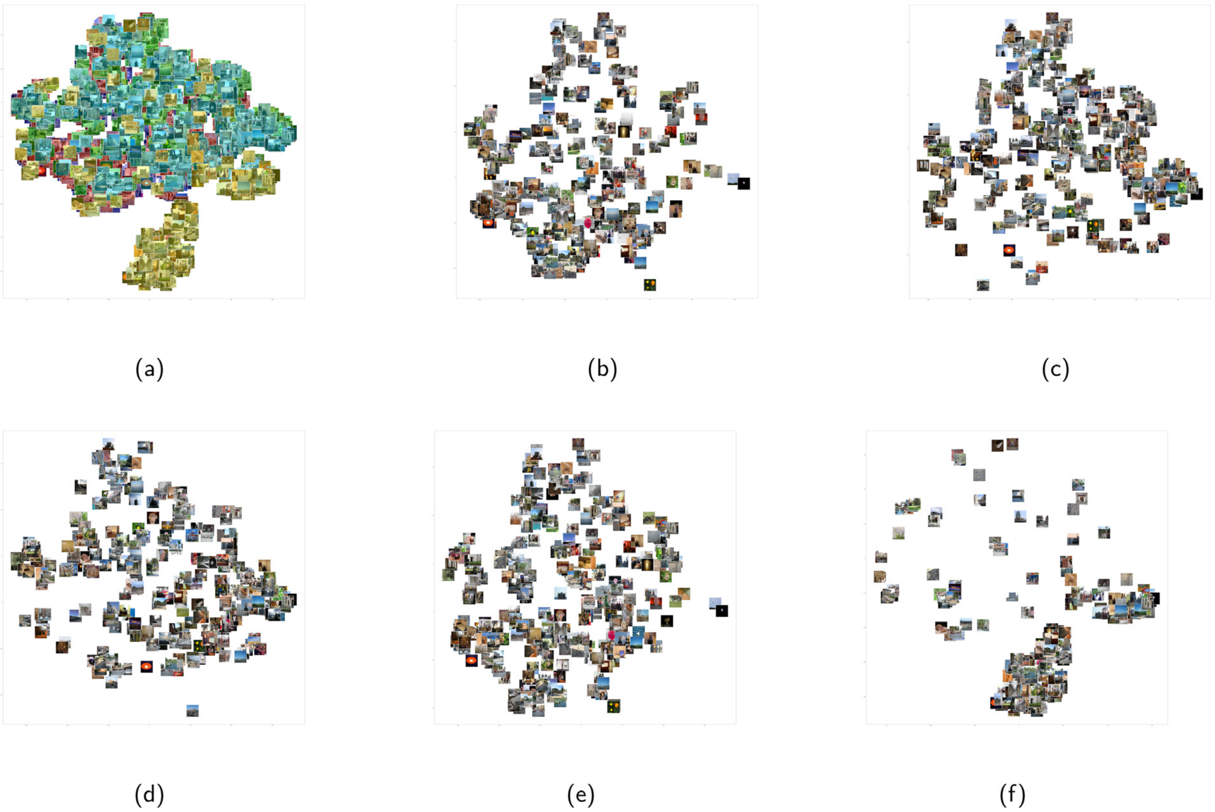


Fig. 7. t-sne embeddings in two dimensions of images based on their scores. (a) Aggregated representation of scores from 5 different losses. (b) Zoom on MLNET-MSE scores (blue). (c) Zoom on W-BCE ($w = 0.5$) scores (red). (d) Zoom on CC scores (green). (e) Zoom on DF-AE scores (cyan). (f) Zoom on LC-6 scores (yellow).

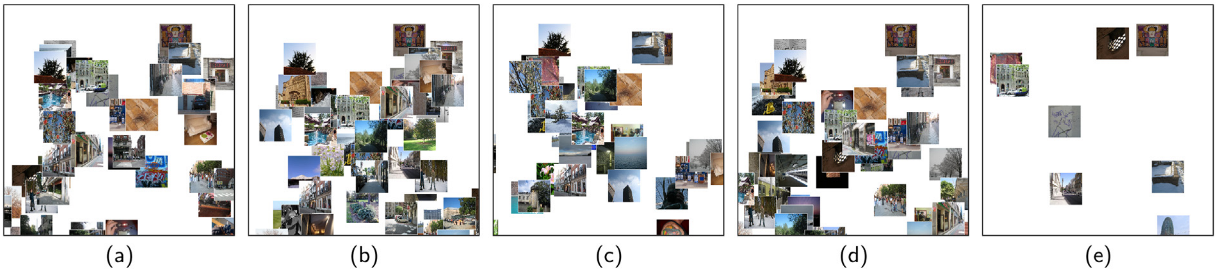


Fig. 8. Zoom on the top-part of the t-sne representation for each of the losses. (a) MLNET-MSE, (b) W-BCE ($w = 0.5$), (c) CC, (d) DF-AE, (e) LC-6.

Table 4
Performance of the loss functions after training on the SALICON dataset. Best performances are in bold. (AUC-B = AUC-Borji; AUC-J = AUC-Judd)

	CC ↑	SIM ↑	AUC-J ↑	AUC-B ↑	NSS ↑	EMD ↓	KL ↓
MSE	0.6204	0.4425	0.8269	0.8470	2.1223	2.1305	0.9472
BCE	0.6631	0.4529	0.8320	0.8496	2.2077	**2.0942**	0.8368
NSS	0.6729	0.4192	0.8375	0.8416	2.4081	3.0695	0.9343
DF-VGG	0.6002	0.4328	0.8461	0.8304	2.0257	3.2059	0.9264
LC-6	**0.7238**	**0.5620**	**0.8707**	**0.8442**	**2.3966**	2.1592	**0.7983**

using the original protocol, in order to verify the effectiveness of the proposed loss design. Using LC-6 give similar results for most metrics, but significantly improves the AUC-B score, for which the original Sam-ResNet was not among the best models. Such mild improvement could be explained by the fact that the Sam-ResNet network is originally trained using a combination of loss functions, LC-2.

5. Conclusion

In this paper, we introduced a deep neural network which purpose was to evaluate the impact of loss functions on the prediction capacity. We evaluated several well-known and commonly used losses, and introduced a new kind of loss function (a perceptual-based loss) that, to the best of our knowledge, has not been applied

Table 5
Performance of our model and Sam-ResNet, trained on the SALICON dataset with the LC-6 combination. Best performances are in bold. (AUC-B = AUC-Borji; AUC-J = AUC-Judd)

	CC ↑	SIM ↑	AUC-J ↑	AUC-B ↑	NSS ↑
Itti	0.28	0.35	0.71	0.71	0.79
Rare2012	0.46	0.43	0.78	0.79	1.40
GBVS	0.49	0.44	0.81	0.81	1.38
AWS	0.39	0.40	0.75	0.76	1.22
Sam-ResNet	0.74	**0.67**	0.87	0.81	**2.51**
SalGan	0.73	0.64	0.88	**0.86**	2.22
Sam-VGG	0.71	0.60	0.86	0.79	2.39
ShallowNet	0.63	0.51	0.84	0.83	1.87
DeepConvNet	0.67	0.55	0.85	0.85	1.94
Our model	0.72	0.56	0.87	0.84	2.40
(ranking)	*4/11*	*5/11*	*3/11*	*4/11*	*3/11*
Sam-ResNet (LC-6)	**0.74**	0.65	**0.88**	0.85	2.49
(ranking)	*1/11*	*2/11*	*1/11*	*3/11*	*2/11*

to saliency prediction. These new loss functions improve, at least partially, the performances of a deep saliency model, by bringing a different contribution to the loss aggregation. We showed that a simple linear combination of different losses can significantly improve over individual losses, especially when different types of loss functions are combined (pixel-based, distribution-based, perception-based, saliency-based).

More importantly we showed that this combination strategy generalizes well on different datasets and also with a different deep network architecture. Optimization on the coefficients of those linear combinations is also possible to obtain the best performances possible out of the combination. This approach could moreover easily be extended to other kinds of architectures, not necessarily based on convolutional neural networks.

Finally, one of the main idea that motivated our work was to highlight the importance of the choice of the loss function. We showed that a careful design of the loss function can significantly improve the performances of a model without increasing the number of trainable parameters.

The main recommendation that we would like to draw from this work would be to give a particular care to the loss function when designing a deep saliency model. We would recommend using a combination of several losses, especially losses with different dynamics, based on different characteristics: mixing one pixel-based loss with one distribution-based loss, several saliency-inspired losses, and a style-transfer loss, for instance. We find that using this approach allows a much more even distribution of the performances over all the metrics used to evaluate saliency models.

In this study, we did not include GAN-based loss functions, such as the one used in [41]. Since using such losses requires a different training process (*i.e.* adversarial training), it can be hard to compare them to more classical losses. The fact that it is a trained loss makes it also harder to interpret. However, such a comparison would definitely prove interesting. Extending this work to the dynamic domain would also prove of interest, as temporal-specific losses could be discussed and designed. Linear combination approaches have already been used with success, in models such as ACLNet [54]. For instance, the training protocol proposed in [1] could probably expect better performances using a saliency-specific loss rather than the euclidean distance.

CRediT authorship contribution statement

Alexandre Bruckert: Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Hamed R. Tavakoli:** Conceptualization, Writing - original draft, Methodology. **Zhi Liu:** Conceptualization, Writing - original draft. **Marc Christie:** Supervision, Writing - review & editing. **Olivier Le Meur:** Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Bak, A. Kocak, E. Erdem, A. Erdem, Spatio-temporal saliency networks for dynamic saliency prediction, IEEE Transactions on Multimedia 20 (2017) 1688–1698.

[2] A. Borji, Saliency prediction in the deep learning era: An empirical investigation, 2018. arXiv preprint arXiv:1810.03716.

[3] A. Borji, L. Itti, State-of-the-art in visual attention modeling, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 185–207.

[4] A. Borji, L. Itti, Cat 2000: A large scale fixation dataset for boosting saliency research, 2015, arXiv preprint arXiv:1505.03581.

[5] A. Borji, D.N. Sihite, L. Itti, Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study, IEEE Transactions on Image Processing 22 (2013) 55–69.

[6] A. Borji, H.R. Tavakoli, D.N. Sihite, L. Itti, Analysis of scores, datasets, and models in visual saliency prediction, IEEE International Conference on Computer Vision (2013) 921–928.

[7] N.D.B. Bruce, C. Catton, S. Janjic, A deeper look at saliency: Feature contrast, semantics, and beyond, IEEE Conference on Computer Vision and Pattern Recognition (2016) 516–524.

[8] N.D.B. Bruce, J.K. Tsotsos, Saliency based on information maximization, International Conference on Neural Information Processing Systems (2005) 155–162.

[9] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, A. Torralba, Mit saliency benchmark, 2015.

[10] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, F. Durand, Where should saliency models look next?, European Conference on Computer Vision (2016) 809–824.

[11] G. Chang, Y. Zhang, Y. Wang, An element sensitive saliency model with position prior learning for web pages, ICIAI (2018) 157–161.

[12] L. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, 2017. arXiv preprint arXiv:1706.05587.

[13] M. Cornia, L. Baraldi, G. Serra, R. Cucchiara, A deep multi-level network for saliency prediction, International Conference on Pattern Recognition (2016) 3488–3493.

[14] M. Cornia, L. Baraldi, G. Serra, R. Cucchiara, Predicting human eye fixations via an LSTM-based saliency attentive model, IEEE Transactions on Image Processing 27 (2018) 5142–5154.

[15] A. Garcia-Diaz, X.R. Fdez-Vidal, X.M. Pardo, R. Dosil, Saliency from hierarchical adaptation through decorrelation and variance normalization, Image and Vision Computing 30 (2012) 51–64.

[16] L. Gatys, A. Ecker, M. Bethge, A neural algorithm of artistic style, 2015. arXivpreprint arXiv:1508.06576.

[17] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, International Conference on Neural Information Processing Systems (2006) 545–552.

[18] S. He, H.R. Tavakoli, A. Borji, Y. Mi, N. Pugeault, Understanding and visualizing deep visual saliency models, 2019. arXiv preprint arXiv:1903.02501.

[19] X. Hou, J. Harel, C. Koch, Image signature: Highlighting sparse salient regions, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 194–201.

[20] X. Huang, C. Shen, X. Boix, Q. Zhao, Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks, IEEE International Conference on Computer Vision (2015) 262–270.

[21] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 1254–1259.

[22] S. Jetley, N. Murray, E. Vig, End-to-end saliency mapping via probability distribution prediction, IEEE Conference on Computer Vision and Pattern Recognition (2016) 5753–5761.

[23] S. Jia, EML-NET: an expandable multi-layer network for saliency prediction, 2018. CoRR abs/1805.01047. http://arxiv.org/abs/1805.01047, arXiv:1805.01047.

[24] M. Jiang, S. Huang, J. Duan, Q. Zhao, Salicon: Saliency in context, IEEE Conference on Computer Vision and Pattern Recognition (2015) 1072–1080.

[25] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, European Conference on Computer Vision (2016) 694–711.

[26] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, IEEE International Conference on Computer Vision (2009) 2106–2113.

[27] A. Kendall, R. Cipolla, Geometric loss functions for camera pose regression with deep learning, IEEE Conference on Computer Vision and Pattern Recognition (2017) 5974–5983.

[28] A. Kroner, M. Senden, K. Driessens, R. Goebel, Contextual encoder-decoder network for visual saliency prediction, 2019. arXivpreprint arXiv:1902.06634.

[29] M. Kummerer, L. Theis, M. Bethge, Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet, in: ICLR Workshop, 2015.

[30] M. Kümmerer, T. Wallis, M. Bethge, Information-theoretic model comparison unifies saliency metrics, Proceedings of the National Academy of Science 112 (2015) 16054–16059.

[31] M. Kümmerer, T. Wallis, M. Bethge, Deepgaze ii: Reading fixations from deep features trained on object recognition, 2016. arXiv preprint arXiv:1610.01563.

[32] M. Kümmerer, T. Wallis, M. Bethge, Saliency benchmarking made easy: Separating models, maps and metrics, IEEE Conference on Computer Vision and Pattern Recognition (2018) 798–814.

[33] M. Kummerer, T. Wallis, L.A. Gatys, M. Bethge, Understanding low- and high-level contributions to fixation prediction, IEEE International Conference on Computer Vision (2017) 4799–4808.

[34] O. Le Meur, T. Baccino, Methods for comparing scanpaths and saliency maps: strengths and weaknesses, Behavior Research Method 45 (2013) 251–266.

[35] O. Le Meur, A. Coutrot, Introducing context-dependent and spatially-variant viewing biases in saccadic models, Vision Research 121 (2016) 72–84.

[36] O. Le Meur, P. Le Callet, D. Barba, Predicting visual fixations on video based on low-level visual features, Vision Research 47 (2007) 2483–2498.

[37] O. Le Meur, P. Le Callet, D. Barba, D. Thoreau, A coherent computational approach to model bottom-up visual attention, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 802–817.

[38] O. Le Meur, Z. Liu, Saccadic model of eye movements for free-viewing condition, Vision Research 116 (2015) 152–164.

[39] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE International Conference on Computer Vision (2017) 2999–3007.

[40] N. Liu, J. Han, A deep spatial contextual long-term recurrent convolutional network for saliency detection, IEEE Transactions on Image Processing 27 (2018) 3264–3274.

[41] J. Pan, C. Canton, K. McGuinness, N.E. O'Connor, J. Torres, E. Sayrol, X. Giro-i Nieto, Salgan: Visual saliency prediction with generative adversarial networks, 2017. arXiv preprint arXiv:1701.01081.

[42] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, N.E. O'Connor, Shallow and deep convolutional networks for saliency prediction, IEEE Conference on Computer Vision and Pattern Recognition (2016) 598–606.

[43] R.J. Peters, A. Iyer, L. Itti, C. Koch, Components of bottom-up gaze allocation in natural images, Vision Research 45 (2005) 2397–2416.

[44] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, T. Dutoit, Saliency and human fixations: State-of-the-art and study of comparison metrics, IEEE International Conference on Computer Vision (ICCV) (2013) 1153–1160.

[45] C. Shen, Q. Zhao, Webpage saliency, IEEE European Conference on Computer Vision (2014) 33–46.

[46] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. arXiv preprint arXiv:1409.1556.

[47] H.R. Tavakoli, F. Ahmed, A. Borji, J. Laaksonen, Saliency revisited: Analysis of mouse movements versus fixations, IEEE Conference on Computer Vision and Pattern Recognition (2017) 6354–6362.

[48] H.R. Tavakoli, A. Borji, J. Laaksonen, E. Rahtu, Exploiting inter-image similarity and ensemble of extreme learners for fixation prediction using deep features, Neurocomputing 244 (2017) 10–18.

[49] L. Van Der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (2008) 2579–2605.

[50] E. Vig, M. Dorr, D. Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, IEEE Conference on Computer Vision and Pattern Recognition (2014) 2798–2805.

[51] W. Wang, J. Shen, Deep visual attention prediction, IEEE Transactions on Image Processing 27 (2017) 2368–2378.

[52] W. Wang, J. Shen, X. Dong, A. Borji, R. Yang, Inferring salient objects from human fixations, IEEE Transactions on Pattern Analysis and Machine (2019), Intelligence.

[53] W. Wang, J. Shen, L. Shao, Video salient object detection via fully convolutional networks, IEEE Transactions on Image Processing 27 (2018) 38–49.

[54] W. Wang, J. Shen, J. Xie, M. Cheng, H. Ling, A. Borji, Revisiting video saliency prediction in the deep learning era, IEEE Transactions on Pattern Analysis and Machine (2019), Intelligence.

[55] J. Zhang, S. Sclaroff, Saliency detection: A boolean map approach, IEEE International Conference on Computer Vision (2013) 153–160.

[56] Q. Zhao, C. Koch, Learning saliency-based visual attention: A review, Signal Processing 93 (2013) 1401–1407.

Alexandre Bruckert received his B.S. and M.S. degrees in applied mathematics, computer engineering and data sciences from Paris-Saclay University and École Nationale d'Informatique pour l'Industrie et l'Entreprise, in 2016 and 2018 respectively. He is currently pursuing a Pd.D. degree at the University of Rennes 1, Rennes, France. His research interests include machine learning, human vision analysis, and image/video processing.

Hamed R. Tavakoli received his B.S. and M.S. degrees in computer engineering, software and artificial intelligence, from Azad University, Mashhad Branch, Mashhad, Iran, in 2004 and 2008, respectively. He received his Ph.D. degree in computer science from the University of Oulu, Oulu, Finland, in 2014. From 2015 to 2019, he was a postdoctoral researcher at the Finnish Centre of Excellence in Computational Inference Research (COIN) and the Department of Computer Science, Aalto University, Espoo, Finland. He is currently working as Principal Scientist in machine learning at Nokia Technologies, Espoo, Finland. His research interests include machine learning and computer vision with emphasis on biologically inspired vision systems, visual attention, and bio-signal processing.

Zhi Liu received the B.E. and M.E. degrees from Tianjin University, Tianjin, China, and the Ph.D. degree from Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai, China, in 1999, 2002, and 2005, respectively. He is currently a Professor with the School of Communication and Information Engineering, Shanghai University, Shanghai, China. From Aug. 2012 to Aug. 2014, he was a Visiting Researcher with the SIROCCO Team, IRISA/INRIA-Rennes, France, with the support by EUFP7 Marie Curie Actions. He has published more than 170 refereed technical papers in international journals and conferences. His research interests include image/video processing, machine learning, computer vision and multimedia communication. He was a TPC member/session chair in ICIP 2017, PCM 2016, VCIP 2016, ICME 2014, WIAMIS 2013, etc. He co-organized special sessions on visual attention, saliency models, and applications at WIAMIS 2013 and ICME 2014. He is an area editor of Signal Processing: Image Communication and served as a guest editor for the special issue on Recent Advances in Saliency Models, Applications and Evaluations in Signal Processing: Image Communication. He is a senior member of IEEE.

Marc Christie received his Ph.D. degree from the University of Nantes, Nantes, France, in 2003. He is currently an associate professor at University of Rennes 1. His focus is on virtual cinematography, which is the application of real cinematography techniques to virtual 3D environments. His research covers a wide range of challenges, such as extracting cinematographic features from real movies or learning elements of style (shots/transitions/editing patterns), understanding the relations between the orchestration of low visual features (such as image saliency) and the intended cognitive and emotional effects on the audience, but also proposing generative approaches in which camera angles/trajectories and cuts can be generated automatically or controlled interactively. Outputs from this research have been applied to the automated orchestration of multiple cinematographic drones, or exploited in e-sports to create cinematographic sequences from users' playing sessions.

Olivier Le Meur received his Ph.D. degree from the University of Nantes, Nantes, France, in 2005. From 1999 to 2009, he has worked in the media and broadcasting industry. In 2003 he joined the research center of Thomson-Technicolor at Rennes, France, where he supervised a research project concerning the modelling of the human visual attention. Since 2009 he has been an associate professor for image processing at the University of Rennes 1, Rennes, France. In his research team, PERCEPT, his research interests are dealing with the understanding and the modelling of the human visual attention. More specifically, Dr. Le Meur aims to design computational models for simulating the gaze deployment of human. He is also focussing on saliency-based applications, such as objective assessment of video quality, retargeting and image editing.