Learning Visual Attention for Robot Vision

Denis Musinguzi, Kevin Sebineza, Jean de Dieu Nyandwi, Muhammed Danso

Carnegie Mellon University Africa
Kigali, Rwanda
{dmusingu, ksebinez, jeandedi, mdanso}@andrew.cmu.edu

Abstract

Visual saliency maps represent the attention-grabbing area of an image based on the human vision nerve system. The concept plays a crucial role in robot vision but faces the underlying challenges of heavy baseline architectures which are difficult to run on robots with limited computation power. In this study, we re-implement the TransalNet and DeepGazeII visual saliency models, evaluate them on the SALICON and MIT1003 benchmark datasets, and experiment with their performances on different backbone architecture. We find a linear relationship between model performance on ImageNet benchmark and saliency map predictions. We also observed that deeper architectures performed better. The best performing model is TransalNet with ConvNeXT(large) backbone which achieved a saliency correlation coefficient of 0.8117 and SIM of 0.6663. The code accompanying this report can be found here: `https://github.com/Nyandwi/learning-visual-attention-for-robotic-vision`.

1 Introduction

Human visual salience represents the human brain's process of drawing the eyes to a certain point when the subject looks at an image. We apply this concept in our everyday life, where we select relevant information from vast amounts of visual sensory information our brain receives [1]. According to studies in cognitive science, vision information accounts for 80% of all information humans receive daily [2]. However, since our vision nerve system is limited, the visual attention mechanism plays an important role in processing important regions.

Predicting visual saliency maps plays a crucial role in perceiving regions where people are likely to focus. In cognitive robotics, visual saliency prediction enables the robot to accomplish tasks by focusing on regions aligned with the task [4]. Additionally, for a robot to collaborate effectively with a human, it requires abilities like anticipation and prospection, which heavily rely on visual attention. Visual saliency prediction is also applied in human-computer interaction to improve user interfaces and system usability, among other functionalities [5].

Modeling human visual attention in artificial vision systems such as robotics remains challenging, given limited computational power as compared to the natural visual attention mechanism [3]. Recent advances in deep learning models for computer vision have proposed various networks to predict saliency maps from an image, ranging from traditional classic models that employ low-level cues to deep learning networks that extract high-level features. However, existing deep learning models are heavy for robot applications, and lack generalization because they are usually trained on constrained datasets with upright images only [2].

The aim of this research is to experiment with the performance of the existing baseline models with different depths and different backbone models. We re-implement TransalNet[24] and DeepGazeII[14] models, and test the effect of different backbone architectures, such as ConvNext and Resnet-150, on the baseline model and their potential to improve its generalization performance.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Literature Review

Research on saliency detection mainly involves two types of tasks: saliency prediction or eye fixation prediction and salient Object detection [6]. Both tasks involve the detection of the most significant area of the picture or video. Saliency prediction focuses on predicting the possibility of the human eye staying in a certain position in the scene, while salient object detection focuses on the perception and description of the object level, which is a pure computer vision task.

2.1 Division by type of approach to saliency prediction

Work in the visual saliency prediction space can be categorized into the bottom-up approach, also known as the data-driven or task agonistic approach, and the top-down approach, also known as the task-specific methods [7]. Bottom-up visual saliency models extract low-level features like contrast, color, and texture. This is because the difference between low-level features and background features attracts attention. The model proposed by Itti [8] was the first of such models. It could simulate the process of shifting human visual attention without any prior information. The model selected salient areas in the image depending on the saliency intensity of different positions.

Top-down visual saliency models were based on specific tasks. Modeling was difficult due to the diversity and complexity of tasks. The probabilistic combination model used combines scene and prior information according to Bayes rules[1]. Torrallba et al. [8] proposed a model that multiplied the bottom-up and top-down saliency maps to obtain the final ones. Following this work, Ehinger et al. [9] incorporated feature prior information of the target into the above framework. The SUN model proposed by Zhang et al. [10] used visual features and spatial location as the prior knowledge. The performance of classical models gradually reached a bottleneck due to handcrafted features.

2.2 Models used for saliency prediction

The application of deep networks in visual saliency prediction was due to their successful application in other computer vision tasks such as object recognition. The performance of deep networks in visual saliency prediction was much better than that of classical models. Since there were limited visual saliency datasets, researchers using deep models applied transfer learning from models trained on general image recognition and retrained the models for saliency prediction. This kind of procedure allowed deep models to use the semantic visual knowledge already learned by the CNN and transfer them to visual saliency.

Vig et al. [11] proposed the ensemble of Deep Networks (eDN), the first attempt to use CNNs to predict saliency. The model could automatically learn the image representation and generate a final saliency map by fusing the feature maps from different layers. They used hyperparameter optimization to search for independent models that were predictive of saliency and combined them into one model by training a linear SVM. They were, however, limited by the number of datasets and inadequate depth of their model.

Kummerer et al. [12] proposed DeepGazeI based on the pre-trained AlexNet [8] network. They introduced the idea of using pre-trained model weights from image classification tasks for saliency prediction. Their network also uses center deviation, which they then convert to probability distribution by applying a SoftMax function. They later proposed DeepGazeII [14], which switched to using the VGG-16 network [15].They trained it on the SALICON [16] dataset and then fine-tuned it on the MIT1003 dataset [17]. This significantly improved its performance over the DeepGazeI network. This showed that the backbone model used influenced the performance of the model on saliency prediction. They later introduced DeepGazeIIE [18], which uses different backbones to extract features for saliency prediction. They discovered that concatenating multiple backbones pre-trained on ImageNet was effective in predicting visual saliency. Their model achieves state-of-the-art performance on the MIT300 dataset in the sAUC, AUC, and KLD metrics. This model is chosen as the baseline model, given its performance.

Kruthiventi et al. [19] proposed the DeepFix model, which uses a pre-trained VGG-16 network for feature extraction. It was the first model to apply a fully convolutional neural network for saliency prediction. The model was trained end-to-end and captured semantics at multiple scales while accounting for global context using Location Based Convolution Layers. This overcame the problematic spatial invariance in classic fully convolutional neural networks [14]. They used five convolution blocks whose weights were initialized from the VGG16 network. In addition, they incorporated Gaussian priors to improve the learned weights further.

ML-Net, proposed by Cornia et al. [20], combined features from multiple layers of the VGG16 network to compute saliency instead of using the final CNN layer. Their model consisted of a feature extraction DCNN, a feature coding network, and an apriori learning network. To incorporate center bias, they learn a set of Gaussian parameters end to end as opposed to using a fixed Gaussian. They also introduced a new loss function that was weighted on NSS, CC, and SIM metrics.

Liu and Han [21] proposed the Deep Spatial Contextual Long-term Recurrent Convolution Neural Network (DSCLRCN), which first learns the local saliency of small image regions using a CNN. The image is then scanned both horizontally and vertically using a deep spatial LSTM to capture the global context. This allowed their model to incorporate local and global contexts to infer image saliency simultaneously.

The EML-Net [22] model introduced the idea of training the encoder and decoder parts of the FCNN separately to achieve scalability. They also suggested the use of multiple CNNs with different architectures in the encoder. Their parallel design reduced the computational cost of model training and overcame limits to the variety of information that can be combined at the encoder towards deeper networks. In the decoder, they combined features from the two CNNs trained in the encoding stage. They applied 1x1 convolutions and ReLU to each feature map to reduce the amount of space required to store intermediate representations. Their model can also be expanded to include more pre-trained CNNs with no additional cost.

Wang et al. [23] proposed the Deep Visual Attention model in which an encoder-decoder architecture is trained on multiple scales to predict pixel-wise saliency. The encoder was built by stacking convolution layers. Three decoders were formed by taking inputs from different stages of the encoder network, which they then fused to generate a saliency map.

Due to the successful application of transformers in the field of Natural Language Processing, there have been attempts to apply them in computer vision tasks. Lou et al. [24] proposed TransalNet, a model that uses transformer encodes in CNNs to capture long-range contextual visual information. Their experiments showed that transformers had the ability to overcome CNNs' lack of sufficient long-range contextual encoding capacity.

2.3 Shortcomings of classic and deep learning models

Classic models fell short when extracting high-level features like faces which are important for high visual saliency prediction [7]. Some classical models tried to solve this problem by integrating object detectors such as face and text detectors. The hierarchical nature of deep networks allows them to capture complex cues that attract the gaze [7], opening the gap in performance with classical models. However, deep models also fail to capture some high-level attention cues, such as body posture and gaze direction.

2.4 Saliency metrics

Visual saliency prediction metrics use similarity and differences between estimated predicted values and the Ground Truth to give an evaluation score. The metrics can be categorized into location based and distribution-based metrics depending on the representation of the ground truth. Location based metrics adopt fixation maps whereas distribution-based metrics use saliency map in form of a grayscale image as the ground truth for visual saliency evaluation. Six popular metrics i.e., CC, SIM, KLD, NSS, AUC, sAUC are widely used to quantify the general performance of saliency models.

Normalized Scanpath Saliency (NSS)[27] is used to calculate the average normalized significance value at the point of interest. The mathematical form of NSS is given by

$$NSS = \frac{1}{N} \sum P(i) \times Q(i)$$

where P is the average value at the gaze point Q of the human eye, N is the total number of human eye gazes, I represents the i-th pixel, and N is the total number of pixels at the gaze point. A positive NSS indicates consistency between mappings whereas a negative NSS is the opposite.

Linear Correlation Coefficient (CC): The CC is the statistic used to measure the linear correlation between two random variables. The predicted significance map and the true value view are regarded as the two random variables in the significance prediction evaluation. The mathematical form of CC is given by

$$CC = \frac{cov(P, G)}{\sigma(P) + \sigma(G)}$$

where cov is the covariance, σ is the standard deviation. The CC can equally distinguish false positive and false negative at the value range of -1 and 1. A value close to the two ends indicates a better performance.

Kullback-Leibler (KL) Divergence is an information theory metric corresponding to the difference between two probability distributions. The mathematical form of the KL divergence is

$$KL(P, G) = \sum G_i log\left(e + \frac{G}{\epsilon + P_i}\right)$$

The KL divergence uses the predicted saliency map as an approximation of the Ground truth. ϵ is the regularization term. A low score indicates that the saliency map is close to the true value.

Similarity Metric (SIM) measures the similarity between two distributions. After normalizing the input map, SIM is calculated as the sum of the minimum values at each pixel. The mathematical form of SIM is given by

$$SIM(P, G) = \sum min(P_i, G_i)$$

Given the predicted significance map P and the true value view G, a SIM of 1 means that the distribution is the same, whereas a SIM of 0 means no overlap. SIM penalizes predictions that fail to consider all true densities.

3 Model Description

3.1 TransalNet Model

The base model is composed of a CNN model encoder, transformer layers and a CNN decoder. The input image is processed by the encoder and a set of three feature maps are extracted from the encoder and passed to transformer encoders. The layers extracted from the encoder have different spatial sizes and depths. The output of the transformer encoders is passed to another CNN decoder to obtain the saliency map.

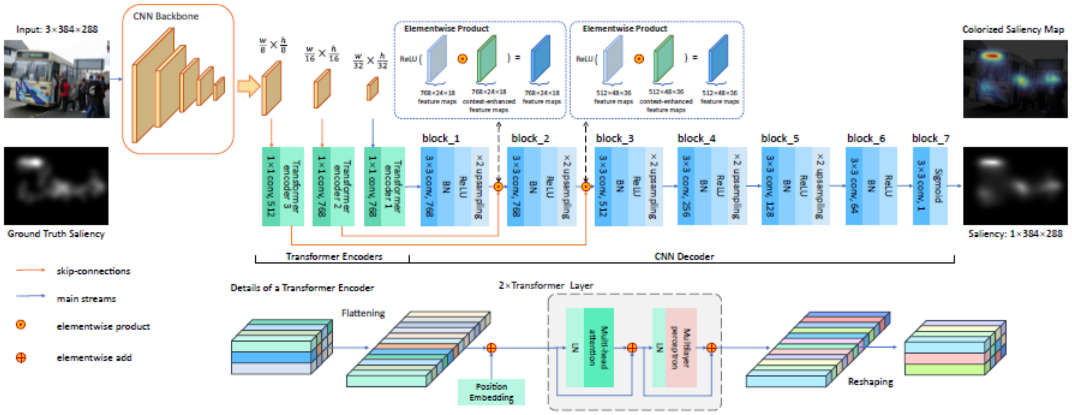

Figure 1: TransalNet Model Architecture

3.1.1 CNN Encoder

CNN-based models have been proven to be effective at extracting features for saliency prediction especially those trained on large image datasets. The fully connected layers of the CNN architectures are removed to expose the deeper convolution layers. Three feature maps of with a spatial size of $\frac{w}{8} \times \frac{h}{8}$, $\frac{w}{16} \times \frac{h}{16}$ and $\frac{w}{32} \times \frac{h}{32}$ are extracted from the encoder. In this study, several feature extraction models like the Resnet50, Resnet152, DenseNet161, DenseNet201, EfficientNet_v2_l, EfficientNet_B5, ConvNext Base, ConvNext Large were tested. The models were chosen based on their performance on the imagenet classification task as stated in the introduction. The feature maps extracted from these models had different depths. From the resnet architectures, we extracted conv3_x, conv4_x and conv5_x layer. For the DenseNet architectures, which are composed of four blocks, we extracted feature maps from the DenseBlock 2, DenseBlock 3, and DenseBlock 4. For the ConvNext architecture, we extracted feature maps from the last three blocks of the model.

3.1.2 Transformer Encoder

The feature maps extracted from the backbone models are passed to three transformer encoders in order to enhance long range and contextual information. In the transformer encoder, a 1 x 1 convolution layer is used to reduce the computation complexity. The features from the last two layers are reduced to 768 dimensions while the other feature map is reduced from to 512 dimensions. Positional encoding is used to add positional awareness to the model before feeding the input to the transformer encoders. Absolute positional embedding which performs an elementwise addition to the input and a learnable matrix with the same shape as the input. Each transformer encoder contains two layers of multi-head self-attention and mlp blocks. The layers of the transformer encoder 1 and 2 have 12 attention heads, while transformer encoder 3 has 8 heads. The MLP block contains two layers with GELU activation function. Layer Normalization and residual connections are applied before and after each block respectively.

3.1.3 CNN decoder

The transformer encoder is followed by a CNN decoder in order to restore the original image resolution. The CNN decoder is a fully connected CNN network containing 7 blocks, which are used to compute pixelwise classification to predict the saliency map. Batch normalization and ReLU activation are used for blocks 1 to 6 and Sigmoid activation is used for block 7 after each 3 x 3 convolution operation. Given that the image size is down sampled by 32 by the decoder network, a 2 scale up sampling that uses nearest neighbour interpolation is performed on the feature map in the first five blocks to obtain a saliency map of the same size as the input. The up sampled feature map and the transformer output from the corresponding skip-connection are fused by an elementwise product.

3.2 DeepGazeIIE model

The model is composed of a backbone network made up of pre-trained CNN models, a readout network of 1×1 convolution blocks and a finalization block. The input image is first processed by a CNN backbone composed of an ensemble of ShapeNet-C, EfficientNet-B5, ResNext-50 and DenseNet-201 to extract deep activations which are then processed by a readout network. The readout network is composed of six blocks of 1×1 convolutions, a layernorm and a softplus function. The channel sizes of the convolutions are 8, 16, 1, 128, 16 and 1. The readout network finally outputs an image with a single channel that is then resized and blurred before a center bias is added to it and fed into a SoftMax to yield a two-dimensional fixation distribution. The weights of the backbone model are frozen during training while the readout network, the blur size and the center bias weights are trained. The structure of the model is shown in figure 2. The model is pre-trained on the SALICON dataset and fine tuned on the MIT1003 dataset. The model is evaluated on the 10-fold cross validation scheme on the MIT1003 dataset. The model is evaluated using metrics described in section 4.1.

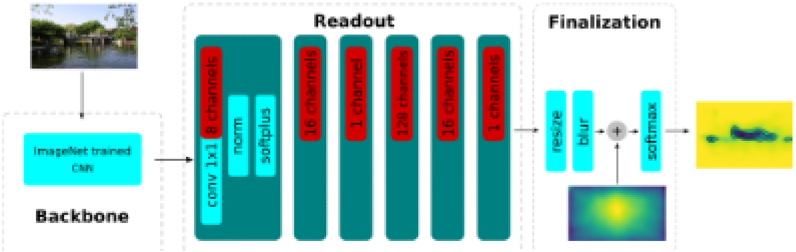

Figure 2: A diagram of the DeepGazeIIE architecture

4 Experiments, Results and Discussion

4.1 Datasets

Two benchmark saliency datasets were used to train and evaluate our proposed saliency model and variants.

- Salicon dataset contains 10,000 training, 5,000 and 5,000 testing images. The ground truth saliency maps for this dataset are obtained using mouse clicks unlike other datasets that use eye trackers.
- The MIT1003 dataset consists of 1003 natural indoor and outdoor images with eye tracking data from 15 observers.

4.2 Experiments

We carried out experiments to determine an appropriate model that can run on robots like Pepper given their computational power constraints. Both baseline models in this paper utilize backbone models pretrained on the ImageNet dataset. This is intended to pass on the information learned by the models from the larger dataset to another task. The backbone models were selected based on their performance on the ImageNet dataset. We also paired the models from one family for instance, we experimented with ResNet50 and ResNet152 to test the effect of the depth. The models used in our experiments are listed in table1 together with the layers extracted from the model. We developed the following questions for our ablation study in order to identify an architecture that was both highly accurate and computationally efficient.

- How does the performance of the model on the ImageNet benchmark correlate with its performance on saliency map prediction?
- How does depth affect the performance of the model in saliency map prediction?
- What is the effect of reducing the number of layers extracted from the backbone model and the number of transformer encoders?

Architecture	**Layer 1**	**Layer 2**	**Layer 3**
ResNet50	$512 \times 36 \times 48$	$1024 \times 18 \times 24$	$2048 \times 9 \times 12$
Resnet152	$512 \times 36 \times 48$	$1024 \times 18 \times 24$	$2048 \times 9 \times 12$
Densenet161	$768 \times 36 \times 48$	$2112 \times 18 \times 24$	$2208 \times 9 \times 12$
Densenet201	$512 \times 36 \times 48$	$1792 \times 18 \times 24$	$1920 \times 9 \times 12$
ConvNextLarge	$512 \times 36 \times 48$	$768 \times 18 \times 24$	$1536 \times 9 \times 12$
ConvNextBase	$512 \times 36 \times 48$	$512 \times 18 \times 24$	$1024 \times 9 \times 12$
EfficientNet_v2_L	$512 \times 36 \times 48$	$224 \times 18 \times 24$	$640 \times 9 \times 12$
EfficientNet_b5	$512 \times 36 \times 48$	$176 \times 18 \times 24$	$512 \times 9 \times 12$

Table 1: Table the models that backbone models and dimensions of extracted feature maps

In order to answer these questions, we re-implemented the baseline models and made changes to the backbone models to answer the questions stated above. Each of the models built was trained on the SALICON dataset described in section 4.1 for 30 epochs on Tesla T4 GPUs. The initial learning rate was set to 1e-4 and used a CosineAnnealing scheduler. We used the Adam optimizer to update the weights of the model. The batch size for training was set to 4. We saved and recorded the best performance achieved by the model.The models were then evaluated on the MIT1003 dataset.

4.3 Results

Architecture	ImageNet(5% acc)	CC	SIM	KLD
ResNet50	95.434	0.527	0.4974	0.9140
Resnet152	96.002	0.5387	0.5059	0.8652
Densenet161	96.56	0.6106	0.5705	0.8059
Densenet201	96.37	0.5388	0.5028	0.9348
ConvNextBase	96.87	0.8038	0.6651	0.5297
ConvNextLarge	96.976	0.8117	0.6663	0.5670
EfficientNet_v2_L	96.628	0.5318	0.3163	1.4386
EfficientNet_b5	97.788	0.5554	0.2949	1.6303

Table 2: Table shows the performance of the backbone models on the ImageNet benchmark, Correlation Coefficient, SIM and KL Divergence

Architecture	ImageNet(5% acc)	CC	SIM	KLD
ConvNextLarge	96.976	0.7443	0.6314	0.6345
Resnet152	96.002	0.5238	0.4936	0.8781
Densenet161	93.56	0.4976	0.4781	0.9798

Table 3: Table shows the performance of the TransalNet model with one Encoder Layer

Architecture	LL	AUC	NSS
ResNet50	0.6795	0.6729	0.9701
Resnet101	0.6825	0.6733	0.9723

Table 4: Table shows the performance of the DeepGazeII model with one Encoder Layer

Table 2 shows the performance of the TransalNet model with different backbone models. It can be seen from the results that models that performed better on the ImageNet benchmark also perform better on the visual saliency benchmark. This is consistent with the results reported by Kummerer et al[14]. This is with the exception of EfficientNet family of models. The ConvNext Large model had the best performance in terms of the correlation coefficient (CC) metric and the SIM metric, however, it has a worse performance than the ConvNextBase model in terms of the KL Divergence metric. The worst performance was by the ResNet50 model yet this is a ubiquitous backbone in saliency prediction models. It can be seen from the results that deeper models outperformed the shallower ones except the DenseNet models where the DenseNet161 model performed better than the DenseNet201 model.
Table 3 shows the performance of the model when only one of the layers shown in Table 1 is used. In all the cases we used Layer 1 which has a spatial size of 36×48. We also found, unsurprisingly, that extracting features from fewer number of layers resulted in faster inference but at the expense of reduced performance.
Table 4 shows the results obtained from the DeepGazeII model. It can be seen from the results that ResNet101 model performs better than the ResNet50 model. This is consistent with the results seen from the TransalNet model.

4.4 Discussion

The results from the TransalNet model tests show that models that performed better on the ImageNet benchmark consistently performed better in saliency prediction. This can be attributed to the better representation power that these models possess. Given the challenge of small datasets in visual saliency prediction, models pretrained on image classification tasks can be used as feature extractors. Attempts to train the backbone models in addition to the rest of the network resulted in poor results which we attribute to the model quickly overfitting to the small dataset. Hence it is recommended that the backbone models are frozen during training. EfficientNet model performance, however, did not follow the same trend as the other models. This is consistent with results reported by Kummerer et al[14]. This can be attributed to high levels of uncertainty that are associated with its architecture. More studies need to be done to understand the underlying cause. In terms of backbone model depth, DenseNet201 performs worse than DenseNet161 unlike all the other models where the deeper models outperform the shallower ones. The reduction in the number of feature maps extracted from the backbone model affect performances negatively because there is a reduction in the long-range information and the transformer encoder is less effective compared to when three feature maps are extracted and passed to three transformer encoder layers.

5 Conclusion

In this paper, we have re-implemented the DeepGazeII model and assessed the influence of different configurations of the backbone model. The results revealed that ImageNet performance can be used as an indicator for a better backbone for the visual saliency prediction task. We also concluded that decreasing the number of layers extracted from the backbone model would degrade performance, albeit reducing the training and inference time required for the model. In terms of saliency map prediction for robotics where we are seeking a balance between the two, we identified ConvNext_Base as a backbone model with the best trade-off. In our future work we are to investigate the effect of the other components of the model such as the transformer block and the decoder used for the TransalNet model. We would also like to do more experiments with the deepgazeII model and whether incorporating attention modules in its architecture would improve its performance.

References

[1] S. Treue, “Visual attention: the where, what, how and why of saliency,” Curr Opin Neurobiol, vol. 13, no. 4, pp. 428–432, Aug. 2003, doi: 10.1016/S0959-4388(03)00105-3.

[2] R. Cong et al., “Review of Visual Saliency Detection with Comprehensive Information,” 2018.

[3] M. Begum and F. Karray, “Visual attention for robotic cognition: A survey,” in IEEE Transactions on Autonomous Mental Development, Mar. 2011, pp. 92–105. doi: 10.1109/TAMD.2010.2096505.

[4] U. DigitalCommons, U. All Graduate Theses, and F. Xu, “Visual Saliency Estimation and Its Applications,” 2020. [Online]. Available: https://digitalcommons.usu.edu/etd/7820

[5] P. K. Pook, “Saliency in Human-Computer Interaction *,” 1996. [Online]. Available: www.aaai.org

[6] R. Cong et al., “Review of Visual Saliency Detection with Comprehensive Information,” 2018.

[7] A. Borji, “IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 1 Saliency Prediction in the Deep Learning Era: Successes, Limitations, and Future Challenges.” [Online]. Available: http://salicon.net

[8] L. Elazary and L. Itti, “Interesting objects are visually salient,” J Vis, vol. 8, no. 3, pp. 3–3, Mar. 2008, doi: 10.1167/8.3.3.

[9] K. A. Ehinger, B. Hidalgo-Sotelo, A. Torralba, and A. Oliva, “Modeling Search for People in 900 Scenes: A combined source model of eye guidance,” Vis cogn, vol. 17, no. 6–7, p. 945, Aug. 2009, doi: 10.1080/13506280902834720.

[10] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, “SUN: A Bayesian framework for saliency using natural statistics,” J Vis, vol. 8, no. 7, pp. 32–32, May 2008, doi: 10.1167/8.7.32.

[11] E. Vig, M. Dorr, and D. Cox, “Large-Scale Optimization of Hierarchical Features for Saliency Prediction in Natural Images.”

[12] L. Theis and M. Bethge, “DEEP GAZE I: BOOSTING SALIENCY PREDICTION WITH FEATURE MAPS TRAINED ON IMAGENET,” 2015.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Adv Neural Inf Process Syst, vol. 25, 2012, Accessed: Mar. 29, 2023. [Online]. Available: http://code.google.com/p/cuda-convnet/

[14] M. Kümmerer, T. S. A. Wallis, and M. Bethge, "DeepGaze II: Reading fixations from deep features trained on object recognition," Oct. 2016, Accessed: Mar. 29, 2023. [Online]. Available: https://arxiv.org/abs/1610.01563v1

[15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, Sep. 2014, Accessed: Mar. 29, 2023. [Online]. Available: https://arxiv.org/abs/1409.1556v6

[16] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "SALICON: Saliency in Context," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Oct. 2015, pp. 1072–1080. doi: 10.1109/CVPR.2015.7298710.

[17] "MIT/Tuebingen Saliency Benchmark." https://saliency.tuebingen.ai/ (accessed Mar. 29, 2023).

[18] A. Linardos, M. Kümmerer, and M. Bethge, "DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling."

[19] S. S. S. Kruthiventi, K. Ayush, and R. V. Babu, "DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations," Oct. 2015, [Online]. Available: http://arxiv.org/abs/1510.02927

[20] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "A Deep Multi-Level Network for Saliency Prediction," Proceedings - International Conference on Pattern Recognition, vol. 0, pp. 3488–3493, Sep. 2016, doi: 10.1109/ICPR.2016.7900174.

[21] N. Liu and J. Han, "A Deep Spatial Contextual Long-term Recurrent Convolutional Network for Saliency Detection," IEEE Transactions on Image Processing, vol. 27, no. 7, pp. 3264–3274, Oct. 2016, doi: 10.1109/TIP.2018.2817047.

[22] S. Jia and N. D. B. Bruce, "EML-NET:An Expandable Multi-Layer NETwork for Saliency Prediction."

[23] W. Wang and J. Shen, "Deep Visual Attention Prediction," May 2017, doi: 10.1109/TIP.2017.2787612.

[24] J. Lou, H. Lin, D. Marshall, D. Saupe, and H. Liu, "TRANSALNET: TOWARDS PERCEPTUALLY RELEVANT VISUAL SALIENCY PREDICTION A PREPRINT." [Online]. Available: https://github.com/

[25] T. Judd, F. Fr´, F. Durand, and A. Torralba, "A Benchmark of Computational Models of Saliency to Predict Human Fixations," Jan. 2012, Accessed: Mar. 29, 2023. [Online]. Available: https://dspace.mit.edu/handle/1721.1/68590

[26] A. Borji, D. N. Sihite, and L. Itti, "Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study," IEEE Transactions on Image Processing, vol. 22, no. 1, pp. 55–69, 2013, doi: 10.1109/TIP.2012.2210727.

[27] R. J. Peters, A. Iyer, L. Itti, and C. Koch, "Components of bottom-up gaze allocation in natural images," Vision Res, vol. 45, no. 18, pp. 2397–2416, Aug. 2005, doi: 10.1016/J.VISRES.2005.03.019.