Understanding Low- and High-Level Contributions to Fixation Prediction

Matthias Kümmerer, Thomas S.A. Wallis, Leon A. Gatys, Matthias Bethge
University of Tübingen, Centre for Integrative Neuroscience
{matthias.kuemmerer,thomas.wallis,leon.gatys,matthias}@bethgelab.org

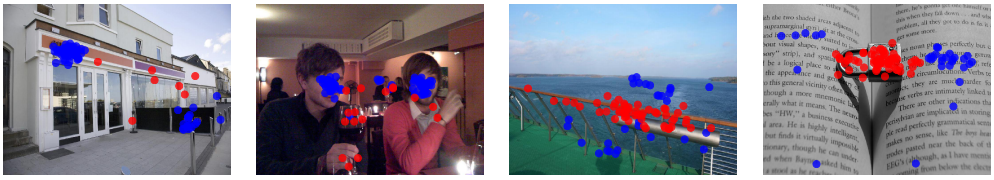


Figure 1: Representative examples for fixation prediction. Fixations are colored depending on whether they are better predicted by the high-level deep object features (DeepGaze II) model (blue) or the low-level intensity contrast features (ICF) model (red). This separates the images into areas where fixations are better predicted by high-level and low-level image features respectively. DeepGaze II is very good at predicting the human tendency to look at text and faces (first and second image), while ICF is better at predicting fixations driven by low-level contrast (third image). In particular, DeepGaze II fails if fixations are primarily driven by low-level features, although high-level features like text are present in the image (fourth image).

Abstract

Understanding where people look in images is an important problem in computer vision. Despite significant research, it remains unclear to what extent human fixations can be predicted by low-level (contrast) compared to high-level (presence of objects) image features. Here we address this problem by introducing two novel models that use different feature spaces but the same readout architecture. The first model predicts human fixations based on deep neural network features trained on object recognition. This model sets a new state-of-the art in fixation prediction by achieving top performance in area under the curve metrics on the MIT300 hold-out benchmark (AUC = 88%, sAUC = 77%, NSS = 2.34). The second model uses purely low-level (isotropic contrast) features. This model achieves better performance than all models not using features pre-trained on object recognition, making it a strong baseline to assess the utility of high-level features. We then evaluate and visualize which fixations are better explained by low-level compared to high-level image features. Surprisingly we find that a substantial proportion of fixations are better explained by the simple low-level model than the state-of-the-art model. Comparing different features within the same powerful readout architecture allows us to better understand the relevance of low- versus high-level features in predicting fixation locations, while simultaneously achieving state-of-the-art saliency prediction.

1. Introduction

Humans make several eye movements per second, *fixating* their high-resolution fovea on things they want to see. Understanding the factors that guide these eye movements is therefore an important component of understanding how humans process visual information and thus has a wide range of applications in image processing. In computer vision this problem is framed as *saliency prediction*[1]: predicting human fixation locations for a given image [21, 26, 25]. Saliency prediction performance has rapidly improved in the last few years, driven by the advent of models based on pre-trained deep neural networks. The models make use of convolutional filters that have been learned on other tasks, most notably object recognition in the ImageNet dataset [10]. The success of these saliency prediction models suggests that the high-level image features encoded by deep networks (e.g. sensitivity to faces, objects and text) are extremely useful to predict human fixation locations.

Despite recent advances, state-of-the-art models remain below the gold standard model of predicting one human's fixations from all others. Given the success of deep learning approaches, it may be tempting to believe that achieving gold standard performance simply requires employing even deeper, more abstracted feature sets. Here, we instead suggest that saliency prediction models may be neglecting low-level image features (local contrast) and overweighting

[1] Note that the term saliency prediction is sometimes also used in different context not related to eye movements.

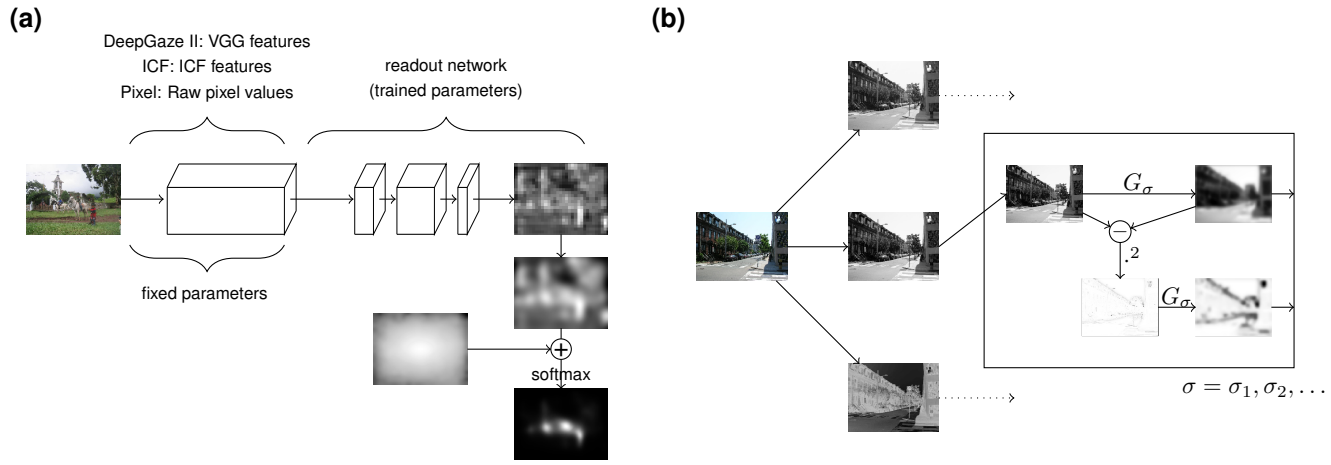


Figure 2: **(a)** The architecture of our models. Each model has a fixed feature space that feeds into the readout network: DeepGaze II uses VGG-19 features, ICF uses simple local intensity and contrast at different scales and the pixel model uses the raw pixel values. These feature activations are passed to a second neural network (the readout network) that is trained for fixation prediction. The readout network consists of four layers of 1×1 convolutions implementing a pixelwise nonlinear function. This results in a saliency map, which is then blurred, combined with a center bias and converted into a probability distribution by means of a softmax. **(b)** The ICF feature space. The network projects an RGB image onto the luminance and two color channels. For each channel we compute local intensities on 5 different scales using Gaussian convolutions. Additionally we square and blur the high-pass residuals from each scale to extract local contrast. The resulting 30 output channels are concatenated and constitute the input to the readout network.

the contribution of high-level features (the presence of objects such as faces or text) in explaining human fixations. We come to this conclusion via three novel contributions:

- A new state-of-the-art model for saliency prediction (the DeepGaze II model) that is based on deep neural network features pre-trained on object recognition [39].The model achieves top performance in area under the curve metrics on the MIT300 hold-out benchmark (AUC = 88%, sAUC = 77%, NSS = 2.34).

- A strong low-level baseline model for saliency prediction (Intensity Contrast Feature or ICF) that is based on local intensity and contrast. The model achieves top performance among all models not using features pre-trained on object recognition.

- Extensive quantitative and qualitative analysis to compare the predictions of these models. While DeepGaze II tends to perform better on images containing faces or text, the ICF model still performs better than DeepGaze II on about 10% of the images in the dataset.

Importantly, because both models use the same well-constrained readout architecture (see below), our comparison only reflects differences in the feature spaces (low- vs high-level).

2. Related Work

Beginning with the seminal image-computable model by Itti and Koch [21], many models have been proposed to predict fixations using local low-level features [52, 27], incorporating global features and statistics [15, 47, 18, 6, 14, 12, 37, 36], using simple heuristics [51] or combinations of low- and high-level features [26] (see [3] for a comprehensive review of saliency models before the advent of pre-trained deep features). In parallel, the effects of biases [42, 43, 45, 7] and tasks [38, 28, 44] on fixation placement have been studied. While these considerations are crucial, in this paper we are concerned not with top-down influences such as task, but rather we seek to understand to what extent fixations in free viewing are driven by low-level features or by high-level features [49, 11, 4, 9, 20, 5].

The state-of-the-art in saliency prediction improved markedly since 2014 with the advent of models using deep neural networks. The first model to use deep features (eDN; [48]) trained them from scratch to predict saliency. Subsequently, the DeepGaze I model showed that DNN features trained on object recognition (AlexNet [29] trained on the ImageNet dataset [10]) could significantly outperform training from scratch [32]. The success of this transfer-learning approach is exciting because it captializes on the presumably tight relationship between high-level tasks such as object recognition and human fixation location selection.

Since the initial success of transfer learning for saliency

prediction, a variety of new models followed this example to further improve saliency prediction performance. The SALICON model [19] fine tunes a mixture of deep features from AlexNet [29], VGG-16 [39] and GoogLeNet [41] for saliency prediction using the SALICON and OSIE [50] datasets. DeepFix [30] and PDP [22] fine-tune features from the VGG-19 network [39] for saliency prediction using the SALICON and the MIT1003 dataset. FUCOS [5] finetunes features trained on PASCAL-Context. SALICON and DeepFix substantially improved performance over DeepGaze I in the MIT benchmark ([8]; see below). The main difference of the new state-of-the art model we introduce here is that rather than fine-tuning the VGG-19 features for saliency prediction, we train a read-out network that uses a point-wise nonlinear combination of deep features. Furthermore we train our model in a probabilistic framework optimising the log-likelihood [31] and model the center bias as an explicit prior (as in Deep Gaze I [32]).

3. Models

We formulate our models as probabilistic models that predict fixation densities. Building on previous work applying probabilistic modelling to fixation prediction [2, 49], Kümmerer *et al*. [31, 33] recently showed that formulating existing models appropriately can remove most of the inconsistencies between existing model evaluation metrics. Furthermore, they argued that using log-likelihood as an evaluation criterion represents a useful and intuitive loss function for model evaluation, with close ties to information theory (though other loss functions may have advantages for some use cases [22]). Therefore we train and evaluate our models using the framework of log-likelihood (specifically reported as information gain explained, see [31]) and additionally report key metrics (AUC, sAUC and NSS) on the MIT1003 dataset and from the MIT Saliency Benchmark.

3.1. Deep Object Features (DeepGaze II) model

Here we describe the architecture of our saliency prediction model that is based on deep features that are trained on object recognition (Fig. 2). A given input image is subsampled by a factor 2 and passed through the normalized VGG-19 network for which all filters have been rescaled to yield feature maps with unit mean over the ImageNet dataset [13]. Next, the feature maps of a selection of high-level convolutional layers (conv5_1, relu5_1, relu5_2, conv5_3, relu5_4; selected via random search, see supplement) are up-sampled by a factor of 8 such that spatial resolution is sufficient for precise prediction. These feature maps are then combined into one 3-dimensional tensor with 2560 (5×512) channels, which is used as input for a second neural network that we term the *readout network*. This readout network consists of four layers of 1×1 convolutions followed by ReLu nonlinearities. The first three layers use 16, 32, and 2 features (see supplement for details). The last layer has only one output channel $O(x, y)$. Crucially, the readout network is only able to represent a *point-wise* nonlinearity in the VGG features. This means that the readout network is only able to learn interactions between existing features across channels but not across pixels—i.e. it cannot learn new spatial features.

The final output from the readout network is convolved with a Gaussian to regularize the predictions:

$$S(x, y) = O(x, y) \star G_\sigma \tag{1}$$

Fixations tend to be near to the center of the image in a way which is strongly task and dataset dependent [42]. Therefore we explicitly model the center bias as a prior distribution that is added to S:

$$S'(x, y) = S(x, y) + \log p_{\text{baseline}}(x, y) \tag{2}$$

We use a Gaussian Kernel density estimate over all fixations from the training dataset for p_{baseline} (for more details see 3.4). Finally, $S'(x, y)$ is converted into a probability distribution over the image by the means of a softmax (as for DeepGaze I and for PDP):

$$p(x, y) = \frac{\exp(S'(x, y))}{\sum_{x,y} \exp(S'(x, y))} \tag{3}$$

3.2. Intensity Contrast Feature (ICF) model

The architecture of our low-level ICF model closely follows that of DeepGaze II (Fig. 2). The main difference is that we replace the VGG features that were trained on object recognition by a feature space that can only extract purely low-level image information (intensity and intensity contrast).

To that end we first subsample the image by a factor of 2 and project the RGB color channels onto their principal components for natural images (computed on the MIT1003 dataset, see supplement), which yields the luminance channel and two color channels. For each of these channels we independently compute local intensity and contrast at different spatial scales. For local intensity we simply compute a Gaussian Pyramid with 5 different scales. The standard deviations the Gaussian kernels are 5,10,20,40,80 px and the window size is 171 px. We use nearest-padding so that the output feature map has the same spatial dimensions as the input feature map. For local contrast we first compute 5 high-pass residuals by subtracting each level of the Gaussian Pyramid from the input channel. Then we square these residuals to compute pixel-wise contrast and finally we blur the squared residuals with the same Gaussian kernel that was used to compute the residual (Fig. 2). This procedure yields 5 intensity and 5 contrast feature maps for each input channel and thus results in 30 feature maps that constitute

the input to the readout network. The readout network and the following stages (blurring and adding of center bias) are the same as for DeepGaze II.

3.3. Pixel model

To compute a baseline that evaluates how powerful the readout network is on its own, we also trained a model that applies the readout network and the following stages directly to the RGB pixel values. This model computes no spatial features and can only learn non-linear combinations of the color channels.

Our models are implemented using Lasagne and Theano [1]. For the computation of the VGG features we used the caffe toolbox [23].

3.4. Model Training

Our models are trained using maximum likelihood learning (see [31] for an extensive discussion of why log-likelihoods are a meaningful metric for saliency modelling). If $p(x, y \mid I)$ denotes the probability distribution over coordinates x and y predicted by our model for an image I, the log-likelihood of a dataset is

$$\frac{1}{N}\sum_{i}^{N}\log p(x_i, y_i|I_i), \tag{4}$$

where i indexes the N fixations in the groundtruth data: The ith fixation occured in the image refered to by I_i, at location $(x_i,\ y_i)$. For both models we minimize this loss function only with respect to the parameters of the readout network and the kernel size of the Gaussian used to regularize the prediction. Since the loss function is differentiable in these parameters, we can use the of-the-shelf *Sum-of-Functions-Optimizer* (SFO, [40]), a mini-batch-based version of L-BFGS.

The feature representations that feed into the readout network (VGG for the high-level and local mean and contrast for the low-level model) are kept fixed during training.

In the pretraining phase, the readout network is initialized with random weights and trained on the SALICON dataset [24]. This dataset consists of 10000 images with pseudofixations from a mouse-contingent task and has proven to be very useful for pretraining saliency models [19, 22, 30]. All images are downsampled by a factor of two. We use 100 images per mini-batch for the SFO. All fixations from the SALICON dataset are used to compute the centerbias.

The MIT1003 dataset is used to determine when to stop the training process. After each iteration over the whole dataset (one epoch) we calculate the performance of the model on the MIT1003 (test) dataset. We wish to stop training when the test performance starts to decrease (due to overfitting). We determine this point by comparing the performance from the last three epochs to the performance five epochs before those. Training runs for at least 20 epochs, and is terminated if all three of the last epochs show decreased performance or if 800 epochs are reached. As it is more expensive to use images of many different sizes, we resized all images from the MIT1003 dataset to either a size of 1024×768 or 768×1024 depending on their aspect ratio, before downsampling by a factor of two. All fixations from the MIT1003 dataset except the ones from the image in question are used to compute the centerbias.

After pre-training, the model is fine-tuned on the MIT1003 dataset and performance is cross-validated over images: the images from the dataset are randomly split into 10 parts of equal size. Then ten models are trained starting from the result of the pre-training, each one using 8 of the 10 parts for training, one part for the stopping criterion (following the stopping criterion as above) and keeping one part for testing. All fixations from the training set are used to compute the centerbias for training, validation and test purposes. We use 10 images per mini-batch in the SFO. For evaluation on the MIT300 benchmark dataset we train on MIT1003 using a ten-fold 9-1 training-validation split and average the predictions from the resulting models, using all fixations from the MIT1003 dataset for the centerbias.

Model	AUC	sAUC	NSS
DeepGaze I [32]	84%	66%	1.22
DSCLRCN [34]	87%	72%	**2.35**
DeepFix [30]	87%	71%	2.26
SALICON [19]	87%	74%	2.12
DeepGaze II	**88%**	**77%**	2.34

Table 1: DeepGaze II performance in the MIT300 Saliency Benchmark. DeepGaze II achieves top performance in both AUC and sAUC, and comes a close second in NSS. Note that we use saliency maps without center bias for the sAUC result (see text for more details).

3.5. Model Evaluation

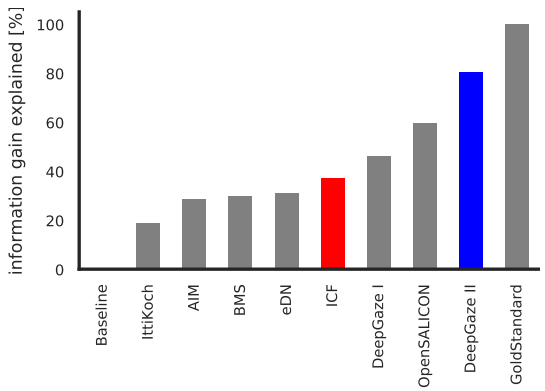
To evaluate model performance we focus on computing *information gain* for its intuitive information-theoretic properties. We additionally report more classic metrics (AUC, sAUC and NSS) to compare to other recent models. Finally, we also report the performance of DeepGaze II on the MIT300 hold-out test set [8].

Information gain tells us what the model knows about the data beyond a given baseline model [31], for which we use the image-independent center bias, expressed in bits / fixation:

$$IG(\hat{p}\|p_{\text{baseline}}) = \frac{1}{N}\sum_{i}\log\hat{p}(x_i, y_i|I_i) - \log p_{\text{baseline}}(x_i, y_i) \tag{5}$$

Here $\hat{p}(x, y|I)$ is the density of the model at location (x, y)

(a)



(b)

Model	IG	IGE	AUC	sAUC	NSS
Centerbias	0.00	0.0	79.6	50.0	1.22
Pixel	0.13	10.7	81.2	60.2	1.38
IttiKoch [16]	0.23	18.6	82.3	64.1	1.41
AIM [6]	0.27	22.6	82.9	65.6	1.50
eDN [48]	0.38	31.1	83.8	68.7	1.61
ICF	0.45	37.2	84.4	70.1	1.74
DeepGaze I [32]	0.56	46.1	85.8	73.0	1.92
OpenSALICON [46]	0.73	59.7	86.4	74.2	2.14
DeepGaze II	**0.98**	**80.3**	**88.3**	**77.7**	**2.48**
Gold Standard	1.22	100.0	89.9	81.2	2.82

Figure 3: Performance on the MIT1003 dataset. **(a)** Ranking of the models according to information gain explained. Our models are marked by the colored bars. All models to the right of ICF use pre-trained deep features. **(b)** Detailed results for a larger set of metrics. IG = information gain (bits / fixation), IGE = information gain explained (%), AUC = area under the ROC curve (%), sAUC = shuffled area under the ROC curve (%), NSS = normalized scanpath saliency.

when viewing image I, and p_{baseline} is the density of the baseline model.

To evaluate the absolute performance of a model we also compute *information gain explained*. This relates the model's performance to the performance of a gold standard model that predicts one subject's fixations for a given image from the fixations of all other subjects using a Gaussian kernel density estimate.

In particular, it is the proportion of the gold standard information gain accounted for by the model:

$$\frac{IG(p\|p_{\text{baseline}})}{IG(p_{\text{gold}}\|p_{\text{baseline}})} \tag{6}$$

where p_{gold} is the density of the gold standard model. Thus it intuitively ranks a model on a scale from 0 to 1, where 0 is a model that does not know the image and 1 is a perfect model that is only limited by inter-subject variablility.

Additionally, we evaluate the traditional area under the ROC curve metrics *AUC* and *sAUC* and the more recent Normalized Scanpath Saliency (NSS, [35]). For AUC and NSS the model's density prediction is the right saliency map to use for evaluation. For sAUC we need to divide the density prediction by the center bias density (which is the non-fixation density in that case) [33].

In all our results we report the test performance of the models. Specifically, for each image in the MIT1003 dataset there is exactly one model from the fine-tuning crossvalidation procedure that did not use that image for training or validation. We use the density prediction from this model to evaluate model performance for that image. For the gold standard model we report leave-one-subject-out crossvalidation performance (which is an image-specific prediction crossvalidated over subjects).

To obtain meaningful results for other models on the information gain metric, we applied the procedure suggested by [31] to convert them to probabilistic models. Specifically, this involves optimizing a pointwise nonlinearity and a center bias (unlike [31], here we do not optimize a blur kernel for the models because all state-of-the-art models produce smooth saliency maps). The conversion usually improves the performance of the models also on the classic metrics. Thus we only report the post-conversion model performances for these models below.

4. Results

4.1. MIT300 Saliency Benchmark

Here we report the performance of our Deep Object Feature model DeepGaze II on the MIT saliency benchmark (the held-out MIT 300 set) (Table 1). DeepGaze II beats the nearest competitors SALICON, DeepFix and DSCLRCN [34] by one percent in AUC. For shuffled AUC, our model beats the nearest competitors by a larger margin. DSCLRCN beats our model by a small margin on NSS (note that this model was optimized for NSS).

Because the MIT Benchmark requires submission of model predictions as JPEG images, one must decide how to store the saliency maps as JPEG images. For AUC, we quantized the density for each image into 256 values such that each value receives the same number of pixels. For sAUC, we divided the density by the density of the MIT1003 center bias and quantized as above. For NSS we quantized the density without histogram normalization. Note that this does not mean we report the results of three different models. The different metrics interpret the saliency maps differently and we translated the predic-

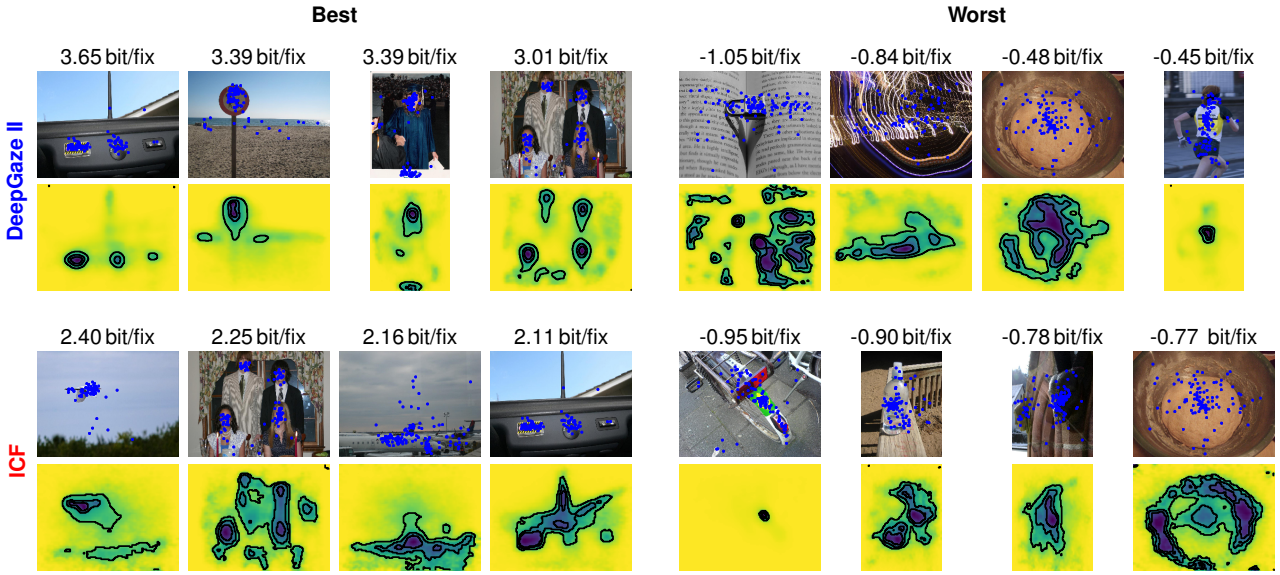


Figure 4: Example images and predictions. For both DeepGaze II and the ICF model we present the best (left) and worst (right) four images with respect to information gain. Ground-truth fixations are plotted in blue over the images. Below each image we show the prediction of the corresponding model. Above each stimulus we report the information gain performance of the model on this image.

tions of our model into the language of the different metrics (without any retraining, see [33] for details). This could partially explain the larger difference between our model and competitor models on sAUC: most state-of-the-art models include a center bias and evaluate sAUC on the saliency maps with a center bias, resulting in a penalty.

4.2. MIT1003 dataset

The MIT300 hold-out set determines the state-of-the-art in saliency prediction. However, precisely because its ground-truth fixations are not publicly available, it is not useful for understanding why models perform the way they do. To develop a deeper understanding of the performance of DeepGaze II and compare it to the ICF model, we therefore evaluate test performance for the MIT1003 dataset. This also allows us to compare models using the intuitive information gain measure described above. Unfortunately we cannot include some recent and competitive models in this analysis (SALICON and DeepFix) because their code is not publicly available. To give at least an approximate result for the previous state-of-the-art, we include results for the OpenSALICON implementation [46].

We evaluate a number of important saliency models using information gain explained (Fig. 3). We display the ranking of the models in Figure 3(a). Our Pixel Model performs the worst, but still remarkably well, accounting for 10% of the information gain of the gold standard over the center bias. Next are models that use hand-crafted low-level features (AIM and BMS) and a convolutional network that is trained from scratch (eDN). Our low-level baseline, the ICF model performs best among all models that do not use pre-trained deep features and accounts for a remarkable 37% of the information gain. Top performance is achieved by models that use pre-trained deep neural network features such as DeepGaze I, OpenSALICON and our new state-of-the-art model DeepGaze II, which can explain 81% of the information gain. Additionally we report the classic measures, AUC, sAUC and NSS to show their consistency with information gain (Fig. 3(b)).

See the supplement for details on how the readout network, the VGG features and pretraining on SALICON contribute to the performance of DeepGaze II.

4.3. What features drive human fixation locations?

Here we compare our low-level ICF and high-level DeepGaze II saliency models to improve our understanding of the features that can explain human fixation locations.

First we look at the images for which each model performs best and worst compared to the center bias and show the respective saliency predictions of the models (Fig. 4). We find that the ICF model performs best on images for which fixations are localized in high contrast regions, for example when there is a single plane in the blue sky (Fig. 4, bottom left panel, first image). At the same time it performs worst when there is a high contrast region that does not attract human fixations or attracts them only in part. For example, it expects people to fixate exclusively on the colored sticker on the bike whereas true fixations are more scattered in the image (Fig. 4, bottom right panel, first image). Note that even though the model only extracts low-level fea-

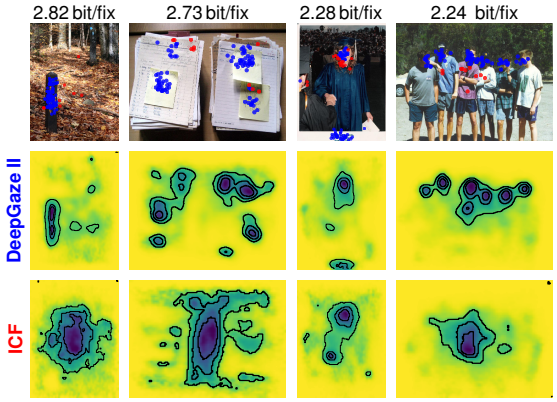


Figure 5: Images for which DeepGaze II has the largest improvement over ICF. Fixations that are better explained by DeepGaze II are colored in blue. Fixations that are better explained by ICF are colored in red. Fixations best explained by the center bias are omitted. Predicted fixation densities for both models are plotted below the images. Above each stimulus we report the difference in information gain between DeepGaze II and ICF for this image.

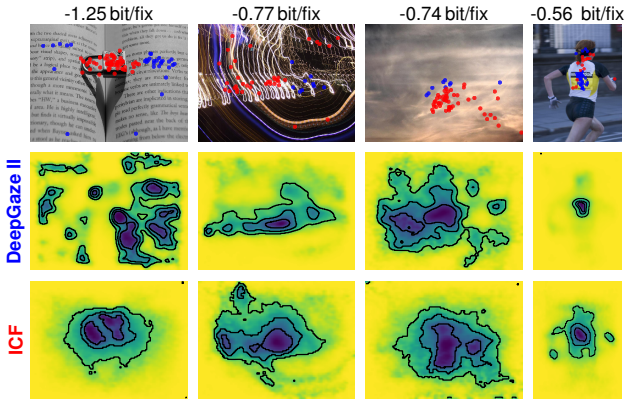


Figure 6: Images for which ICF has the largest improvement over DeepGaze II. Other elements as in Figure 5.

tures, it can still perform well on images where fixations are driven by high-level features such as human faces—if the presence of a face is correlated with the local intensity or contrast of the image (Fig. 4, bottom left panel, second image).

We find that DeepGaze II excels at predicting fixations that are driven by the presence of objects, such as controls in a car, a roadsign or human faces (Fig. 4, top left panel). It fails for images where high-level content is not associated with fixations (e.g. the text in Fig. 4, top right panel, first and last image) or images that are texture-like without any particular objects (Fig. 4, top right panel, second image).

Even though the best images for ICF and DeepGaze II are partly the same, the predicted saliency maps clearly separate the models. While DeepGaze II is extremely accurate

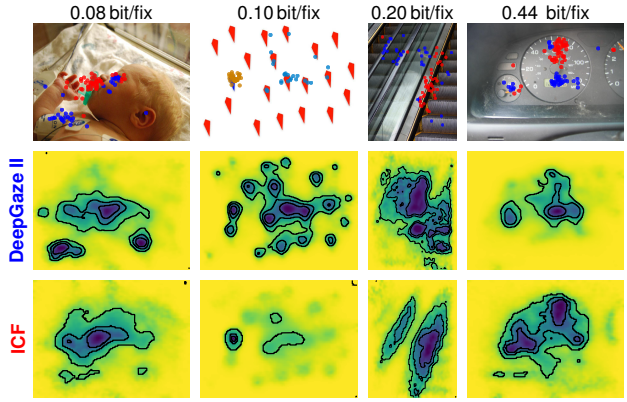


Figure 7: Images for which DeepGaze II and ICF show similar performances but predict the fixations in different locations, separating the image into areas of low-level and high-level fixations. Other elements as in Figure 5, except that in the second image ICF fixations are colored orange and DeepGaze II light blue to better separate them from the blue and red elements in the image.

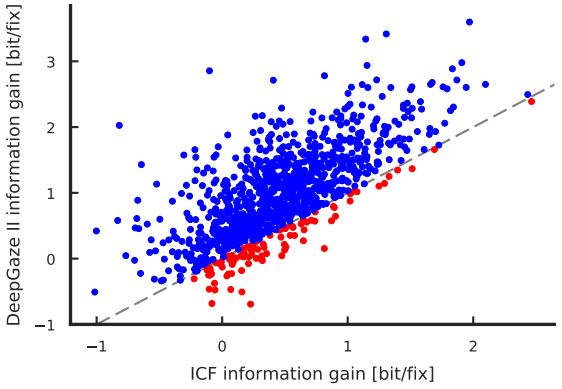


Figure 8: Performances of DeepGaze II and ICF on MIT1003. Each point corresponds to one image, with the performance of DeepGaze II (y-axis) and the ICF model (x-axis) on that image expressed as information gain relative to the center bias. For images above the diagonal (blue dots) DeepGaze II is better than the ICF model, while for images below the diagonal (red dots) the ICF model is better.

at predicting fixations at the location of the important high-level objects (faces, text), the ICF model also predicts fixations at other high-contrast locations in the images.

The difference between the models is made more explicit by looking at the images for which DeepGaze II is maximally better than ICF (Fig. 5). One advantage of training two separate models is that we can easily assess which individual fixations within an image are better explained by each of the models. This allows us to better understand which features drive fixations. In Figure 5, DeepGaze II correctly predicts a concentration of fixations over text (two leftmost images) and faces (two rightmost images) whereas

the ICF model is 'distracted' by high-contrast regions of the image that do not correspond to the presence of objects [49]. For example, ICF strongly predicts fixations in the high-contrast gap between the stacks of papers in the second image. It also predicts high fixation probability for skin on a dark background no matter whether it is the face or the hand of a person (third image). In contrast, DeepGaze II only predicts high fixation probability for the face, in agreement with the ground-truth data.

On the other hand, in Figure 6, we show images where the ICF model performs better than DeepGaze II. In two examples (first and fourth images), DeepGaze II seems to be distracted by high-level features that humans tend not to fixate. For example, in the first image, DeepGaze II predicts that humans will look at the text printed in the book whereas ICF correctly predicts that humans will fixate the padlock lying over the page (forming a high-contrast region). Similarly, in the fourth image, DeepGaze II predicts fixations on the text on the runner's shirt whereas the runner's head and shoulders happen to correspond to higher-local-contrast regions (which are picked up by the ICF model). The middle two images show abstract patterns (motion blur and clouds) for which human fixations appear to be better explained by local contrast in the absence of high-level features.

Finally, we show a sample of images in which DeepGaze II and ICF show similar performance at the image level but predict fixations in different locations (Figure 7). In the first image, DeepGaze II correctly predicts fixations to the baby's eyes and to the text on the arm, but ICF correctly predicts fixations to the pacifier. In the second image, ICF correctly predicts fixations to the color-singleton search item (blue element amongst red) but fails to predict fixations elsewhere. DeepGaze II predicts fixations to the glass window whereas ICF predicts fixations to the high-contrast border of the escalator in the third image, and DeepGaze II predicts fixations to text but not the needle of the speedometer in the fourth image.

The comparison images we have highlighted above show that DeepGaze II can correctly predict fixations to high-level features such as text and faces (see also examples in Figure 1), in accordance with its status as a far more powerful model than ICF (more parameters with pre-trained features). However, there are striking failure cases when comparing against the ICF model, in particular when high-level features are present in the image but are not fixated (e.g. the text and padlock image in Figure 6). On the MIT1003 dataset as a whole, we find that there is a substantial subset of images (94 of 1003) for which the ICF model produces better predictions than DeepGaze II (Figure 8). In terms of individual fixations this proportion is even higher, with around 25% of the fixations in the dataset being better explained by ICF than either DeepGaze II or the center bias. Given the simplicity of the ICF model relative to DeepGaze II, this is remarkable. Because in principle DeepGaze II should also have access to low-level features [17], this result suggests that DeepGaze II may be underweighting the importance of low-level features in guiding fixations.

5. Discussion

In this paper we compare the predictive performance of low- and high-level features for saliency prediction by introducing two new saliency models that use the same readout architecture on top of different feature spaces. DeepGaze II uses transfer learning from the VGG-19 deep neural network to achieve state-of-the-art performance on the MIT300 benchmark. The ICF model uses simple intensity contrast features to achieve better performance than all models that do not use pre-trained deep features.

While the high-level DeepGaze II model significantly outperforms low-level ICF for the dataset as a whole, we find a surprisingly large set of images for which the ICF model is better than DeepGaze II. Thus, while high-level features (the presence of objects, faces and text) are very important for explaining free viewing behaviour in natural scenes [11, 44], our results show that low-level local contrast features do make a small but dissociable contribution over a representative scene database (see also [7, 5]).

The fact that the simple ICF model outperforms all models before transfer learning of deep features shows that the predictive value of low-level features has been historically underestimated. One possible reason for this is that many historical models were not trained on data but rather hand-tuned. On the other hand, the ICF model is isotropic—it does not even have access to orientation filters—which makes its performance improvement relative to earlier models even more remarkable.

Our results suggest that explicitly modelling low-level contributions to saliency could be used to improve the robustness of saliency models. In future work it may prove fruitful to train the DeepGaze II and ICF models jointly, reducing DeepGaze II's tendency to over-emphasize the importance of high-level image structure. Ultimately however, we believe that improvements will come from a better understanding of what features causally drive fixation behaviour, including different task constraints [44, 28].

We provide a webservice to test our models on arbitrary stimuli at `deepgaze.bethgelab.org`.

6. Acknowledgements

Funded by the the German Excellency Initiative (EXC307), the German Science Foundation (DFG; priority program 1527, BE 3848/2-1 and Collaborative Research Centre 1233), the German Academic Foundation and the BCCN Tübingen (BMBF; FKZ: 01GQ1002).

References

[1] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, Y. Bengio, A. Bergeron, J. Bergstra, V. Bisson, J. Bleecher Snyder, N. Bouchard, N. Boulanger-Lewandowski, X. Bouthillier, A. de Brébisson, O. Breuleux, P.-L. Carrier, K. Cho, J. Chorowski, P. Christiano, T. Cooijmans, M.-A. Côté, M. Côté, A. Courville, Y. N. Dauphin, O. Delalleau, J. Demouth, G. Desjardins, S. Dieleman, L. Dinh, M. Ducoffe, V. Dumoulin, S. Ebrahimi Kahou, D. Erhan, Z. Fan, O. Firat, M. Germain, X. Glorot, I. Goodfellow, M. Graham, C. Gulcehre, P. Hamel, I. Harlouchet, J.-P. Heng, B. Hidasi, S. Honari, A. Jain, S. Jean, K. Jia, M. Korobov, V. Kulkarni, A. Lamb, P. Lamblin, E. Larsen, C. Laurent, S. Lee, S. Lefrancois, S. Lemieux, N. Léonard, Z. Lin, J. A. Livezey, C. Lorenz, J. Lowin, Q. Ma, P.-A. Manzagol, O. Mastropietro, R. T. McGibbon, R. Memisevic, B. van Merriënboer, V. Michalski, M. Mirza, A. Orlandi, C. Pal, R. Pascanu, M. Pezeshki, C. Raffel, D. Renshaw, M. Rocklin, A. Romero, M. Roth, P. Sadowski, J. Salvatier, F. Savard, J. Schlüter, J. Schulman, G. Schwartz, I. V. Serban, D. Serdyuk, S. Shabanian, E. Simon, S. Spieckermann, S. R. Subramanyam, J. Sygnowski, J. Tanguay, G. van Tulder, J. Turian, S. Urban, P. Vincent, F. Visin, H. de Vries, D. Warde-Farley, D. J. Webb, M. Willson, K. Xu, L. Xue, L. Yao, S. Zhang, and Y. Zhang. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[2] S. Barthelmé, H. Trukenbrod, R. Engbert, and F. Wichmann. Modelling fixation locations using spatial point processes. *Journal of Vision*, 13(12), 2013.

[3] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, Jan. 2013.

[4] A. Borji, D. N. Sihite, and L. Itti. Objects do not predict fixations better than early saliency: A re-analysis of Einhauser et al.'s data. *Journal of vision*, 13(10):18, Jan. 2013.

[5] N. D. Bruce, C. Catton, and S. Janjic. A deeper look at saliency: feature contrast, semantics, and beyond. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 516–524, 2016.

[6] N. D. Bruce and J. K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of vision*, 9(3), 2009.

[7] N. D. Bruce, C. Wloka, N. Frosst, S. Rahman, and J. K. Tsotsos. On computational modeling of visual saliency: Examining what's right, and what's left. *Vision Research*, 116:95–112, nov 2015.

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

[9] M. Cerf, J. Harel, W. Einhaeuser, and C. Koch. Predicting human gaze using low-level saliency combined with face detection. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 241–248. Curran Associates, Inc., 2008.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[11] W. Einhauser, M. Spain, and P. Perona. Objects predict fixations better than early saliency. *Journal of Vision*, 8(14):18–18, Nov. 2008.

[12] E. Erdem and A. Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of vision*, 13(4), 2013.

[13] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[14] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1915–1926, 2012.

[15] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2006.

[16] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2006.

[17] H. Hong, D. L. K. Yamins, N. J. Majaj, and J. J. DiCarlo. Explicit information for category-orthogonal object properties increases along the ventral stream. *Nature Neuroscience*, 19(4):613–622, Feb. 2016.

[18] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[19] X. Huang, C. Shen, X. Boix, and Q. Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[20] L. Itti. Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes. *Visual Cognition*, 12(6):1093–1123, 2005.

[21] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254–1259, 1998.

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

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[24] M. Jiang, S. Huang, J. Duan, and Q. Zhao. SALICON: Saliency in context. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers (IEEE), jun 2015.

[25] T. Judd, F. d. Durand, and A. Torralba. A Benchmark of Computational Models of Saliency to Predict Human Fixations A Benchmark of Computational Models of Saliency to Predict Human Fixations. *CSAIL Technical Reports*, 2012.

[26] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th international conference on*, pages 2106–2113. IEEE, 2009.

[27] W. Kienzle, M. O. Franz, B. Schölkopf, and F. A. Wichmann. Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of Vision*, 9(5), 2009.

[28] K. Koehler, F. Guo, S. Zhang, and M. P. Eckstein. What do saliency models predict? *Journal of Vision*, 14(3):14–14, mar 2014.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[30] S. S. Kruthiventi, K. Ayush, and R. V. Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *CoRR*, abs/1510.02927, 2015.

[31] M. Kümmerer, T. S. A. Wallis, and M. Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112(52):16054–16059, dec 2015.

[32] M. Kümmerer, L. Theis, and M. Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. In *2015 International Conference on Learning Representations - Workshop Track (ICLR)*, 2015.

[33] M. Kümmerer, T. S. A. Wallis, and M. Bethge. Saliency benchmarking: Separating models, maps and metrics. *arXiv e-prints*, abs/1704.08615, 2017.

[34] N. Liu and J. Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *arXiv preprint arXiv:1610.01708*, 2016.

[35] R. J. Peters, A. Iyer, L. Itti, and C. Koch. Components of bottom-up gaze allocation in natural images. *Vision Research*, 45(18):2397–2416, aug 2005.

[36] S. Rahman and N. Bruce. Saliency, scale and information: Towards a unifying theory. In *Advances in Neural Information Processing Systems 28*, pages 2188–2196, 2015.

[37] N. Riche, M. Mancas, M. Duvinage, M. Mibulumukini, B. Gosselin, and T. Dutoit. RARE2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. *Signal Processing: Image Communication*, 28(6):642–658, July 2013.

[38] C. A. Rothkopf, D. H. Ballard, and M. M. Hayhoe. Task and context determine where you look. 7(14):16–16.

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

[40] J. Sohl-Dickstein, B. Poole, and S. Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-newton methods. *CoRR*, abs/1311.2115, 2013.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[42] B. W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):4, 2007.

[43] B. W. Tatler, R. J. Baddeley, and I. D. Gilchrist. Visual correlates of fixation selection: effects of scale and time. *Vision Research*, 45(5):643 – 659, 2005.

[44] B. W. Tatler, M. M. Hayhoe, M. F. Land, and D. H. Ballard. Eye guidance in natural vision: Reinterpreting salience. *Journal of vision*, 11(5), 2011.

[45] B. W. Tatler and B. T. Vincent. Systematic tendencies in scene viewing. *Journal of Eye Movement Research*, 2(2):1–18, 2008.

[46] C. Thomas. Opensalicon: An open source implementation of the salicon saliency model. *CoRR*, abs/1606.00110, 2016.

[47] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006.

[48] E. Vig, M. Dorr, and D. Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Computer Vision and Pattern Recognition, 2014. CVPR'14. IEEE Conference on*. IEEE, 2014.

[49] B. T. Vincent, R. J. Baddeley, A. Correani, T. Troscianko, and U. Leonards. Do we look at lights? Using mixture modelling to distinguish between low- and high-level factors in natural image viewing. *Visual Cognition*, 17(6-7):856–879, 2009.

[50] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao. Predicting human gaze beyond pixels. *Journal of vision*, 14(1):28–28, 2014.

[51] J. Zhang and S. Sclaroff. Saliency detection: a boolean map approach. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 153–160. IEEE, 2013.

[52] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7), 2008.

Understanding Low- and High-Level Contributions to Fixation Prediction: Supplementary Material

1 Contributions of architectural components to performance

Our DeepGaze II model uses a similar architecture to DeepGaze I [1], with four primary changes: replacing AlexNet features by VGG features, using a readout network instead of a linear readout, pre-training on the SALICON dataset, and using image-wise crossvalidation over the full MIT1003 dataset rather than subject-wise crossvalidation over only a subset. We quantified the contributions of these changes to achieving our model performance using the full MIT1003 dataset. As seen in Table 1, switching to image-wise crossvalidation on the full dataset (DeepGaze I′) provides a substantial performance boost over the original DeepGaze I model. After considering this change, the largest single improvement over DeepGaze I′ comes from using the pre-trained VGG features in place of AlexNet (though note that we also include more channels from VGG than from AlexNet). Training DeepGaze I′ on the SALICON dataset does not change performance, suggesting that the 258 parameters of this model are already sufficiently constrained by the MIT1003 dataset. Combining SALICON pre-training with the VGG features yields the largest intermediate model performance improvement. Using the readout network without additional pre-training on the SALICON dataset never gives substantially better performance (compare DeepGaze I′ to "readout network", or "VGG" to "readout net + VGG"), suggesting that SALICON pre-training is required for the readout network to avoid overfitting.

Model	IG	IGE	AUC	sAUC	NSS
Centerbias	0.00	0.0	79.6	50.0	1.22
DeepGaze I	0.56	46.1	85.8	73.0	1.92
DeepGaze I′	0.76	62.3	86.9	75.0	2.16
readout network	0.75	62.0	87.0	75.0	2.16
SALICON	0.76	62.6	86.9	75.0	2.16
VGG	0.84	69.3	87.7	76.4	2.32
Readout net+SALICON	0.82	67.5	87.3	75.6	2.25
Readout net+VGG	0.85	70.0	87.3	76.2	2.34
SALICON+VGG	0.90	74.3	88.0	76.9	2.42
DeepGaze II	**0.98**	**80.3**	**88.3**	**77.7**	**2.48**
Gold Standard	1.22	100.0	89.9	81.2	2.82

Table 1: Contributions of changes between DeepGaze I and DeepGaze II to performance. DeepGaze I′ is the DeepGaze I model trained with image-wise crossvalidation over the full MIT1003 dataset just like our models. "Readout network" = replacing a linear readout with a nonlinear readout network, "VGG" = replacing AlexNet with VGG features, "SALICON" = pre-training on the SALICON dataset. Metrics as in main paper. The primary improvement in our model compared to DeepGaze I′ comes from using VGG features.

2 Readout network

Our model architecture uses a readout network consisting of multiple layers of 1×1 convolutions on top of a fixed set of features. This allows the models to learn nonlinear combinations of the features and fit the scale of the final log density better while still being comparatively constrained. We estimate how much these two features contribute to the performance when compared to a simple linear readout for ICF and DeepGaze II. In Figure 1, we show models with different readout networks: first, we just use a linear readout as baseline to compare to. Then we use a readout network with layers of 1, 128 and 1 channels ("LN"). Since the first layer has only one feature, this allows the readout network only to learn a nonlinear transformation of a saliency map but keeps it from exploiting interactions between features. Finally we show the performance of the model with the full readout network, which therefore is able to fit the log density scale as well as make use of interactions between features.

We find that the linear DeepGaze II model already accounts for roughly 74% of the explainable information gain. The LN readout network manages to close around two thirds of the performance gap to the full readout network, indicating that DeepGaze II mainly uses the readout network to transform the scale of the saliency prediction and not so much to exploit interactions between features.

For the ICF model on the other hand, the LN readout network increaes the performance only by one third of the difference between the linear readout and the full readout. This shows that the ICF model makes much more use of interactions between features and DeepGaze II.

In Figure 2 we compare the performance of DeepGaze II when using different depths for the readout network. Going from a purely linear readout to one hidden layer gives more than half of the performance gain to the final model with three hidden layers. Two hidden layers yields a performance which is only slightly worse than three hidden layers.

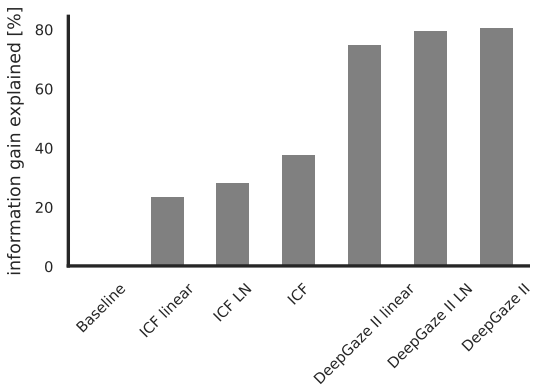


Figure 1: Performances of ICF and DeepGaze II when using either a linear readout, a linear-nonlinear readout network with layers of 1, 128 and 1 channels which cannot exploit feature interactions and the full readout network as described in the main paper.

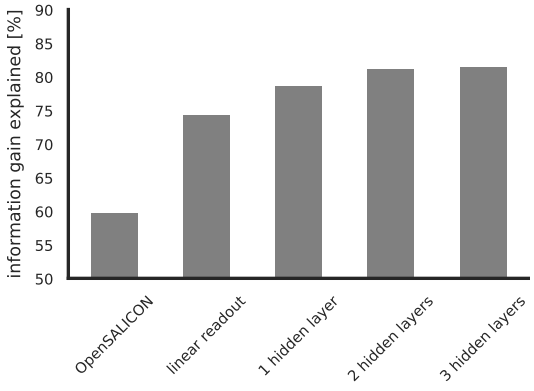


Figure 2: Influence of the depth of the readout network. We show the performance of DeepGaze II when using a linear readout, one hidden layer (16 units), two hidden layers (16 and 32 units) and the final readout network with three hidden layers of 16, 32 and 2 units.

3 VGG features

In DeepGaze II presented in the main paper, we use the conv5_1, relu5_1, relu5_2 conv5_3 and relu5_4 layers from VGG-19 as feature space. These layers have been

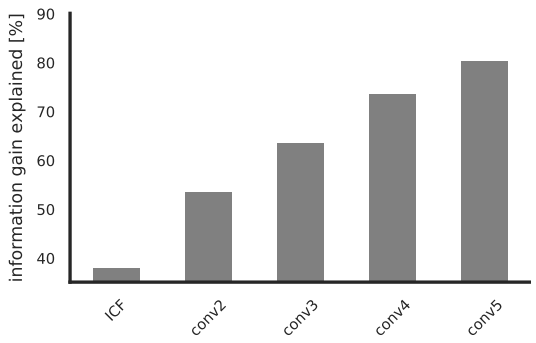


Figure 3: Performance of DeepGaze II when using features from different levels in VGG.

choosen with a random search which trained models using a random selection of layers from the conv4 and conv5 blocks. To compare the predictive power of the different layer blocks in VGG-19, in Figure 3 we show the performance of DeepGaze II when using features from conv2, conv3, conv4 or conv5. For conv3 and conv4 we used conv$n$_1, relu$n$_1, relu$n$_2 conv$n$_3 and relu$n$_4, corresponding to the layers from conv5 used in the final model. For conv2 we used conv2_1, relu2_1, conv2_2, relu2_2. The performances increase steadily from the conv2 model to the conv5 model, but already the conv2 model is significantly better than the ICF model.

4 Principal component analysis for ICF features

The ICF model projects the RGB color channels onto their principal components for natural images. We computed the principal components using all pixels in the MIT1003 dataset. The resulting compontents are up to small deviations: 1) grayscale intensity 2) 50/50 Red/Green 3) 25/50/25 Red/Blue/Green. This color space is not likely to be overfit to the MIT1003 datset, because the SALICON dataset gave almost identical numbers.

References

[1] M. Kümmerer, L. Theis, and M. Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. In *2015 International Conference on Learning Representations - Workshop Track (ICLR)*, 2015. 1