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.

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

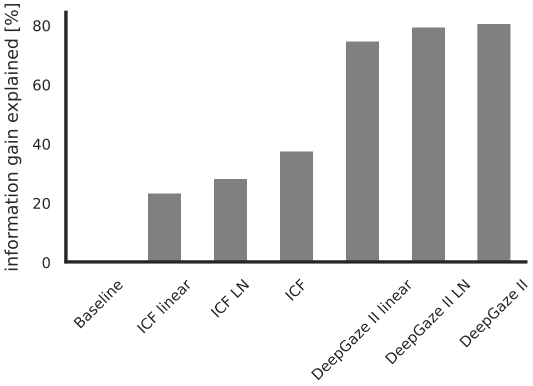

Figure 1: Performances of ICF and DeepGaze II when using either a linear readout, a linear-nonlinear readout network with layers of 1, 128 and 1 channels which cannot exploit feature interactions and the full readout network as described in the main paper.

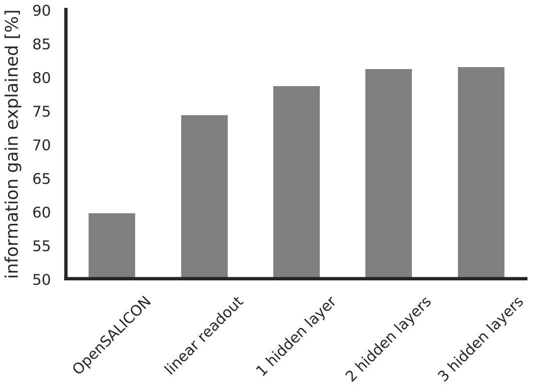

Figure 2: Influence of the depth of the readout network. We show the performance of DeepGaze II when using a linear readout, one hidden layer (16 units), two hidden layers (16 and 32 units) and the final readout network with three hidden layers of 16, 32 and 2 units.

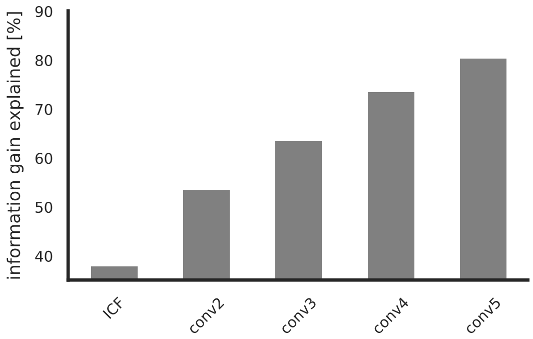

Figure 3: Performance of DeepGaze II when using features from different levels in VGG.

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