Original Paper

GreenSaliency: A Lightweight and Efficient Image Saliency Detection Method

Zhanxuan Mei[*], Yun-Cheng Wang and C.-C. Jay Kuo

University of Southern California, USA

ABSTRACT

Image saliency detection is crucial in understanding human gaze patterns from visual stimuli. The escalating demand for research in image saliency detection is driven by the growing necessity to incorporate such techniques into various computer vision tasks and to understand human visual systems. Many existing image saliency detection methods rely on deep neural networks (DNNs) to achieve good performance. However, the high computational complexity associated with these approaches impedes their integration with other modules or deployment on resource-constrained platforms, such as mobile devices. To address this, we propose a novel image saliency detection method named GreenSaliency, which has a small model size, minimal carbon footprint, and low computational complexity. GreenSaliency can be a competitive alternative to the existing deep-learning-based (DL-based) image saliency detection methods with limited computation resources. GreenSaliency comprises two primary steps: 1) multi-layer hybrid feature extraction and 2) multi-path saliency prediction. Experimental results demonstrate that GreenSaliency achieves comparable performance to the state-of-the-art DL-based methods while possessing a considerably smaller model size and significantly reduced computational complexity.

*Corresponding author: Zhanxuan Mei, zhanxuan@usc.edu

1   Introduction

The attention mechanism within the human visual system indicates the regions
humans are interested in within the observed scenes [15]. Such an attention
mechanism is often studied through the analysis of human eye movements
recorded via gaze-tracking technology during the presence of visual stimuli,
such as images. These fixations, collected from eye-tracker data, indicate the
most compelling locations within a scene. Typically, a saliency map is derived
from fixation maps through convolution with a Gaussian kernel, enhancing
the representation of salient locations. This saliency map, constructed at
the pixel level, represents regions of attention within the stimuli. Image
saliency detection is used to detect the most informative and conspicuous
fixations within visual scenes to emulate the attention mechanisms exhibited
by human eyes. Fixation maps, obtained through human subject studies,
and their corresponding saliency maps generated from these fixation maps
are commonly regarded as ground truths (GTs), utilized in the training and
evaluating of image saliency detection models. Image saliency detection
research predominantly falls within two categories: 1) human eye fixation
prediction, which involves the prediction of human gaze locations on images
where attention is most concentrated [3], and 2) salient object detection
(SOD) [38], which aims to identify salient object regions within an image.
This paper focuses on the former, which predicts the human gaze from visual
stimuli.

Comprehending human gaze patterns within visual stimuli is essential
in modeling visual attention. Image saliency detection, serving as either a
preparatory step or a guiding principle in image processing, facilitates the iden-
tification of regions likely to command initial human attention. This insight
finds applications across diverse domains. In saliency-driven perceptual image
compression methodologies [31], saliency cues inform the allocation of coding
resources, directing more bits towards salient regions while economizing on
less crucial areas. Recognizing the non-uniform distribution of visual attention
across image regions, local image saliency detection has been integrated into no-
reference image quality assessment techniques [36] to capture spatial attention
discrepancies, thereby enhancing quality prediction accuracy. Furthermore,
investigations into saliency within visual stimuli yield valuable insights into
the cognitive mechanisms governing human visual processing and attentional
allocation, thus informing strategies for data augmentation [34] and enhanc-
ing model interpretability through saliency-guided training procedures [14].
Overall, image saliency detection plays a fundamental role in various visual
processing applications and human perception applications. It is an essential
area of research and development in computer vision and related fields.

During earlier epochs, conventional methodologies for image saliency de-
tection primarily followed a bottom-up approach [15, 12], where the features

are not required to be pre-computed, thus offering versatility across a wide range of applications. Recent advancements have witnessed the increase of DL-based saliency detection methods [29, 7], which have demonstrated remarkable efficacy of feature representation capabilities of Convolutional Neural Networks (CNNs), driven by large-scale image saliency datasets [19, 1, 17]. Given the limited volume of data within the saliency domain compared to some of the more prominent computer vision tasks, transfer learning emerges as a pivotal mechanism for enhancing image saliency detection. Drawing inspiration from the massive success of deep convolutional models such as VGGNet [33] and ResNet [13] in classification tasks, particularly on benchmarks like ImageNet [8], researchers have leveraged transfer learning to introduce pre-trained features from large-scale and external datasets into the image saliency detection methods. As a result, these methods achieve superior performance due to the expressive image features. Integrating these large pre-trained models into mobile or edge devices is economically burdensome due to their substantial computational demands and expansive model sizes. Given that image saliency detection methods typically serve as preliminary stages in image processing pipelines, allocating such considerable computational resources may be deemed unwarranted.

To address these challenges, we propose a lightweight image saliency detection method called GreenSaliency in this work. GreenSaliency features a smaller model size and lower computational complexity while achieving comparable performance against DL-based methods. Compared to other green-learning-based methods presented before [24], the novelty of GreenSaliency lies in two completely new modules: 1) multi-layer hybrid feature extraction and 2) multi-path saliency prediction. Without backpropagation as done in neural networks, the proposed method employs an one-pass pipeline to reduce the model size (i.e., the number of model parameters). Compared with conventional image saliency detection methods, the multi-layer hybrid feature extraction module relies on something other than human prior knowledge to extract features. It combines unsupervised feature generation (via hierarchical Saab transforms [25]) with supervised feature selection (using the Relevant Feature Test, RFT [37]) to efficiently extract task-relevant features. The supervised feature selection module is a data-driven approach, which is achieved automatically without human intervention. Additionally, the data-driven multi-path saliency prediction module leverages features from diverse layers that encapsulate information from disparate receptive fields. This diversity enhances the model's ability to accommodate the unique characteristics of different input images. Consequently, the one-pass pipeline of GreenSaliency establishes a balance between conventional and DL-based methods, offering an efficient compromise between prediction accuracy and model complexity. We conduct experiments on two popular image saliency datasets and show that GreenSaliency can offer satisfactory performance in five evaluation met-

rics while demanding significantly small model sizes, short inference time, and low computational complexity. This paper has the following two main contributions.

- Introduction of a novel method for saliency detection termed Green-Saliency, which is characterized by a transparent and modularized design. This method features a feedforward training pipeline distinct from the utilization of DNNs. The pipeline encompasses multi-layer hybrid feature extraction and multi-path saliency prediction.

- Execution of experiments on two distinct datasets to assess the predictive capabilities of GreenSaliency. The findings illustrate its superior performance compared to conventional methods and its competitive standing against early-stage DL-based methods. Furthermore, GreenSaliency exhibits efficient prediction performance compared to state-of-the-art methods employing pre-trained networks on external datasets while necessitating a significantly smaller model size and reduced inference complexity.

2   Related Work

2.1   Conventional Image Saliency Detection Methods

In the early stage of image saliency detection research, conventional methodologies predominantly adhered to a bottom-up approach. For example, in Itti *et al.* [15], a visual attention system was inspired by early primate visual systems' behavioral and neuronal paradigms. This system combines multi-scale image features into a unified topographical saliency map, subsequently utilized by a dynamical neural network to sequentially prioritize attended locations based on decreasing saliency levels. This process, informed by inputs from early visual processes, effectively simulates bottom-up attention mechanisms observed in primates. Another prominent bottom-up model is the Graph-Based Visual Saliency (GBVS) framework [12], comprising two sequential steps: 1) the formation of activation maps on specific feature channels followed by their normalization to accentuate salient features and 2) enabling combination with other maps. Conversely, SUN [39] introduces a Bayesian framework where bottom-up saliency naturally emerges as the self-information of visual features. In contrast, overall saliency, incorporating top-down and bottom-up influences, arises as the pointwise mutual information between features and the target during target search tasks. Furthermore, Kienzle *et al.* [20] proposes an image saliency model derived directly from human eye movement data, characterized by a nonlinear mapping from image patches to real values. This model is trained to produce positive outputs for fixated regions and negative outputs for

randomly selected image patches. In conclusion, while conventional saliency detection methods are noted for their efficiency, they predominantly focus on evaluating contrast based on low-level features. These methods often lack supervision from human feedback or subjective datasets, which limits their effectiveness. This lack of sophisticated feature analysis and the absence of adaptive learning from large-scale annotated data restrict their applicability, especially in scenarios requiring understanding of visual content.

2.2   DL-based Image Saliency Detection Methods

In recent years, deep-learning-based methodologies for saliency detection [29, 7] have achieved notable performance owing to the advanced feature representation capabilities inherent in CNNs. An early work in this domain was the emergence of eDN [35], which introduced hierarchical feature learning principles to visual saliency. Additionally, Pan *et al.* [29] introduced the concept of training a shallow CNN architecture from scratch, marking the inception of end-to-end CNN models tailored explicitly for image saliency detection tasks. However, a significant impediment encountered in leveraging deep learning frameworks for image saliency detection lies in the scarcity of available data, primarily stemming from the laborious and cost-intensive nature of collecting fixation data. Furthermore, the sensitivity of image saliency to image transformations poses another challenge, significantly constraining potential data augmentations [3]. Consequently, transfer learning has emerged as a pivotal strategy for refining image saliency detection performance. Inspired by the remarkable success of deep convolutional networks in classification tasks, particularly exemplified by benchmarks like ImageNet, most high-performing saliency models have embraced transfer learning, typically relying on pretrained models on ImageNet as the foundation. Noteworthy milestones in this trajectory include the pioneering work of DeepGaze I [23], which has since evolved into DeepGaze II [22], leveraging the VGG19 architecture. Following this paradigm, DeepGaze IIE [27] was proposed, demonstrating the attainment of robust confidence calibration on unseen datasets through the principled amalgamation of multiple backbone architectures. Additionally, the Saliency Attentive Model (SAM) [7] introduced a novel Attentive Convolutional Long Short-Term Memory (LSTM) mechanism, sequentially directing attention to diverse spatial regions within a feature stack to enhance image saliency detection. EML-NET [16] introduced a scalable approach for integrating multiple deep convolutional networks of varying complexities as encoders for saliency-relevant features. Meanwhile, UNISAL [11] unified saliency prediction across both image and video modalities, leveraging the entirety of available saliency prediction datasets to enrich analysis outcomes. UNISAL adopted MobileNet-V2 [32], a model with lightweight and efficient architecture, as its backbone encoder for the simplicity and reduced computational demands.

Various models have been developed employing intricate, deep architectures or extending existing ones that have demonstrated efficacy in other domains. Nonetheless, all these models have uniformly embraced transfer learning, initializing their architectures by pre-training them on outside datasets. For instance, SalGAN [30] introduces a generative adversarial model tailored for image saliency detection, comprising two interconnected networks: one responsible for generating saliency maps from raw pixel data of input images. At the same time, the other discriminates between predicted saliency maps and ground truth. Similarly, GazeGAN [3] leverages a modified U-Net architecture as its generator, amalgamating classic "skip connections" with a novel "center-surround connection" (CSC) module. Departing from conventional feedforward architectures for saliency prediction, FBNet [9] integrates feedback convolutional connections to establish links between high-level blocks and low-level layers, thereby enhancing feature representation. Subsequently, SalFBNet [10] introduces a lightweight feedback-recursive convolutional framework that enhances saliency detection by integrating feature pathways from high-level blocks to low-level layers. Additionally, Ding *et al.* [10] pioneers creating a large-scale Pseudo-Saliency dataset, addressing concerns regarding data scarcity in image saliency detection.

Despite the high performance achieved by DL-based methods in saliency detection, many of them rely on models pre-trained on extensive external datasets. Integrating these large pre-trained models into mobile or edge devices presents significant economic challenges due to their considerable computational demands and substantial model sizes. Additionally, for the methods that require training or fine-tuning of models with a large number of parameters entails considerable offline costs. These factors can severely limit the deployment and scalability of DL-based saliency detection methods in resource-constrained environments.

2.3   Green Machine Learning

Recently, the concept of green learning [24] has emerged as a novel machine learning paradigm to develop efficient models with a reduced carbon footprint. These models are distinguished by their compact sizes and minimized computational complexities during the training and inference stages. An additional benefit lies in their mathematical transparency, facilitated by a modularized design principle. In contrast to DNNs, which rely on backpropagation for iterative parameter updates, green learning adopts a one-pass training pipeline comprising three sequentially cascaded modules: 1) unsupervised representation learning, 2) supervised feature learning, and 3) supervised decision learning. The unsupervised representation learning module employs a hierarchical approach utilizing multiple Saab transforms [25, 6], thereby broadening the spectrum of candidate representations to enhance the discriminant capa-

bilities of the subsequent supervised feature learning module. Subsequently, the Relevant Feature Test (RFT) [37] is employed to discern the most powerful features from a plethora of candidate representations obtained from the preceding module. Finally, a regressor, such as XGBoost [5], is trained to map the feature space to the target label space. Green Learning techniques have found applications across diverse domains, including deepfake detection [4], blind image quality assessment [28], image generation [26], etc. In this paper, we propose a lightweight saliency detection method inspired by the principles of green learning.

3 Proposed GreenSaliency Method

An overview of the proposed GreenSaliency method is depicted in Figure 1. As shown in the figure, GreenSaliency has a modularized solution that consists of two modules: 1) multi-layer hybrid feature extraction and 2) multi-path saliency prediction. They are elaborated below.
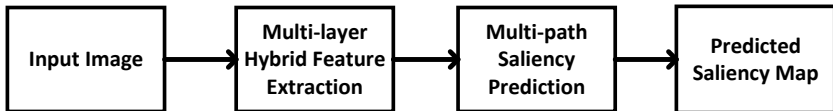

Figure 1: An overview of the proposed GreenSaliency method.

3.1 *Multi-layer Hybrid Feature Extraction*

Figure 2 shows the multi-layer hybrid feature extraction pipeline. In this section, we commence by delineating the multi-layer structure. Each layer comprises analogous components, such as spatial feature extraction and two Saab transforms, aimed at extracting hybrid features for each layer. Subsequently, the details of the spatial feature extraction module and the saab transform are introduced.

3.1.1 *Multi-layer Structure*

A hierarchical multi-layer structure is employed to capture features spanning from local to global contexts across different receptive fields. This structure comprises five layers denoted as d4, d8, d16, d32, and d64, respectively. Within each layer, except d4 and d64, a spatial feature extraction module accompanied by two Saab transform modules exists. The spatial feature extraction module
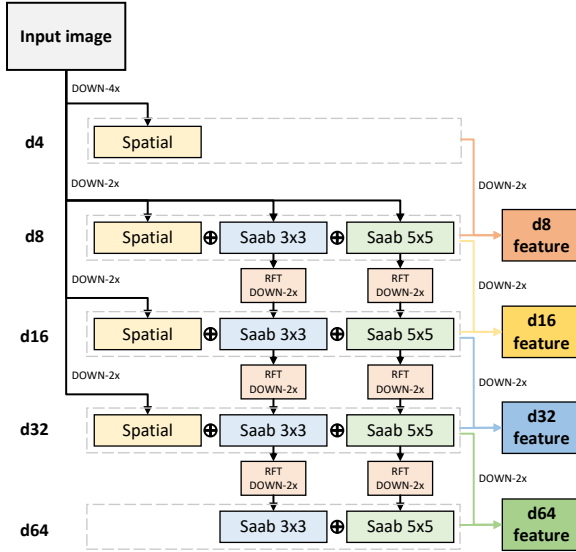
Figure 2: Multi-layer hybrid feature extraction pipeline.

facilitates the direct extraction of spatial features from input images that have undergone downsampling. Notably, the prefix "d" within the naming of each layer signifies downsampling, wherein, for instance, d4 indicates that spatial features at this layer are computed based on input images downsampled by a factor of 4. Specifically, we use the Lanczos downsampling method. The two Saab transform modules, employing kernel sizes of 3×3 and 5×5, respectively, compute Saab coefficients derived from those of the previous layer. This recursive process enables extracting high-level features as the layers progress deeper. Further elaboration on the spatial feature extraction module and Saab transform mechanisms is provided in Section 3.1.2 and Section 3.1.3, respectively.

The Relevant Feature Test (RFT) is employed to discern the most powerful coefficients or channels meriting propagation to the subsequent layer after obtaining Saab coefficients from the two Saab transform modules within the current layer. Section 3.1.4 provides a detailed exposition of the RFT methodology. Additionally, the selected coefficients undergo downsampling at a ratio of 2 to mitigate the computational burden associated with processing coefficients in the subsequent layer. Within each layer, the computed spatial features and Saab coefficients derived from the two Saab transform modules are concatenated to form hybrid features. Subsequently, these hybrid features

are combined with hybrid features from other layers to compose sets of features utilized for image saliency prediction. Illustrated in Figure 2, four sets of features are discernible, denoted as d8 features, d16 features, d32 features, and d64 features, respectively. Each set encompasses extracted features from two layers. For instance, d8 features comprise hybrid features derived from layers d4 and d8. These four sets of features are transmitted to the multi-path saliency prediction module, as elaborated in Section 3.2, to predict the corresponding saliency map or residual.

3.1.2 Spatial Feature Extraction

In contrast to features computed by the Saab transform, which are propagated from shallow layers to deeper layers, the spatial features extracted in each layer are calculated directly from down-sampled input images and are not transmitted to deeper layers. The spatial features encapsulate local and spatial information inherent in the input images, while features computed by the Saab transform primarily capture spectral information. The rationale for not propagating spatial features to deeper layers is threefold. First, the edge and location features are low-level features. No further propagation is needed as their utility stays in the initial layers. Second, the local Saab features complement high-level Saab features recursively computed from two Saab transform modules, providing a more nuanced feature set at each layer. Third, propagating spatial features to deeper layers would introduce additional computational complexity, contradicting the design goal of maintaining a lightweight and low-complexity architecture.

Spatial features consist of three primary components: 1) local Saab features, 2) edge features, and 3) location features. The local Saab features are derived by implementing two sequential cascade Saab transforms. Distinguished from the two Saab transforms within the same layer, these cascade Saab transforms are characterized by smaller kernel sizes in the spatial domain, specifically 2×2 and 3×3. This distinction in kernel sizes is attributed to their primary focus on capturing and emphasizing local information within the directly downsampled input images. Utilizing smaller kernel sizes enables these cascade Saab transforms to effectively highlight and extract intricate details locally, contributing to the overall feature representation and analysis. Concerning edge features, they are derived utilizing the Canny edge detector [2], a widely employed method for detecting edges in images. Numerous prior investigations have established that objects near an image's center garner greater attention from observers [19]. This observation suggests that locations proximal to the image center are more likely to exhibit saliency than those farther away. A Gaussian distribution can effectively model this empirical observation. Let (c_x, c_y) represent the center coordinates of an image; subsequently, the location

feature $f(x, y)$ at coordinates (x, y) can be expressed as a Gaussian map defined by

$$f(x, y) = exp\left(-\frac{(c_x - x)^2 + (c_y - y)^2}{\sigma^2}\right), \tag{1}$$

where σ is a pre-defined parameter.

3.1.3   Subspace Approximation with Adjusted Bias (Saab) Transform

The initial step in the processing pipeline involves partitioning the input images into overlapping blocks, typically of sizes 3×3 or 5×5, followed by applying the Saab transform [25]. The Saab transform, a principal component analysis (PCA) method with mean-removal, distinguishes itself by incorporating an additional bias vector. Within the framework of the Saab transform, a constant-element kernel is utilized to compute the average value of image patches, commonly known as the DC (Direct Current) component. Subsequently, PCA is applied to these patches post-removal of the computed mean, yielding data-driven AC (Alternating Current) kernels. Applying these AC kernels on individual patches leads to extracting AC coefficients associated with the Saab transform. For instance, an input cuboid has dimensions $(H \times W) \times C$, where H, W, and C denote the height, width, and number of channels, respectively. To execute a 3×3 Saab transform, the input cuboid is initially subdivided into overlapping cuboids of size $(3 \times 3) \times C$, with a stride of 1 and padding applied as necessary. Then, these cuboids are flattened into 1-dimensional vectors, each possessing a length of $9C$. The mean values of these vectors are computed to yield the DC channel, while the mean-removed vectors undergo PCA transform, resulting in the generation of $9C - 1$ AC channels. Following the computation of the 3×3 Saab transform, the input cuboid, originally of size $(H \times W) \times C$, transforms to dimensions $(H \times W) \times 9C$, preserving the spatial resolution and featuring $9C$ Saab coefficient channels.

In the feature extraction pipeline context, a multi-layer pipeline comprising four successive Saab transforms, located in d8, d16, d32, and d64, is adopted to decorate the DC coefficients and generate higher-level representations. This multi-layer pipeline facilitates the transformation of input data into a more informative and discriminative feature space, thereby enhancing the effectiveness of subsequent processing steps.

3.1.4   Relevent Feature Test (RFT)

The objective of feature selection is to identify highly discriminative features from a diverse candidate set of extracted features. To accomplish this task, a robust technique known as the Relevant Feature Test (RFT) [37] is employed. RFT entails dividing a feature dimension into two left and right segments

and evaluating the total mean-squared error (MSE) between them. The resultant approximation error is the RFT loss function, with a smaller RFT loss indicating a more informative feature dimension. Given a dataset comprising N data samples and P features, each feature dimension, denoted by f_i where $1 \leq i \leq P$, each feature dimension possesses a minimum and maximum range of f_{min}^i and f_{max}^i, respectively. The deployment of RFT involves three distinct steps, delineated as follows.

- Training Sample Partitioning. The primary objective entails identifying the optimal threshold, f_{op}^i, within the range $[f_{min}^i, f_{max}^i]$, facilitating the partitioning of training samples into two subsets: S_L^i and S_R^i. If the value of the ith feature, x_n^i, for the nth training sample x_n is lower than f_{op}^i, then x_n is assigned to S_L^i; otherwise, x_n is allocated to S_R^i. To refine the search space for f_{op}^i, the entire feature range, $[f_{min}^i, f_{max}^i]$, is divided into B uniform segments, and the optimal threshold is sought among $B - 1$ candidates.

- RFT Loss Measured by Estimated Regression MSE. Denoting the regression target value as y, y_L^i, and y_R^i represent the mean target values in S_L^i and S_R^i, respectively. These mean values are the estimated regression values for all samples in S_L^i and S_R^i. The RFT loss is defined as the summation of the estimated regression MSEs of S_L^i and S_R^i, given by

$$R_t^i = \frac{N_{L,t}^i R_{L,t}^i + N_{R,t}^i R_{R,t}^i}{N}, \tag{2}$$

where $N_{L,t}^i$, $N_{R,t}^i$, $R_{L,t}^i$, and $R_{R,t}^i$ represent the sample numbers and estimated regression MSEs in subsets S_L^i and S_R^i, respectively. Each feature f^i is characterized by its optimized estimated regression MSE over a set, T, of candidate partition points:

$$R_{op}^i = \min_{t \in T} R_t^i. \tag{3}$$

- Feature Selection based on the Optimized Loss. The optimized estimated regression MSE value, R_{op}^i, is computed for each feature dimension, f_i. These values are subsequently arranged in ascending order, reflecting the relevance of each feature dimension. A lower R_{op}^i value denotes a higher relevance of the i-th-dimensional feature, f^i.

After computing the R_{op}^i value for each feature dimension, f^i, representation indices i are arranged based on their MSE values in ascending order. Within the multi-layer hybrid feature extraction pipeline context, RFT is employed between two consecutive layers to diminish the feature dimensionality from the shallow layer onwards. The RFT outcomes of the 3×3 Saab coefficients

computed in layers d8, d16, and d32 are shown in Figure 3. In these figures, discernible elbow points are observed in the curves of the RFT outcomes, especially for the results from d16 and d32. The dot lines represent the number of selected features. Consequently, we select the top-ranking features, representing the most influential ones, for propagation to deeper layers. The primary rationale behind incorporating RFT between successive layers lies in the recognition that not all channels of coefficients necessitate processing at deeper layers, primarily due to the substantial computational complexity entailed. Hence, RFT identifies the most pertinent coefficients or channels that warrant deeper layer-level processing.
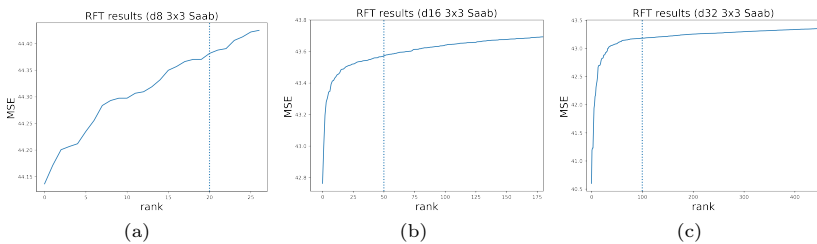

Figure 3: RFT results of 3×3 Saab coefficients from three layers: (a) d8, (b) d16, and (c) d32.

3.2   Multi-path Saliency Prediction

Following the extraction of features from the multi-layer hybrid feature extraction module, these features are utilized for multi-path saliency prediction, as delineated in Figure 4. Four distinct paths are initiated, corresponding to four layers: d8, d16, d32, and d64, each tasked with predicting its corresponding saliency map. Within each path, two conditions are considered: saliency map prediction and saliency residual prediction, whose details are expounded upon in Section 3.2.1. Upon obtaining predicted saliency maps from the four paths, they are aggregated within the ensemble module and subjected to post-processing to yield the final predicted saliency map. Detailed discussions on the ensembles and post-processing modules are provided in Section 3.2.2 and Section 3.2.3, respectively.

The rationale for adopting multi-path saliency prediction instead of a singular path employing all features stems from the recognition that features extracted from diverse layers encapsulate information from disparate receptive fields. Such diversity in receptive fields proves advantageous, catering to the varying characteristics of different input images. In contrast, a single-path approach that amalgamates features from all layers into one predictive model tends to produce a singular predicted saliency map. It constrains the model's
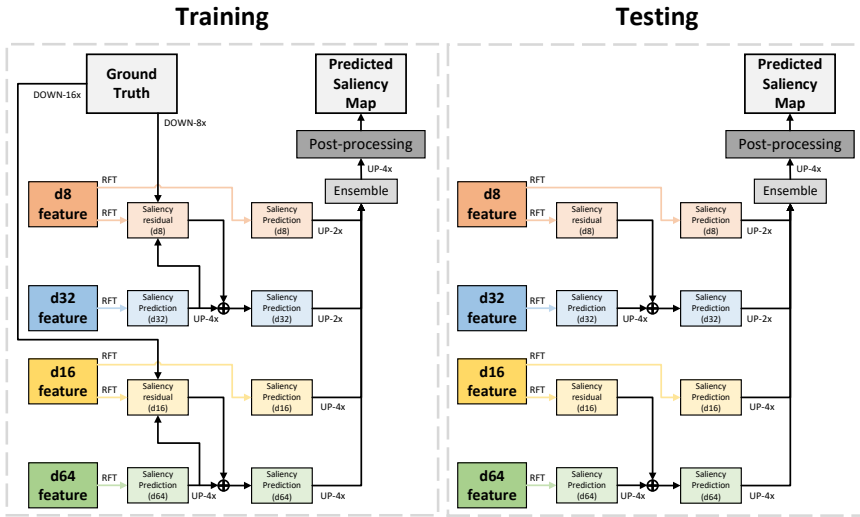
Figure 4: Multi-path saliency prediction pipeline.

adaptability and versatility, limiting its ability to tailor its response to specific input images. The experiments that compare the performance of multi-path versus single-path predictions are given in Section 4.4.

In the context of the multi-layer Saab transform, the features utilized for predicting the saliency map in layers d64 and d32 are relatively high-level features, while those in layers d16 and d8 are low-level features. High-level features, characterized by a large receptive field, concentrate on capturing the overall structure or shape of the entire saliency map rather than focusing on small objects or details. Conversely, low-level features exhibit greater sensitivity towards smaller objects while potentially overlooking the broader context of the scene. Figure 5 demonstrates the efficacy of utilizing distinct feature sets, where predicted saliency maps derived solely from specific feature sets are showcased. In the first two rows, predictions based on d8 and d16 features outperform those based on d32 and d64 features, attributed to the high-level features in d32 and d64 inadequately capturing the relatively small human subjects in these images. However, in the third row, high-level features excel in delineating the shape of the cat in the center. In contrast, predictions based on d8 and d16 features primarily focus on the top-right corner, illustrating the nuanced interplay between feature levels and their corresponding receptive fields. Section 4.4 illustrates more experimental results of different layers.
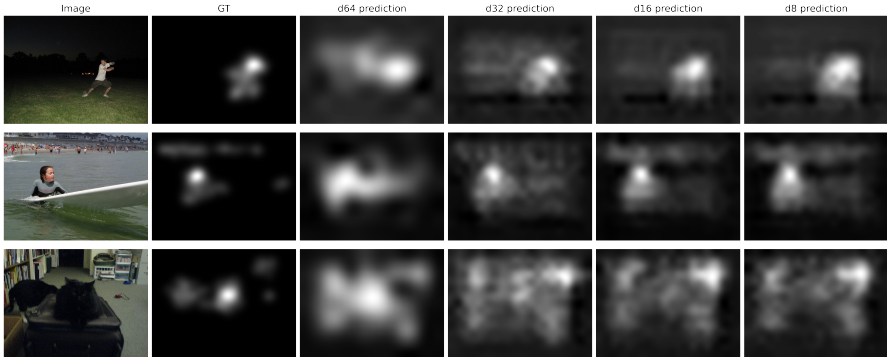
Figure 5: Prediction results from different layers.

3.2.1 Saliency Map Prediction and Saliency Residual Prediction

In each path, leveraging the set of hybrid features extracted from the multi-layer hybrid feature extraction module, an initial step involves utilizing these features to predict their respective saliency maps. For instance, in the path utilizing d64 features, the labels correspond to ground truth saliency maps down-sampled by a factor of 64. Subsequently, an XGBoost regressor is trained to perform pixel-wise prediction, mapping the feature domain to the saliency map domain. Following the direct utilization of extracted hybrid features for saliency map prediction, the predicted saliency maps from d8 and d16 are transmitted to the ensemble module. In contrast, those from d32 and d64 undergo saliency residual prediction. The saliency residual computation determines the disparity between the up-sampled predicted saliency map in the current layer and the ground truth within a shallower layer. For instance, given the predicted saliency map in d64, the residual is calculated between the up-sampled d64 saliency map and the ground truth in d16. Subsequently, d16 features train and predict this saliency residual using another regressor. The resultant predicted saliency residual in d16 is then added to the up-sampled predicted saliency map in d64 to form the final predicted saliency map originating from d64.

In summary, d64 and d32 features exclusively contribute to predicting their corresponding saliency maps, while d16 and d8 features predict both saliency maps and saliency residuals. The rationale underlying the saliency residual prediction lies in the observation that d64 and d32 features primarily capture high-level features, resulting in the comparatively lower resolution of their predicted saliency maps. Consequently, d8 and d32 features are leveraged to refine and calibrate the predicted saliency maps derived from d16 and d64, respectively. This strategy ensures enhanced precision and fidelity across multiple layers of predicted saliency maps.

Within the pipeline of multi-path saliency prediction, RFTs are employed to identify the most influential features for saliency map prediction and residual prediction. It is noteworthy that, for a given set of features, distinct RFTs are utilized to select disparate subsets of features for saliency map prediction and saliency residual prediction. This distinction arises due to the differing labels associated with these two predictions. The label for the former prediction corresponds to the downsampled ground truth. In contrast, the label for the latter prediction represents the residual between the predicted saliency map and the ground truth. The statistical distributions of the selected features employed for predicting saliency maps and saliency residuals on layers d16 and d8 are depicted in Figure 6. Despite some overlapping observed between the two distributions, several distinct features are selected for the two types of predictions. Given the label disparity between these two prediction tasks, selecting different features validates the necessity of employing separate RFTs for the same set of features.
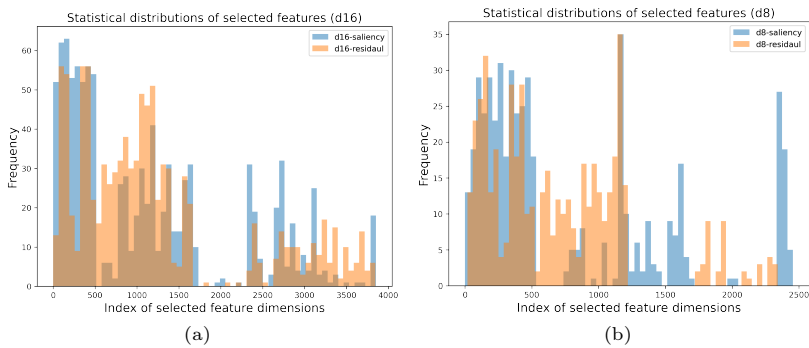

Figure 6: Distribution of RFT results in: (a) d16, and (b) d8 layers.

3.2.2 Ensembles

An ensemble module is deployed to integrate the four predicted saliency maps spanning from d64 to d8. Within this module, the first step involves upsampling the four predicted saliency maps to match the resolution size of d4 and concatenating them. Subsequently, for each pixel location, the neighboring saliency values from the four saliency maps within a 5×5 block centered at this location are collated, yielding a 100-dimensional vector. Ultimately, an XGBoost regressor is trained using labels derived from downsampled ground truth by a factor of 4, facilitating the prediction of the saliency map in the d4 layer.

3.2.3   Post-processing

Upon obtaining the fused predicted saliency maps in d4 and upsampling them to align with the resolution of the input images, several post-processing operations are implemented to furnish the final predicted saliency maps. These operations entail three sequential steps. Initially, a portion of small saliency values is filtered out, as their presence after ensembling signifies a lack of confidence in prediction. Eliminating these small predicted saliency values aids in reducing false positive predictions. Subsequently, a small Gaussian filter, with dimensions of 10×10, is applied to the entire image to enhance its smoothness. Finally, normalization is conducted on the entire image to align its distribution with the ground truth, thereby refining the accuracy and reliability of the predicted saliency maps.

4   Experiments

4.1   Experimental Setup

4.1.1   Datasets

Experiments were conducted utilizing the SALICON [17] and MIT300 [18] datasets, with benchmarks in the domain of image saliency detection. The SALICON dataset, widely recognized within the research community, comprises a substantial collection of images, including 10,000 images in the training set, 5,000 in the validation set, and an additional 5,000 in the testing set. On the other hand, the MIT300 dataset comprises 300 testing images, while the ground truth for this dataset is not publicly available. To mitigate this limitation, the model was trained on the MIT1003 [19] dataset and subsequently evaluated on the MIT300 dataset, a practice commonly adopted by various benchmarks. The MIT1003 dataset, consisting of 1,003 images, features ground truth annotations obtained through eye-tracking devices from 15 observers. Similarly, the MIT300 dataset was constructed following comparable procedures, drawing from the same image repositories and annotation methodologies.

4.1.2   Evaluation Metrics

To evaluate the performance of our saliency model comprehensively, we employ widely recognized metrics, including the linear correlation coefficient (CC), area under the receiver operating characteristic curve (AUC-J), shuffled AUC (s-AUC), normalized scanpath saliency (NSS), and similarity (SIM). These metrics offer multifaceted insights into model efficacy in predicting eye fixation patterns. Notably, higher values of CC, AUC, sAUC, NSS, and SIM signify superior performance of the saliency model. For a more detailed explanation

of these metrics and their relevance to saliency prediction, a comprehensive study is conducted in Kummerer *et al.* [21].

4.1.3   Implementation Details

The initial step in our experimental setup involves resizing the input images to dimensions of $(480 \times 640) \times 3$, utilizing the YUV color format. Subsequently, we implement the multi-layer hybrid feature extraction process. Specifically, after computing the Saab coefficients from Saab 3×3 in layers d8, d16, and d32, the RFTs are employed to select 20, 50, and 100 coefficients, respectively, from each layer, which are then forwarded to the subsequent layer. A similar approach is applied to the Saab 5×5 coefficients. In the multi-path saliency prediction phase, RFTs select various features from different layers: 500 from d8 features, 500 from d16 features, 1,000 from d32 features, and 1,000 from d64 features, respectively. Regarding the training and testing partitioning, we adhere to the official splitting protocol for the SALICON dataset. For the MIT1003 dataset, we allocate 900 images for training and 103 for validation. Subsequently, testing is conducted on the 300 images comprising the MIT300 dataset. All experiments are executed on a server equipped with an Intel(R) Xeon(R) E5-2620 CPU, ensuring consistency and reliability across computations.

4.2   Experimental Results

4.2.1   Benchmarking Methods

We conducted a comprehensive performance evaluation of GreenSaliency compared to eleven benchmarking methods, as summarized in Table 1 and Table 2. These benchmarking methods encompass conventional and DL-based image saliency detection methods, which we categorize into two groups for clarity.

- ITTI [15], GBVS [12], and Judd Model [19]. They are conventional image saliency detection methods that do not rely on neural networks.

- Shallow Convnet [29], GazeGAN [3], EML-NET [16], Deep Convnet [29], SAM-ResNet [7], SalFBNet [10], UNISAL [11], and DeepGaze IIE [27]. This category encompasses diverse DL-based image saliency detection methods, with transfer learning from outside datasets.

4.2.2   Performance Evaluation

We compare the performance of GreenSaliency with eleven benchmarking methods in Table 1 and ten benchmarking methods in Table 2. GreenBaliency outperforms all conventional image saliency detection methods (i.e., ITTI,

Table 1: Performance comparison in five metrics between our GreenSaliency method and eleven benchmarking methods on the MIT300 dataset.

Methods	AUC-J↑	s-AUC↑	CC↑	SIM↑	NSS↑
ITTI [15]	0.543	0.535	0.131	0.338	0.408
GBVS [12]	0.806	0.630	0.479	0.484	1.246
Judd Model [19]	0.810	0.600	0.470	0.420	1.18
Shallow Convnet [29]	0.800	0.640	0.530	0.460	1.430
GazeGAN [3]	0.860	0.731	0.758	0.649	2.211
EML-NET [16]	0.876	0.746	0.789	0.675	2.487
Deep Convnet [29]	0.830	0.690	0.580	0.520	1.510
SAM-ResNet [7]	0.852	0.739	0.689	0.612	2.062
SalFBNet [10]	0.876	0.785	0.814	0.693	2.470
UNISAL [11]	0.877	0.784	0.785	0.674	2.369
DeepGaze IIE [27]	**0.882**	**0.794**	**0.824**	**0.699**	**2.526**
GreenSaliency	0.843	0.700	0.752	0.647	1.713

Table 2: Performance comparison in five metrics between our GreenSaliency method and ten benchmarking methods on the SALICON dataset.

Methods	AUC-J↑	s-AUC↑	CC↑	SIM↑	NSS↑
ITTI [15]	0.667	0.610	0.205	0.378	-
GBVS [12]	0.790	0.630	0.421	0.446	-
Shallow Convnet [29]	0.836	0.670	0.596	0.520	1.458
GazeGAN [3]	0.864	0.736	0.879	0.773	1.899
EML-NET [16]	0.866	0.746	0.868	0.774	**2.058**
Deep Convnet [29]	0.858	0.724	0.622	0.609	1.859
SAM-ResNet [7]	0.865	0.741	**0.899**	**0.793**	1.990
SalFBNet [10]	0.868	0.740	0.892	0.772	1.952
UNISAL [11]	0.864	0.739	0.879	0.775	1.952
DeepGaze IIE [27]	**0.869**	**0.767**	0.872	0.733	1.996
GreenSaliency	0.839	0.679	0.765	0.683	1.605

GBVS, and Judd Model) and some earlier DL-based methods (i.e., Shallow Convnet and Deep Convnet) by a substantial margin in both two datasets. This shows the effectiveness of GreenSaliency in extracting multi-layer hybrid features to cover information from disparate receptive fields. GreenSaliency is also competitive with some DL-based methods (i.e., GazeGAN and SAM-ResNet). Compared with the state-of-the-art DL-based methods (i.e., EML-NET, SalFBNet, UNISAL, and DeepGAZE IIE), there is a gap to reach their performance. However, our model complexity is much lower than those, which is illustrated in Section 4.3.

Additionally, we conducted a qualitative analysis by comparing the predicted saliency maps generated by GreenSaliency with those produced by four benchmark methods. Figure 7 showcases exemplary images that illustrate the instances where GreenSaliency outperforms other methods. Typically, the ground truth saliency maps for images containing multiple objects without a clear dominant focal point exhibit a dispersed and smooth distribution. GreenSaliency excels in these scenarios by effectively attending to all objects within an image, demonstrating its capability to accurately predict saliency without the need for transfer learning from diverse external datasets such as ImageNet. GreenSaliency's methodological independence from transfer learning allows it to maintain uniform attention across an entire scene, leading to high performance in complex images with multiple points of interest. However, in scenarios where images feature prominently captivating objects, such as humans and animals, highlighted in Figure 8, benchmark methods typically outperform GreenSaliency. These methods, having been fine-tuned on extensive and diverse datasets, are better equipped to recognize and prioritize these highly salient features, especially human faces. In such contexts, our GreenSaliency, which does not utilize specialized transfer learning for distinct object categories, might not perform optimally, revealing a potential area for further enhancement in future iterations of the model.
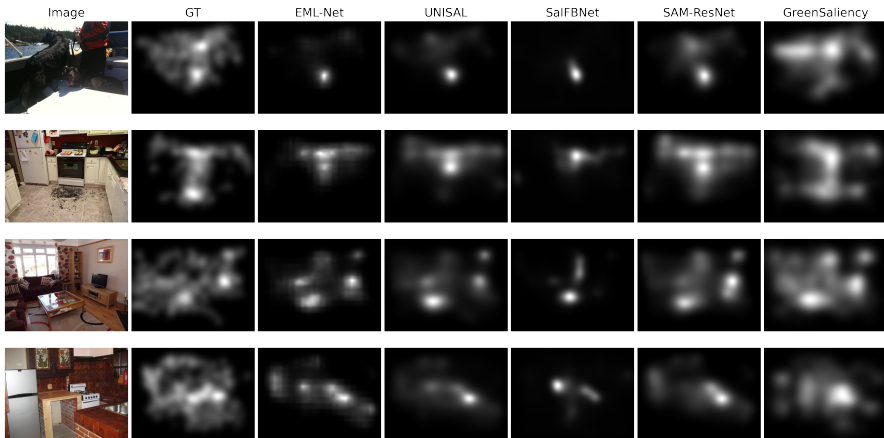

Figure 7: Successful cases in GreenSaliency.

4.3   Model Complexity

The significance of a lightweight model in saliency detection cannot be overstated, especially when it functions as a preliminary processing component in
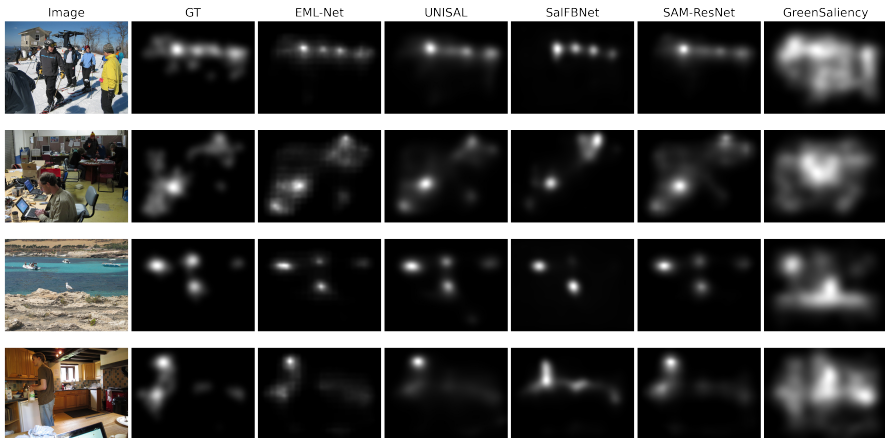
Figure 8: Failed cases in GreenSaliency.

various computer vision applications. Moreover, the complexity of the model plays a pivotal role in determining its suitability for deployment on mobile and edge devices. Our analysis assesses the model complexity of saliency detection methods across three key dimensions: model sizes, inference time, and computational complexity measured in terms of floating-point operations (FLOPs). These metrics are presented comprehensively in Table 3, offering insights into different saliency detection approaches' efficiency and practical feasibility. It is important to clarify that our analysis primarily focuses on comparing model complexity with DL-based methods, due to two main reasons. First, although conventional methods are efficient, the prediction accuracy is substantially lower compared to early-stage DL-based methods and our proposed GreenSaliency method. Second, since conventional methods typically employ unsupervised filters and are often implemented in Matlab rather than Python, directly comparing its complexity with other benchmarks is challenging and potentially unfair.

4.3.1 Model Sizes

The size of a learning model can be assessed through two primary metrics: 1) the total number of model parameters and 2) the actual memory usage. Model parameters can be represented in either floating-point or integer format, typically occupying 4 bytes and 2 bytes of memory, respectively. Given that most model parameters are in floating point, the actual memory usage can be estimated as approximately four times the number of model parameters (as depicted in Table 3). For clarity, we utilize the term "model size" to denote

Table 3: Comparison of no. of model parameters, model sizes (memory usage), no. of GigaFlops, and latency time of several saliency detection methods tested on the SALICON dataset, where "X" denotes the multiplier compared to our proposed method.

Model	#Params (M)↓	Model Size (MB)↓	GFLOPs↓	Runtime (s)↓
Deep Convnet [29]	25.5 (37.5X)	99 (34.7X)	3.2 (20.0X)	0.412 (10.8X)
GazeGAN [3]	208.9 (307.2X)	879.2 (308.5X)	25 (156.2X)	2.540 (66.8X)
EML-NET [16]	43 (63.2X)	180.2 (63.2X)	9.8 (61.3X)	0.365 (9.6X)
Shallow Convnet [29]	620 (911.7X)	2500 (877.2X)	3.9 (24.4X)	0.672 (17.7X)
SAM-ResNet [7]	128.8 (189.4X)	535 (187.7X)	18.8 (117.5X)	0.858 (22.6X)
SalFBNet [10]	5.9 (8.7X)	23.4 (8.2X)	2.30 (14.4X)	0.180 (4.7X)
UNISAL [11]	3.8 (5.6X)	14.7 (5.2X)	1.98 (12.4X)	0.083 (2.2X)
DeepGaze IIE [27]	98.2 (144.4X)	401 (140.7X)	21.2 (132.5X)	6.436 (169.4X)
GreenSaliency	**0.68 (1X)**	**2.85 (1X)**	**0.16 (1X)**	**0.038 (1X)**

memory usage throughout the subsequent discussion. The model sizes of GreenSaliency and eight DL-based benchmark methods are detailed in the second column of Table 3. Notably, GreenSaliency exhibits a significantly lower model than the two DL-based methods featuring lightweight models (i.e., SalFBNet and UNISAL). In contrast to other DL-based methods characterized by notably large model sizes (often exceeding 100MB), GreenSaliency's model size ranges from 34 times to 877 times smaller than these counterparts.

4.3.2 Inference Time

An essential metric for assessing computational efficiency in image saliency detection is the inference time required to generate a saliency map. Our comparative analysis evaluated the inference time of various DL-based methods on a server equipped with an Intel(R) Xeon(R) E5-2620 CPU. The inference time for predicting a single saliency map is documented in the fourth column of Table 3. Notably, GreenSaliency demonstrates a significantly reduced inference time compared to other DL-based methods. Specifically, GreenSaliency achieves an inference time of 0.038 seconds per saliency map prediction, translating to an approximate processing speed of 26 frames per second, utilizing solely CPU resources. It is imperative to acknowledge that as a non-DL-based method, GreenSaliency may not leverage computing acceleration resources as extensively as DL-based counterparts. Nevertheless, with foreseeable advancements in third-party libraries and coding optimizations, GreenSaliency can realize even more excellent efficiency benefits in CPU or GPU-supported environments.

4.3.3 Computational Complexity

The assessment of computational complexity in saliency detection methods can be further elucidated by considering the number of floating-point operations

(FLOPs) required. To this end, we estimated the FLOPs for several DL-based methods essential for predicting a saliency map, compared with those of Green-Saliency. The "GFLOPs" column in Table 3 presents the number of GFLOPs necessary to execute a model once to generate a saliency map. In line with our inference time analysis, GreenSaliency exhibits notably lower computational complexity than other DL-based methods. Expressly, GreenSaliency necessitates 0.16 GFLOPs to predict a single image, whereas other DL-based methodologies require over 2 GFLOPs, resulting in a reduction ranging from 12 to 156 times lower in computational complexity.

4.4   Ablation Study

To evaluate the individual contributions of various components to the overall performance of GreenSaliency, we conducted an ablation study as outlined in Table 4. This study assessed the impact of different feature sets, namely d64, d32, d16, and d8, along with saliency residual prediction (RP) as depicted in Figure 4. Specifically, we independently investigated the efficacy of each set of hybrid features for layers d64, d32, d16, and d8. Our findings revealed that employing a single set of features resulted in the highest performance for the d16 layer, whereas the d64 layer exhibited the lowest performance. Upon incorporating residual prediction, notable enhancements in performance were observed for the d64 and d32 layers, underscoring the importance of residual prediction in improving overall performance. Moreover, through ensembling the performance metrics across all four layers, we observed further improvements across all five evaluation metrics, culminating in achieving the highest performance values. Additionally, we present the performance of single-path predictions, which involve the concatenation of features from all four layers to generate a single saliency map, as described in Section 3.2. It is observed that while the single-path prediction surpasses the performance metrics of individual layers, it does not match the efficacy of the multi-path approach. This finding substantiates the previous discussion regarding the advantages of employing multi-path for saliency prediction, emphasizing the enhanced flexibility and accuracy offered by this method.

5   Conclusion and Future Work

This paper introduces a novel lightweight image saliency detection approach named GreenSaliency, which operates without using DNNs or pre-training on external datasets. GreenSaliency surpasses all conventional (non-DL-based) image saliency detection methods and achieves comparable performance with some early-stage DL-based methods. Compared to state-of-the-art DL-based

Table 4: Ablation study for GreenSaliency on SALICON dataset.

Layer	AUC-J	s-AUC	CC	SIM	NSS
d64	0.822	0.635	0.710	0.616	1.428
d64 + RP	0.827	0.651	0.734	0.655	1.505
d32	0.829	0.639	0.721	0.611	1.425
d32 + RP	0.834	0.658	0.748	0.661	1.516
d16	0.831	0.660	0.726	0.628	1.425
d8	0.830	0.662	0.725	0.635	1.428
Singular-path prediction	0.836	0.670	0.754	0.670	1.541
Multi-path ensmeble	0.839	0.679	0.765	0.683	1.605

methods, GreenSaliency exhibits lower prediction accuracy but offers advantages in smaller model sizes, shorter inference time, and reduced computational complexity. The minimal model complexity of GreenSaliency suggests a more efficient energy usage, making it suitable for integration into extensive image processing systems.

Our proposed GreenSaliency method identifies several limitations, which could be addressed in future iterations. Notably, GreenSaliency struggles to accurately predict particular popular objects of interest, such as human faces and animals, while tending to emphasize all objects within an image, as illustrated in Figure 8. Consequently, implementing a mechanism to rank the saliency levels of different objects could enhance prediction accuracy by focusing on the most appealing objects. Additionally, given the significance of saliency detection in various computer vision tasks, such as blind image quality assessment (BIQA), there exists potential and interest in developing a saliency-guided approach for addressing BIQA challenges.

References

[1] A. Borji and L. Itti, "Cat2000: A large scale fixation dataset for boosting saliency research", *arXiv preprint arXiv:1505.03581*, 2015.

[2] J. Canny, "A computational approach to edge detection", *IEEE Transactions on pattern analysis and machine intelligence*, (6), 1986, 679–98.

[3] Z. Che, A. Borji, G. Zhai, X. Min, G. Guo, and P. Le Callet, "How is gaze influenced by image transformations? dataset and model", *IEEE Transactions on Image Processing*, 29, 2019, 2287–300.

[4]   H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "DefakeHop: A Light-Weight High-Performance Deepfake Detector", in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2021, 1–6.

[5]   T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, 785–94.

[6]   Y. Chen, M. Rouhsedaghat, S. You, R. Rao, and C.-C. J. Kuo, "Pixel-hop++: A small successive-subspace-learning-based (SSL-based) model for image classification", in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, 3294–8.

[7]   M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an lstm-based saliency attentive model", *IEEE Transactions on Image Processing*, 27(10), 2018, 5142–54.

[8]   J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, 248–55.

[9]   G. Ding, N. İmamoğlu, A. Caglayan, M. Murakawa, and R. Nakamura, "Fbnet: Feedback-recursive cnn for saliency detection", in *2021 17th International Conference on Machine Vision and Applications (MVA)*, IEEE, 2021, 1–5.

[10] G. Ding, N. İmamoğlu, A. Caglayan, M. Murakawa, and R. Nakamura, "SalFBNet: Learning pseudo-saliency distribution via feedback convolutional networks", *Image and Vision Computing*, 120, 2022, 104395.

[11] R. Droste, J. Jiao, and J. A. Noble, "Unified image and video saliency modeling", in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, Springer, 2020, 419–35.

[12] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency", *Advances in neural information processing systems*, 19, 2006.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 770–8.

[14] A. A. Ismail, H. Corrada Bravo, and S. Feizi, "Improving deep learning interpretability by saliency guided training", *Advances in Neural Information Processing Systems*, 34, 2021, 26726–39.

[15] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis", *IEEE Transactions on pattern analysis and machine intelligence*, 20(11), 1998, 1254–9.

[16] S. Jia and N. D. Bruce, "Eml-net: An expandable multi-layer network for saliency prediction", *Image and vision computing*, 95, 2020, 103887.

[17] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, 1072–80.

[18] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations", 2012.

[19] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look", in *2009 IEEE 12th international conference on computer vision*, IEEE, 2009, 2106–13.

[20] W. Kienzle, F. A. Wichmann, M. Franz, and B. Schölkopf, "A non-parametric approach to bottom-up visual saliency", *Advances in neural information processing systems*, 19, 2006.

[21] M. Kummerer, T. S. Wallis, and M. Bethge, "Saliency benchmarking made easy: Separating models, maps and metrics", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, 770–87.

[22] M. Kummerer, T. S. Wallis, L. A. Gatys, and M. Bethge, "Understanding low-and high-level contributions to fixation prediction", in *Proceedings of the IEEE international conference on computer vision*, 2017, 4789–98.

[23] M. Kümmerer, L. Theis, and M. Bethge, "Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet", *arXiv preprint arXiv:1411.1045*, 2014.

[24] C.-C. J. Kuo and A. M. Madni, "Green learning: Introduction, examples and outlook", *Journal of Visual Communication and Image Representation*, 90, 2023, 103685.

[25] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, "Interpretable convolutional neural networks via feedforward design", *Journal of Visual Communication and Image Representation*, 60, 2019, 346–59.

[26] X. Lei, G. Zhao, K. Zhang, and C.-C. J. Kuo, "TGHop: an explainable, efficient, and lightweight method for texture generation", *APSIPA Transactions on Signal and Information Processing*, 10, 2021, e17.

[27] A. Linardos, M. Kümmerer, O. Press, and M. Bethge, "DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling", in *Proceedings of the IEEE- /CVF International Conference on Computer Vision*, 2021, 12919–28.

[28] Z. Mei, Y.-C. Wang, X. He, and C.-C. J. Kuo, "GreenBIQA: A Lightweight Blind Image Quality Assessment Method", in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2022, 1–6.

[29] J. Pan, E. Sayrol, X. Giro-i-Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and deep convolutional networks for saliency prediction", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 598–606.

[30] J. Pan, E. Sayrol, X. G.-i. Nieto, C. C. Ferrer, J. Torres, K. McGuinness, and N. E. OConnor, "Salgan: Visual saliency prediction with adversarial networks", in *CVPR scene understanding workshop (SUNw)*, 2017.

[31] Y. Patel, S. Appalaraju, and R. Manmatha, "Saliency driven perceptual image compression", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, 227–36.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, 4510–20.

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

[34] A. Uddin, M. Monira, W. Shin, T. Chung, S.-H. Bae, *et al.*, "Saliencymix: A saliency guided data augmentation strategy for better regularization", *arXiv preprint arXiv:2006.01791*, 2020.

[35] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, 2798–805.

[36] S. Yang, Q. Jiang, W. Lin, and Y. Wang, "SGDNet: An end-to-end saliency-guided deep neural network for no-reference image quality assessment", in *Proceedings of the 27th ACM international conference on multimedia*, 2019, 1383–91.

[37] Y. Yang, W. Wang, H. Fu, C.-C. J. Kuo, *et al.*, "On supervised feature selection from high dimensional feature spaces", *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[38] D. Zhang, J. Han, Y. Zhang, and D. Xu, "Synthesizing supervision for learning deep saliency network without human annotation", *IEEE transactions on pattern analysis and machine intelligence*, 42(7), 2019, 1755–69.

[39] 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, 32–2.