SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks

Xun Huang[1,2†] Chengyao Shen[1] Xavier Boix[1,3] Qi Zhao[1*]

[1]Department of Electrical and Computer Engineering, National University of Singapore
[2]School of Computer Science and Engineering, Beihang University
[3]CBMM, Massachusetts Institute of Technology

xunhuang1995@gmail.com scyscyao@gmail.com elexbb@nus.edu.sg eleqiz@nus.edu.sg

Abstract

Saliency in Context (SALICON) is an ongoing effort that aims at understanding and predicting visual attention. Conventional saliency models typically rely on low-level image statistics to predict human fixations. While these models perform significantly better than chance, there is still a large gap between model prediction and human behavior. This gap is largely due to the limited capability of models in predicting eye fixations with strong semantic content, the so-called semantic gap. This paper presents a focused study to narrow the semantic gap with an architecture based on Deep Neural Network (DNN). It leverages the representational power of high-level semantics encoded in DNNs pretrained for object recognition. Two key components are fine-tuning the DNNs with an objective function based on the saliency evaluation metrics, and integrating information at different image scales. We compare our method with 14 *saliency models on* 6 *public eye tracking benchmark datasets. Results demonstrate that our DNNs can automatically learn features for saliency prediction that surpass by a big margin the state-of-the-art. In addition, our model ranks top to date under all seven metrics on the MIT300 challenge set.*

1. Introduction

Saliency models predict the probability distribution of the location of the eye fixations over the image, *i.e.* the saliency map. Emulating the way that human observers look at an image has attracted much interest, since it may give new insights about the human attentional mechanisms, and allow for new artificial intelligence applications.

The pioneering theory of feature integration served as the basis for many of the initial saliency models [37]. The seminal works by Koch and Ullman [22] and Itti *et al.* [18] introduced the first computational architecture of saliency prediction based on the feature integration theory. Since then, numerous computational models have been proposed. For example, Harel *et al.* [14] extracted multi-scale low-level features including intensity, color and orientation, and predicted saliency based on graph algorithms. Bruce and Tsotsos [2] represented images with Gabor-like features learned from independent component analysis and estimated saliency as self-information. The recent Boolean Map Saliency (BMS) model by Zhang and Sclaroff [42] used color as features and computed saliency based on Boolean maps that are generated by randomly thresholding feature maps.

While effective, it is widely accepted that these models might not be complete in modeling visual attention due to the lack of features to represent semantic objects of interest [17]. In fact, object detectors have been shown to play an important role in improving saliency prediction [40], and several computational models have successfully incorporated object detectors into saliency models [4, 21, 44]. However, these detectors are specifically trained for each category, which makes it difficult to scale. This begs the question whether the models can be designed to automatically learn the cues from the raw images.

Deep Neural Networks (DNNs) allow the automatic learning of image representations, and recently achieved astonishing results in many computer vision tasks, *e.g.* [23, 10]. In saliency prediction, the multi-layer sparse network [33] and Ensemble of Deep Networks (eDN) [39] are both early architectures that automatically learn the representations for saliency prediction. Since the amount of data

[†] This work was done when Xun Huang was a visiting student at National University of Singapore

[*] Corresponding author.

available to learn saliency prediction is scarce, the complexity of the deep architectures cannot be easily scaled to outperform current state-of-the-art. In this case, one strategy is to leverage the big amount of data samples from other domains like object recognition [6]. Kummerer *et al.* [24] initially transferred features directly from a DNN for object recognition, and showed promising results with a model that they called Deep Gaze. Depending on the evaluation metric, Deep Gaze either marginally outperforms previous methods, or obtains an accuracy in the same range as eDN [39] and BMS [42], among others.

In this paper, we investigate saliency prediction using the representational power of the semantic content in DNNs pretrained in ImageNet [6]. To address the difference in the task objective between saliency prediction and object recognition, we use saliency evaluation metrics as the objective to fine-tune the DNNs. Also, since selective attention may happen at different resolution, we incorporate information at multiple-scales.

In a series of experiments, we evaluate our architecture over 6 standard benchmark datasets, namely Object and Semantic Images and Eye-tracking (OSIE) [40], MIT1003 [21], NUS Eye Fixation (NUSEF) [29], Fixations in Faces (FIFA) [4], PASCAL-S [25] and Toronto [2], based on three common recognition networks: AlexNet [23], VGG-16 [34] and GoogLeNet [35]. The results demonstrate that our architectures can surpass the state-of-the-art accuracy of saliency prediction by a big margin in all the tested benchmarks. Our model achieves the best performance to date under all evaluation metrics on the MIT300 benchmark [3]. These results suggest that the information encoded in DNNs is much more informative than current hand-crafted features to predict saliency. This is particularly effective in predicting gazes with semantic objects, attributed to the semantics encoded in DNNs.

2. DNNs for Saliency Prediction

In this Section, we introduce our method to use DNNs in saliency prediction. A DNN is a feedforward neural network with constrained connections between layers, that take the form of convolutions or spatial pooling, besides other possible non-linearities, *e.g.* [23]. The sample complexity of a DNN can be controlled by varying the depth of the network and the number of neurons at each layer. Increasing the sample complexity comes with the risk of overfitting when the number of training samples does not scale accordingly.

Thus, learning a DNN only from saliency maps may be difficult, since even the largest saliency dataset does not contain millions of training images, as in the object recognition datasets used to successfully train large DNNs. A well-known result is that features based on object detectors can be useful to predict saliency maps [4, 7, 21, 40, 44]. Thus, DNNs pretrained for object recognition, that are known to encode powerful semantics features, might be useful as well for saliency prediction. We introduce an architecture that integrates the saliency prediction to a DNN pretrained for object recognition. This allows to learn with back-propagation the parameters of the pretrained DNN, by optimizing a saliency evaluation metric. This yields a substantial improvement of the performance over previous works that also use DNNs, since eDN does not exploit the advantages of pretraining [39], and Deep Gaze only uses the neural responses of the DNN without adapting the DNN to saliency prediction [24]. We illustrate the overall structure of the DNN in Fig. 1. In the following, we first introduce our architecture, and then, the learning of the parameters.

2.1. Architecture of the DNN

We build our architecture from one of the popular DNN architectures for object recognition [6]. The DNNs we use are AlexNet [23], VGG-16 [34] and GoogLeNet [35]. These DNNs contain several max-pooling layers, and cascades of convolutional and non-linear layers between pooling layers. The total number of layers depends on each network. Our architecture is based on one of these DNNs in the context of saliency prediction (see Fig. 1).

Let $\mathbf{Y}_k$ be a three-dimensional table that contains the responses of the neurons of the DNN at layer k. $\mathbf{Y}_k$ has size $m_k \times n_k \times d_k$, that depends on each layer. The first two dimensions of the table index the spatial location of the center of the receptive field of the neuron, and the third one indexes the templates for which the neuron is tuned.

It has been shown that the neural responses at higher layers in the hierarchy encode more meaningful semantic representations than at lower layers [41]. The last layers of the DNN transform the neural responses to classification scores that have little spatial information. Since saliency prediction aims at localizing the salient regions, the neural responses in mid and high layers might be more informative than the responses at the last layer.

We found that a good compromise between semantic representation and spatial information for saliency prediction is to use the last convolution layer, in any of the three DNNs we use. We denote the neural responses of this layer as $\mathbf{Y}_c$. The number of templates at this layer is $d_c = 256$ for AlexNet, $d_c = 512$ for VGG-16, and $d_c = 832$ for GoogLeNet. To have neural responses in the image borders, we add 0 padding in the convolution layers in all the DNN. This yields a spatial resolution for $\mathbf{Y}_c$ of $m_c \times n_c = 37 \times 50$, for an input image of 600×800. The layers after $\mathbf{Y}_c$ in the DNN including all fully-connected layers are not used for saliency prediction and we remove them from our architecture. In other words, we use DNN in a fully convolutional way similar to [27].

The neural responses of $\mathbf{Y}_c$ tend to detect parts or pat-

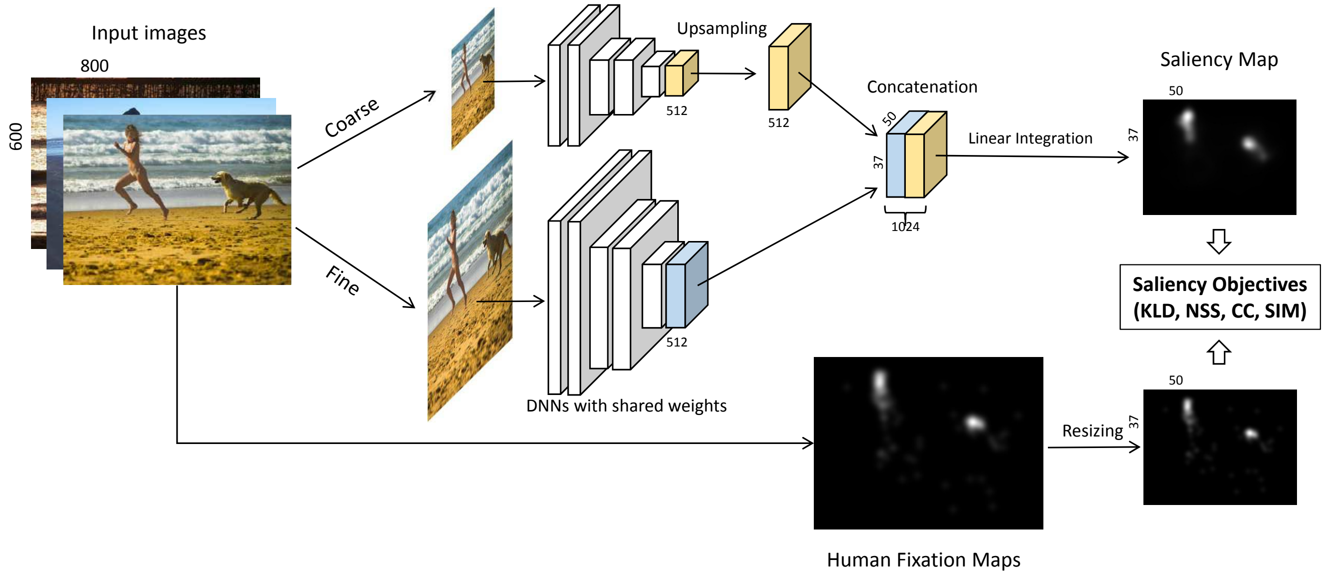


Figure 1: *Learning of the DNN architecture for saliency prediction.* The architecture consists of a DNN applied at two different image scales. Readers are referred to [23, 34, 35] to see the detailed model structures that we use. The last convolutional layer in the pretrained network feeds a randomly initialized convolutional layer with one filter that detects the salient regions. The parameters are then learnt end-to-end with back-propagation. We use objective functions to optimize some common saliency evaluation metrics.

terns in objects that are useful for object recognition. To use $\mathbf{Y}_c$ for saliency prediction, we add one convolutional layer after $\mathbf{Y}_c$. This convolutional layer has only one filter, that detects whether the responses in $\mathbf{Y}_c$ correspond to a salient region or not. We denote the result of convolving $\mathbf{Y}_c$ with the saliency detector filter as $\mathbf{Y}_s$, and it encodes the saliency prediction information at a resolution of $m_s \times n_s$. The filter of the convolutional layer for saliency prediction is of size 1×1. This yields the same spatial resolution as $\mathbf{Y}_c$, *i.e.* $m_s \times n_s = 37 \times 50$. Increasing this size does not improve the accuracy because the receptive field of the neurons of $\mathbf{Y}_c$ capture enough context for saliency prediction. The 1×1 filter of the convolutional layer may select and discard which objects parts or patterns detected by the DNN are useful for saliency prediction. In the experiments, we visualize the patterns in the DNN that are used for saliency prediction.

Finally, to obtain the saliency map, we resize the responses of $\mathbf{Y}_s$ with a linear interpolation to match the image size, and we scale it to take values between 0 and 1. Predicting the saliency map at the spatial resolution of $\mathbf{Y}_s$ is effective, but also, it reduces the computational burden at training time because the number of neural responses to process from $\mathbf{Y}_s$ is much lower than the image size.

A common practice of saliency prediction models is to use information at multiple image scales to improve the accuracy, *e.g.* [1, 8, 11, 14, 16, 30]. We now extend the DNN to capture multi-scale information.

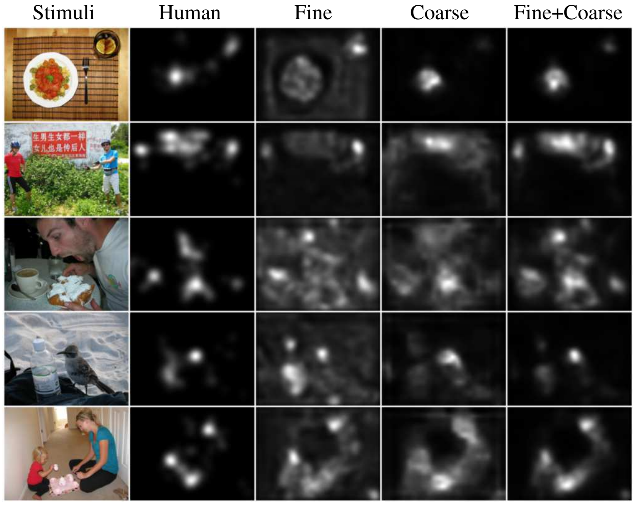


Figure 2: *Saliency prediction with multiple image scales.* The multi-scale DNN can detect salient regions of different sizes. In the fine scale, the DNN detects salient regions of small size, while in the coarse scale, the center of large salient regions stands out. Images are from OSIE dataset.

Extension to Multi-scale. We use the input image at different scales obtained by downsampling the image. Each scale is processed by a DNN, and the neural responses of all DNNs are used to predict the saliency map with the convolutional layer we previously introduced. We use $\mathbf{I}$ to denote the input image, and $\mathbf{I}'$ the image $\mathbf{I}$ downsampled by half. Although in principle more scales can be added, we find adding more scales does not further improve performance. We define $\mathbf{Y}'_c$ as the neural response of the image at the coarse scale, $\mathbf{I}'$. Thus, we have one DNN to generate $\mathbf{Y}_c$ (the blue cuboid in Figure 1) and another to generate $\mathbf{Y}'_c$ (the yellow cuboid in Figure 1). These two DNNs share the same filters, and hence, the neurons are tuned to detect the same patterns but at a different scale.

Note that $\mathbf{Y}'_c$ has half the spatial resolution of $\mathbf{Y}_c$, *i.e.* $(m_c/2) \times (n_c/2) \times d_c$. To combine the responses of $\mathbf{Y}_c$ and $\mathbf{Y}'_c$ for saliency prediction, we upsample $\mathbf{Y}'_c$ with a linear interpolation to match the same spatial resolution as $\mathbf{Y}_c$. The combination of the responses of both $\mathbf{Y}_c$ and the resized $\mathbf{Y}'_c$ yield a number of neural responses equal to $m_c \times n_c \times (2 \cdot d_c)$. There are two neurons tuned for the same pattern that act at two different scales. Finally, these neural responses feed the 1×1 convolutional layer that generates the saliency map $\mathbf{Y}_s$.

The effect of multi-scale is qualitatively illustrated in Fig. 2. It can be seen that by combining features from both scales, our saliency map correctly highlights salient regions of different sizes.

2.2. Learning with Saliency Evaluation Objectives

In our architecture we have integrated the saliency prediction into the DNN, and the parameters can be learnt with back-propagation [31]. We initialize the parameters of the DNN to the pretrained parameters in ImageNet [6], and then, we learn end-to-end the parameters of all the architecture. In the experiments, we show that adapting the features of the DNN to saliency prediction yields significant improvement over directly using the off-the-shelf features.

Another advantage of our learning scheme is that back-propagation can be used to optimize the saliency evaluation metric. Previous works use objective functions that do not directly correspond to the evaluation metric. Typically, a Support Vector Machine (SVM) is used [21, 33, 39, 40]. Pixels in the saliency map are evaluated using a ground-truth label that indicates whether the pixel is salient or non-salient. Back-propagation allows directly optimizing the saliency metrics, which may better guide the learning towards the goal of saliency prediction.

There is a plethora of saliency evaluation metrics available that are complementary to each other. The Area Under the Curve (AUC) [13] is the area under a curve of true positive rate versus false positive rate for different thresholds on the saliency map, and the shuffled-AUC (sAUC) [36] alleviates the effects of center bias in the AUC score. The Normalized Scanpath Saliency (NSS) [28] computes the average value at all fixations in a normalized saliency map. Similarity (Sim) [20] calculates the sum of minimum values of saliency distribution and fixation distribution at each point. Finally, the saliency map can be compared with the human fixation map with the Linear Correlation Coefficient (CC) [19] and the Kullback-Leibler divergence (KLD) [36].

We use 4 evaluation metrics as objective functions of the back-propagation, which are NSS, CC, KLD and Sim. These evaluation metrics have a derivative that can be used by the gradient descend of back-propagation. We do not use AUC and sAUC as objective for back-propagation since the derivative is more involved than for the other evaluation metrics. In the experiments we show that the objective function of KLD achieves a good compromise in all evaluation metrics we use.

3. Experiments

We implement our models within the MatConvNet [38] framework. After describing the experimental settings, we analyze the different components of our architecture, and compare our method with the state-of-the-art.

3.1. Experimental Setup

Datasets. We use 6 popular datasets that differ in terms of image content and experimental settings to ensure a comprehensive comparison. The descriptions of these datasets are listed below:

- *OSIE* [40]: This dataset contains 700 images with annotated objects and attributes content. A total number of 15 subjects free-viewed the images for 3 seconds.
- *MIT1003* [21]: This dataset includes 1003 images with everyday indoor and outdoor scenes. All images are presented to 15 observers for 3 seconds.
- *NUSEF* [29]: Most images contain emotionally affective scenes and objects. We use 714 images available, each viewed on average by 25 subjects for 5 seconds.
- *FIFA* [4]: Most images in this dataset contain faces as dominant objects. 8 subjects free-viewed a total number of 200 images for 2 seconds.
- *PASCAL-S* [25]: This recent dataset contains 850 images from the PASCAL VOC 2010 dataset [9] with eye fixations from 8 viewers, as well as salient object labeling.
- *Toronto* [2]: It contains 120 images and fixation data from 20 viewers. A large proportion of the images do not contain salient objects that attract attention, which makes it interesting to evaluate the DNN pretrained in object recognition.

Evaluation Metrics. We evaluate the saliency map with the metrics previously introduced in Sec. 2.2. For most experiments, we use the implementation of sAUC in [1] to compare the model performance.

A standard practice for evaluation is to Gaussian blur the saliency maps and find the optimal blurring of the saliency map for each model. We use a Gaussian kernel with a standard deviation from 0 to 2 degrees of visual angle with a step size of 0.25. The evaluation scores we report are obtained as the highest scores obtained with blurring. The ground-truth fixation maps are constructed by convolving a Gaussian kernel with a standard deviation of one degree of visual angle [7] over the fixation locations of all subjects.

Training and Testing. We use OSIE [40] dataset to train the parameters of our architecture, even if we test in another dataset. We observed that learning in this dataset leads to a better performance than in MIT1003 dataset. We divide the OSIE dataset into 450 training images, 50 validation images and 200 test images. We learn the parameters of the DNN on the training images with stochastic gradient descent with momentum. We use only one image in each iteration since we did not get improvement from mini-batch. We use momentum of 0.9 and a weight decay of 0.0005. The learning rate is fixed at 10^{-5} for AlexNet, VGG-16, and 5×10^{-5} for GoogLeNet. We stop learning when the objective function does not improve on the validation set. We trained the network in a single NVIDIA Titan GPU, and it took approximately between 1 hour to 2 hours depending on the network used. We tried to augment the training data by flipped and rotated images, but this did not yield noticeable improvement. During testing, each input image is resized to two scales (800 pixels and 400 pixels in its largest dimension). It takes 0.27s to predict one saliency map with GPU.

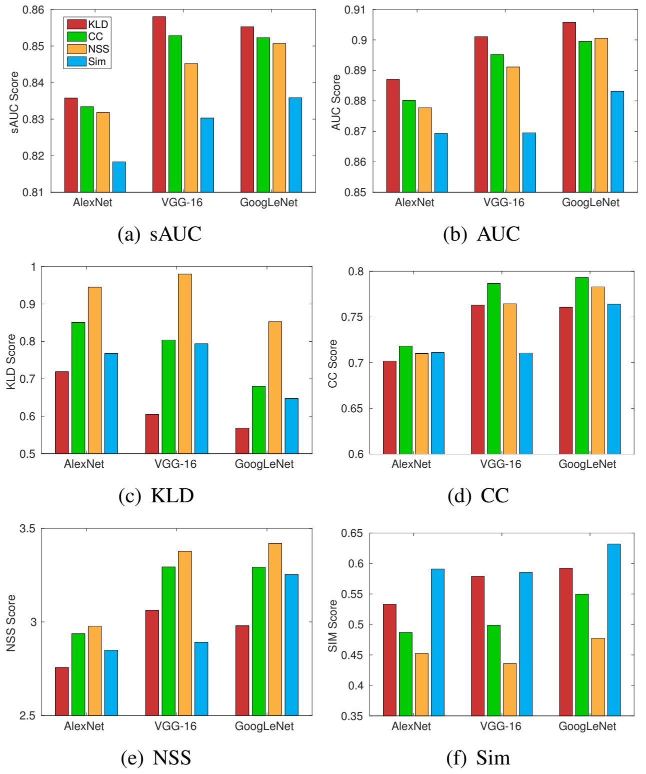
3.2. Analysis of the Architecture

We analyze the components of our architecture, in the OSIE dataset.

Learning Objective. Recall that we can learn the DNN using different saliency evaluation metrics, *i.e.* KLD, CC, NSS, Sim (Sec. 2.2). According to Fig. 3, a model optimized for a particular metric generally achieves the best score in that metric among models using the same DNN. In the rest of the experiments, we use the parameters learnt with the KLD objective, since the KLD obtains a good compromise among all the evaluation metrics, and the best performance for the sAUC and AUC scores, which are used in most benchmarks.

Multi-scale. In Fig. 4, we compare the results for architectures that use one or the two scales, and when we do not do fine-tuning (we fix all weights in the DNN and only train the last convolutional layer with back-propagation). Results show that using information at multiple scales boosts the performance. This performance gain is more evident for AlexNet and VGG-16 model, but much less for GoogLeNet, possibly because GoogLeNet already uses multi-scale information in its inception layers [35].

Fine-tuning. Also in Fig. 4, we can see that fine-tuning the DNN achieves superior performance than without fine-tuning, which shows the advantage of adapting the DNN to saliency prediction. We also report results combining the DNN features with a linear SVM using the common procedure in the literature [21]. The SVM baseline is learnt in the same dataset as our architecture, and uses the features of the DNN at the fine scale. We can see in Fig. 4, that the fine scale without fine-tuning already improves over the SVM baseline. This shows the effectiveness of optimizing saliency metrics rather than the SVM objective.



(a) sAUC

(b) AUC

(c) KLD

(d) CC

(e) NSS

(f) Sim

Figure 3: *Comparison of the learning objectives.* Each plot corresponds to a different evaluation metric, *i.e.* sAUC, AUC, KLD, CC, NSS and Sim. The four color bars are DNNs trained with KLD, CC, NSS and Sim evaluation metric objectives. Lower KLD means better performance, and for other metrics a higher score means better performance.

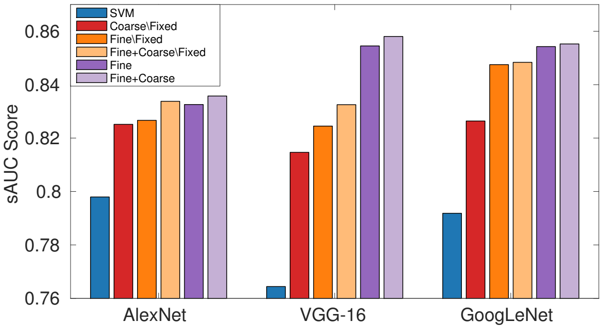


Figure 4: *Results on multi-scale and fine-tuning.* The sAUC scores for DNNs with one or two scales, and with and without fine-tuning are reported. The SVM is uses the DNNs features at the fine scale. Fixed means without fine-tuning.

Note that before fine-tuning, GoogLeNet performs bet-

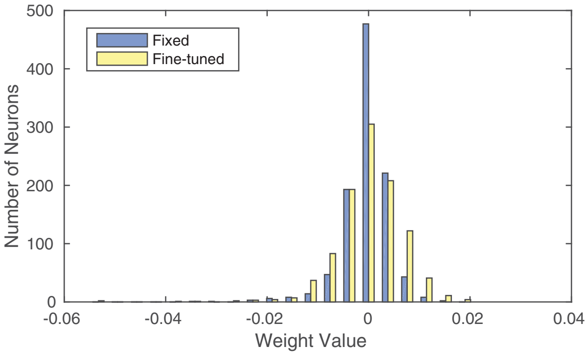


Figure 5: *Histogram of the weights of the convolutional filter for saliency prediction.* The histogram is compared before the fine-tuning, and after. Larger absolute weight value means bigger contribution to saliency prediction.

ter than the other DNN, but after the fine-tuning, the performance of VGG-16 dramatically improves and becomes the best model. Since VGG-16 is the DNN that performs best in this task, we use it in the rest of the experiments.

Visualization. We analyze the changes produced by the fine-tuning to the DNN representations. In Fig. 5, we show the effect of fine-tuning to the histogram of the filter weights in the last convolutional layer. Before the fine-tuning, the filter discards many features of the DNN by setting their weight to 0, and after the fine-tuning, more features are selected. This suggests that fine-tuning effectively adapts the DNN representation for saliency prediction. In order to visualize the change in the neural responses after the fine-tuning, we use the neurons that are selected by the convolutional layer with the 3 highest and lowest weight values, which correspond to the neurons that the filter considers more informative for saliency prediction. Note that the lowest weight values are negative, and encourage suppressing the non-salient regions. The visualization is done by displaying the receptive fields in the image that made the neurons respond more strongly in the training set, as in [41]. In Fig. 6 and 7, we visualize the change produced by the fine-tuning on the 3 neurons with highest and lowest wight value before the fine-tuning, respectively. Also observe that the neurons with highest weight value remain very similar after the fine-tuning (Fig. 6), since they already encode semantic content that may be encoded in the DNN for object recognition. We can see the opposite for the lowest weights, since the same neuron responds to different patterns after the fine-tuning (Fig. 7). The patterns after the fine-tuning have the appearance of non-salient regions without any clear semantic meaning, which may explain why the DNN for object recognition do not encode them initially.

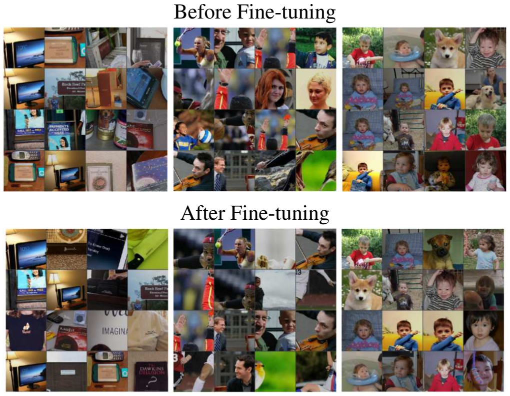


Figure 6: *Visualization of the three neurons with* highest *weight before the fine-tuning.* Each neuron is visualized by displaying the receptive fields of the images that produced the highest response of the neuron. Each column visualizes the same neuron before and after fine-tuning.

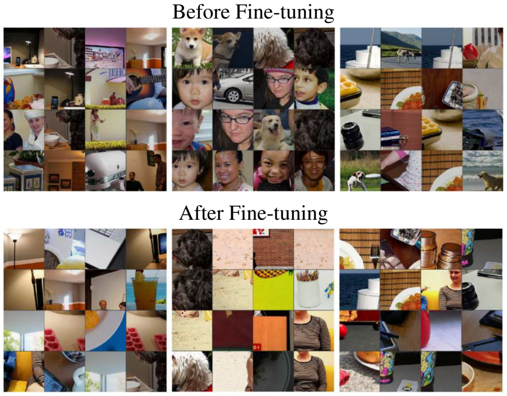
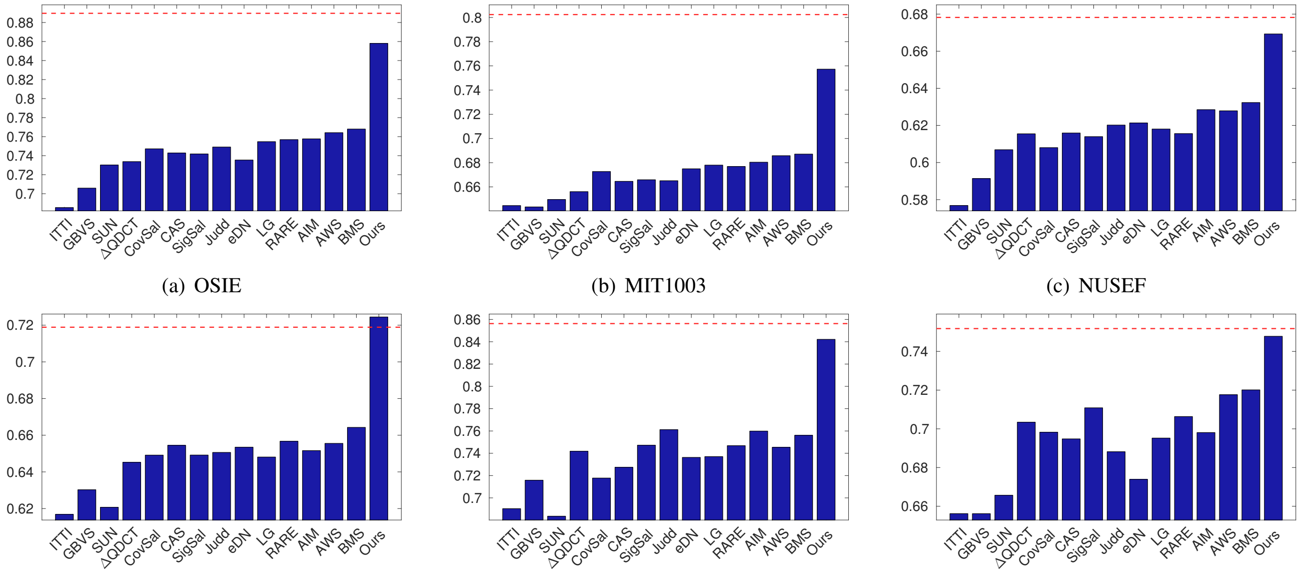


Figure 7: *Visualization of the three neurons with* lowest *weight before the fine-tuning.* The same visualization as in Fig. 6. Each column visualizes the same neuron before and after fine-tuning.

3.3. Comparison with State-of-the-art

We now compare our best-performed model (multi-scale VGG-16, learnt with KLD as objective) with the state-of-the-art in different datasets.

Comparison on Public Eye Tracking Datasets. We compare our method with 14 saliency prediction models, which are BMS [42], Adaptive Whitening Saliency (AWS) [11], Attention based on Information Maximization (AIM) [2], RARE [30], Local and Global saliency (LG) [1], eDN [39], Judd's model [21], image Signature Saliency (SigSal) [15], Context-Aware Saliency (CAS) [12], Covariance-based Saliency (CovSal) [8], multi-scale Quaternion DCT (ΔQDCT) [32], Saliency Using Natural statistics (SUN) [43], Graph-Based Visual Saliency (GBVS) [14], and Itti's model (ITTI) [16]



(a) OSIE

(b) MIT1003

(c) NUSEF

(d) PASCAL-S

(e) FIFA

(f) Toronto

Figure 8: *Comparison to state-of-the-art in* 6 *datasets.* We report the sAUC scores of different saliency models under optimal blurring on 6 datasets. The red line indicates the Human Inter-Observer (IO) score. The models are arranged from left to right in ascending order of average sAUC score on all datasets.

on 6 datasets. Many of them are recently published models that have shown top performance on saliency evaluation datasets. We use the recommended parameter settings provided by the authors. For models with explicit center-bias, we disable their center-bias for a fair comparison, and this improves their sAUC score. We also report the Human Inter-Observer (IO) score [1]. For a given image the fixation maps from all subjects except one are used to predict the eye fixations of the discarded subject. The score is averaged over all images and all subjects to give the IO score.

Fig. 8 shows that our model outperforms all the methods in the 6 datasets by a substantial margin. BMS achieves the second-best performance on all datasets except FIFA. In the datasets with more objects, the gap between our results and the second best is more remarkable. For instance, in the popular MIT1003 dataset, our method achieves 0.76 sAUC while BMS achieves 0.69. In the Toronto dataset, since the images have fewer objects that attract eye fixations, the performance gap between our prediction and others is relatively smaller, but still noticeable.

We also compare our model with a very recently published saliency model based on DNN [26]. We compare the saliency maps in MIT1003 and Toronto, since these are the only two datasets that we both use the same testing images. In MIT1003, the new model obtains a sAUC score of 0.71, which outperforms BMS (0.69) but is much lower than the sAUC score of our model (0.76). In Toronto, the new model obtains an sAUC score similar to BMS (0.72), while our model obtains 0.75. In addition, our inference time (0.27s) is much faster than theirs (14s) even our network is larger, due to the efficiency of our fully convolutional architecture compared to their patch-by-patch scanning strategy.

Evaluation Metric	Gauss	S-o-a	Ours	Infinite Humans	Relative Advance
AUC-Judd	0.78	0.84 Deep Gaze	0.87	0.91	42.9%
AUC-Borji	0.77	0.83 Deep Gaze	0.85	0.87	50.0%
sAUC	0.51	0.68 AWS	0.74	0.80	50.0%
NSS	0.92	1.41 BMS	2.12	3.18	40.1%
CC	0.38	0.55 BMS	0.74	1	42.2%
Similarity	0.39	0.51 BMS	0.60	1	18.4%
EMD	4.81	3.33 OS	2.62	0	21.3%

Table 1: *Results in the MIT300 Online Benchmark.* 7 different evaluation metrics are used to compare the results of state-of-the-art and our model.

Results on MIT300 Online Benchmark. We also submitted our results to the MIT300 benchmark [3]. This benchmark contains 300 natural images with eye fixations from 39 subjects. The eye fixations are not public available to prevent fitting to the test set. To date, 41 saliency models are compared using 7 standard evaluation metrics. Details of these metrics and comparisons can be found in [3]. In Table 1, we report our results, together with previous state-of-the-art for each metric, and two baselines, a Gaussian centered in the image, and the Infinite Humans score which is the maximum achievable score computed by using infinite observers to predict fixations from a different set of infinite observers. Since infinite observers are not available in practice, the benchmark team [20] use extrapolation to find the value of the score as a limit as

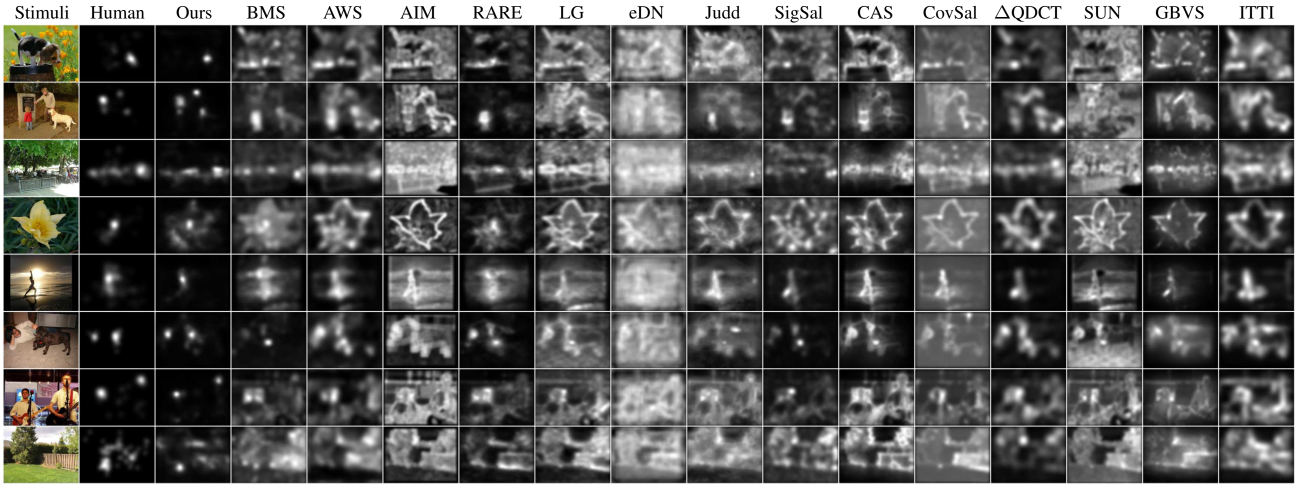


Figure 9: *Qualitative Results.* We compare our results with 14 saliency prediction models, namely BMS [42], AWS [11], AIM [2], RARE [30], LG [1], eDN [39], Judd [21], SigSal [15], CAS [12], CovSal [8], ΔQDCT [32], SUN [43], GBVS [14], and ITTI [16]. Our saliency maps are very localized in the salient regions compared to the rest of the methods.

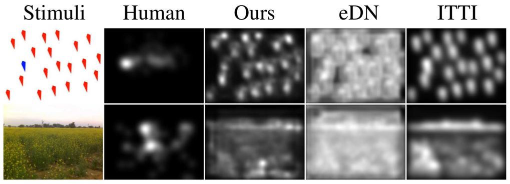


Figure 10: *Example of failure modes.* The first row shows a synthetic image and the second row shows an image with plain natural scenes. Images are selected from MIT1003 and Toronto dataset.

the number of observers goes to infinity. Before our results, it was unclear which method was performing best since BMS [42] ranked top for NSS, CC and Similarity, Deep Gaze [24] ranked top for AUC-Judd and AUC-Borji, AWS [11] for sAUC, and Outlier Saliency (OS) [5] for Earth Mover's Distance (EMD) [3]. Our model outperforms previous state-of-the-art under all metrics, without the need of center bias. To give a clear idea of how much our results relatively advance the state-of-the-art accuracy to perfect prediction, we report the improvement as $(S_{\text{ours}} - S_{\text{state-of-the-art}})/(S_{\text{human}} - S_{\text{state-of-the-art}})$. Thus, a 50% improvement means that our results are half way between current state-of-the-art and perfection.

Qualitative Results. In Fig. 9, we compare the saliency maps of all the models. Our method can effectively detect salient regions with semantic content like faces (row 2, 6 and 7), human (row 3 and 5), animals (row 1, 2 and 6) and text (row 2). When there is no semantic object that attracts fixations (row 8), our model can still perform reasonably well, indicating that some low-level information may be captured by our DNN. Observe that our architecture can detect salient regions in different sizes (row 1, 4 and 8). Also, note that our saliency maps are very localized in the salient regions compared with other methods, even when images have cluttered backgrounds (row 1 and 3).

Limitations. Fig. 10 shows some typical failure cases. The first row shows that our model does not perform well in a synthetic image, while models like ITTI that are based entirely on low-level cues may perform a bit better. The second row shows that when there is no explicit object in the image that may attract attention, the eye fixations tend to be biased toward image center, which our model fails to predict. Note that eDN, which is also based on DNN, does not predict fixations better than our model on these images.

4. Conclusions

Recent studies have suggested the importance of semantic information in predicting human fixations. To reduce the semantic gap between model prediction and human behavior, we re-architect DNNs for object recognition to the task of saliency prediction. We fine-tune the network with saliency metric as an objective function, and use information at multiple scales. This leads to a saliency prediction accuracy that significantly outperforms the state-of-the-art.

Acknowledgment

This research was supported by the Singapore Ministry of Education Academic Research Fund Tier 2 (No.R-263-000-B32-112) and the Defense Innovative Research Programme (No. 9014100596).

References

[1] A. Borji and L. Itti. Exploiting local and global patch rarities for saliency detection. In *CVPR*, 2012.

[2] N. Bruce and J. Tsotsos. Saliency based on information maximization. In *NIPS*, 2005.

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

[4] M. Cerf, J. Harel, W. Einhäuser, and C. Koch. Predicting human gaze using low-level saliency combined with face detection. In *NIPS*, 2008.

[5] C. Chen, H. Tang, Z. Lyu, H. Liang, J. Shang, and M. Serem. Saliency modeling via outlier detection. *JEI*, 2014.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[7] W. Einhäuser, M. Spain, and P. Perona. Objects predict fixations better than early saliency. *JoV*, 2008.

[8] E. Erdem and A. Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *JoV*, 2013.

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013.

[11] A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil. Saliency from hierarchical adaptation through decorrelation and variance normalization. *Image and Vis. Comput.*, 2012.

[12] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *TPAMI*, 2012.

[13] D. Green and J. Swets. *Signal Detection Theory and Psychophysics*. John Wiley, 1966.

[14] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006.

[15] X. Hou, J. Harel, and C. Koch. Image signature: Highlighting sparse salient regions. *TPAMI*, 2012.

[16] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vis. res.*, 2000.

[17] L. Itti and C. Koch. Computational modeling of visual attention. *Nature reviews neuroscience*, 2001.

[18] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *TPAMI*, 1998.

[19] T. Jost, N. Ouerhani, R. Von Wartburg, R. Müri, and H. Hügli. Assessing the contribution of color in visual attention. *Computer Vision and Image Understanding*, 2005.

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

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

[22] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*. 1987.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[24] M. Kümmerer, L. Theis, and M. Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv*, 2014.

[25] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014.

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

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

[28] R. J. Peters, A. Iyer, L. Itti, and C. Koch. Components of bottom-up gaze allocation in natural images. *Vis. res.*, 2005.

[29] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T.-S. Chua. An eye fixation database for saliency detection in images. In *ECCV*. 2010.

[30] N. Riche, M. Mancas, M. Duvinage, M. Mibulumukini, B. Gosselin, and T. Dutoit. Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. *Signal Proc.: Image Comm.*, 2013.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1988.

[32] B. Schauerte and R. Stiefelhagen. Quaternion-based spectral saliency detection for eye fixation prediction. In *ECCV*. 2012.

[33] C. Shen and Q. Zhao. Learning to predict eye fixations for semantic contents using multi-layer sparse network. *Neurocomputing*, 2014.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[35] 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.

[36] B. W. Tatler, R. J. Baddeley, and I. D. Gilchrist. Visual correlates of fixation selection: effects of scale and time. *Vis. res.*, 2005.

[37] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive psychology*, 1980.

[38] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *ACM Multimedia*, 2015.

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

[40] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao. Predicting human gaze beyond pixels. *JoV*, 2014.

[41] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*. 2014.

[42] J. Zhang and S. Sclaroff. Saliency detection: A boolean map approach. In *ICCV*, 2013.

[43] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. Sun: A bayesian framework for saliency using natural statistics. *JoV*, 2008.

[44] Q. Zhao and C. Koch. Learning visual saliency by combining feature maps in a nonlinear manner using adaboost. *JoV*, 2012.