Itty-Koch-Style Saliency Maps

Mohammad Mohammad Beigi[a]

[a]Student,EE Department, Sharif University of Technology

This manuscript was compiled on October 27, 2023

Keywords: Hmax | sumfilter | maxfilter | SVM | d'

This paper presents a novel visual attention system that draws inspiration from the early visual system of primates in terms of behavior and neural architecture. The system combines different scales of image features to create a saliency map, which highlights the most important regions in the visual scene. A dynamic neural network is then used to sequentially select attended locations based on their saliency, starting with the most salient ones.[1] This approach enables the system to efficiently tackle the challenging task of scene understanding by identifying and focusing on conspicuous areas for further analysis and detailed processing.[2]

Primates possess an impressive ability to process detailed visual scenes in real-time, despite the limitations of their neural hardware. To deal with the complexity of visual analysis, intermediate and higher visual processes selectively concentrate on a subset of sensory information. This selection occurs through a limited region in the visual field known as the "focus of attention." The focus of attention operates in two modes: a fast and saliency-driven mode, which is independent of the task, and a slower and task-dependent mode controlled by volition.[1][2][3]

Different attention models exist,[5][6] including "dynamic routing" models, which allow only a small region of the visual field to progress through the cortical visual hierarchy. This region is selected by dynamically modifying cortical connectivity or establishing specific temporal activity patterns. These mechanisms provide both top-down (task-dependent) and bottom-up (scene-dependent) control over the selection of visual information for further processing.

The presented model, based on Koch and Ullman's biologically plausible architecture, shares similarities with the "feature integration theory" that explains how humans perform visual searches.[7] The model initially separates the visual input into feature maps, and within each map, different spatial locations compete for saliency. Only locations that stand out within their surroundings are retained. All feature maps contribute, in a purely bottom-up manner, to a central "saliency map" that encodes local prominence across the entire visual scene. In primates, this saliency map is believed to exist in the posterior parietal cortex and the visual maps within the pulvinar nuclei of the thalamus.

The model's saliency map incorporates internal dynamics that generate attention shifts, eliminating the need for top-down guidance. This framework allows for the rapid selection of a few notable image locations to undergo more complex and time-consuming object recognition processes. Extending this approach, a "guided-search" mechanism incorporates feedback from higher cortical areas, utilizing knowledge about target objects to assign weights to different features. Consequently, only features with higher weights proceed to higher processing levels.

Saliency Map

A saliency map is a representation of an image that highlights the most visually prominent or important regions within the scene. It is a computational technique used in computer vision and visual attention systems to identify areas of an image that are likely to attract human attention or contain significant information.

The saliency map is typically generated by analyzing various low-level visual features of the image, such as color contrast, intensity, orientation, and texture. These features are combined and processed to assign a saliency value to each pixel or region in the image. Higher saliency values indicate regions that are more visually distinctive or attention-grabbing.

By creating a saliency map, it becomes possible to prioritize the processing of relevant image regions for tasks such as object recognition, scene understanding, or visual search. It helps to simulate the human visual system's selective attention mechanism, where certain regions stand out and are more likely to be focused on and processed in greater detail.

Model

The process involves the model choosing specific areas to focus on using a saliency map that represents the level of attention or importance at each location in the visual field. This map assigns a numerical value to indicate the saliency at each point. A dynamic neural network then prioritizes the attended locations based on their decreasing saliency values. By doing so, the system efficiently addresses the challenging task of understanding a scene by quickly identifying and analyzing the most prominent locations in the visual field.

Note that the model does not require any top-down guidance to shift attention. The model's saliency map is endowed with internal dynamics which generate attentional shifts.

Results

Part A

A saliency map is a representation or visualization of the regions or areas within an image or visual stimulus that are deemed visually salient or attention-grabbing. It highlights the areas that are most likely to attract the viewer's attention based on various visual cues and features.

The saliency map is typically generated using computational algorithms or models that analyze different aspects of the visual input, such as color, contrast, orientation, and motion. These algorithms assign saliency values to different re-

Please provide details of author contributions here.

[1]A.O.(Author One) and A.T. (Author Two) contributed equally to this work (remove if not applicable).

gions or pixels in the image, indicating their relative importance or likelihood of drawing attention.

The resulting saliency map can be displayed as an overlay on top of the original image, where the regions with higher saliency values are typically represented with brighter colors or higher visual contrast. By visualizing the saliency map, researchers and designers can gain insights into which parts of an image or scene are more likely to capture the viewer's attention.

Saliency maps have various applications, including computer vision, visual attention modeling, image and video compression, object recognition, and human-computer interaction. They provide a valuable tool for understanding and predicting human visual attention and can assist in optimizing designs and user interfaces to effectively communicate and convey information.

We obtain the focus of attention(heat map), overall Itty salience, conspiquity map of orientation, conspiquity map of colors, conspiquity map of intensity and single color channels by running the code.

Fig. 1. overall Itty salience

Fig. 2. focus of attention

Part B

Now we plot eye tracker data from data set on the saliency map and spot fixation points for three different subjects.

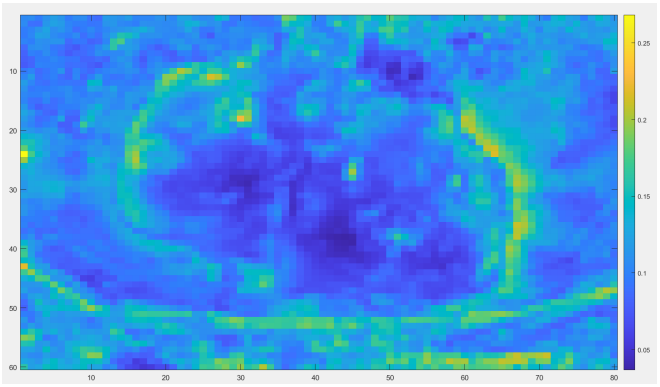

Fig. 3. overall Itty salience

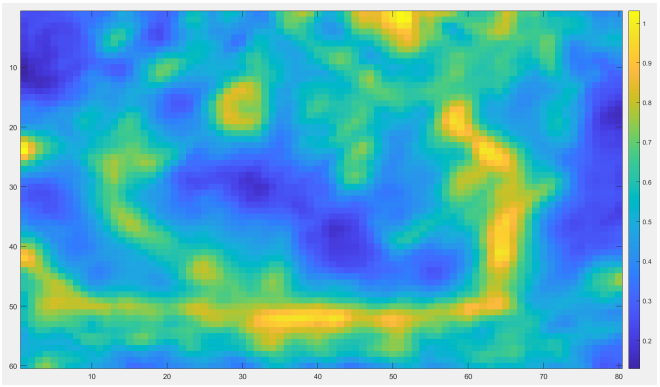

Fig. 4. conspiquity map of orientation

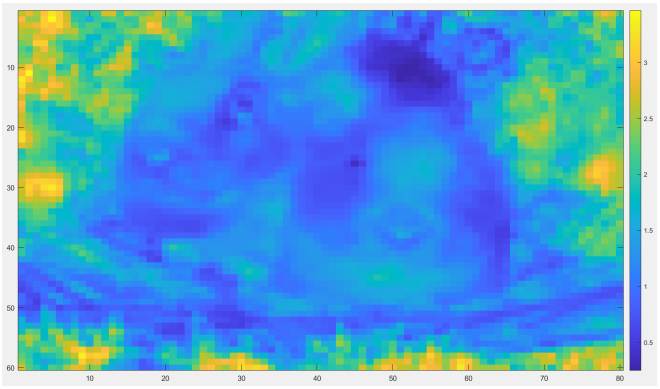

Fig. 5. conspiquity map of colors

When plotting eye tracker data on a saliency map and spotting fixation points, you are combining information about where a person is looking (eye tracker data) with the areas of visual attention highlighted by the saliency map.

Eye tracker data consists of the precise gaze coordinates recorded by an eye-tracking device as a person observes an image or scene. These coordinates indicate the specific locations on the visual stimulus where the person's eyes fixate or focus.

A saliency map, as mentioned earlier, represents the visually salient or attention-grabbing areas within the image. It indicates regions that are likely to capture the viewer's attention based on visual cues and features.

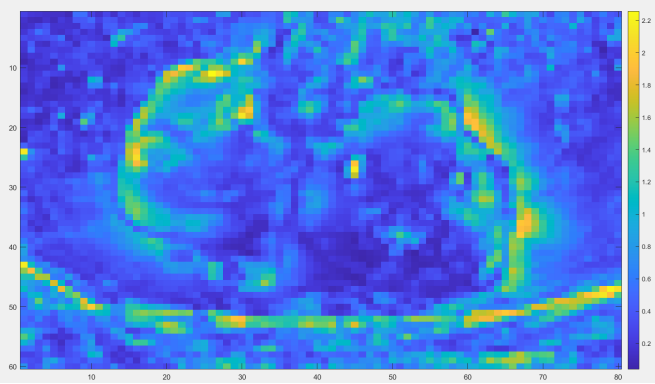

Fig. 6. conspiquity map of intensity

Fig. 7. focus of attention

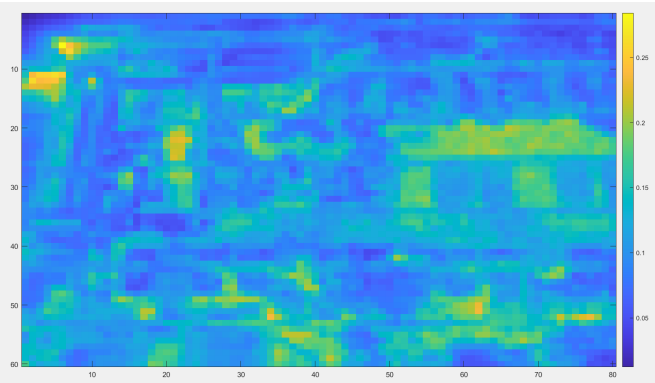

Fig. 8. overall Itty salience

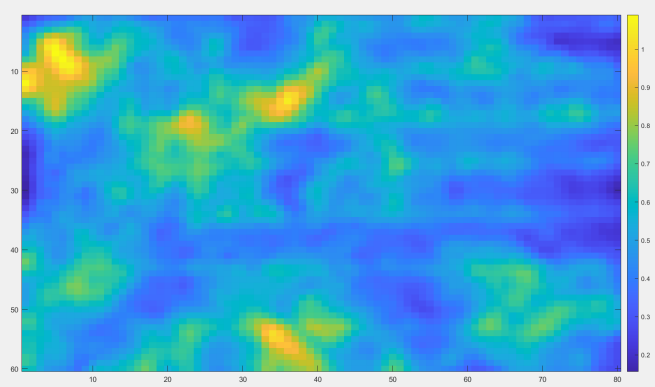

Fig. 9. conspiquity map of orientation

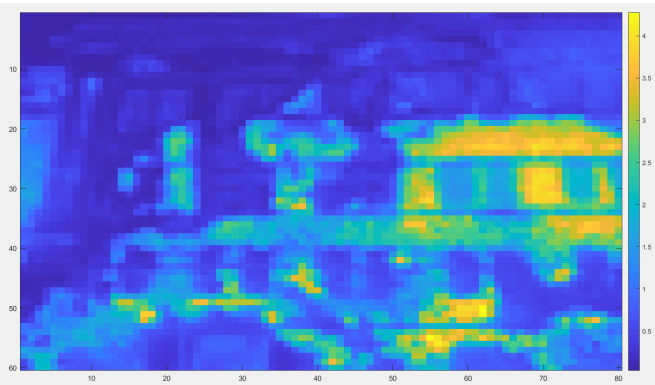

Fig. 10. conspiquity map of colors

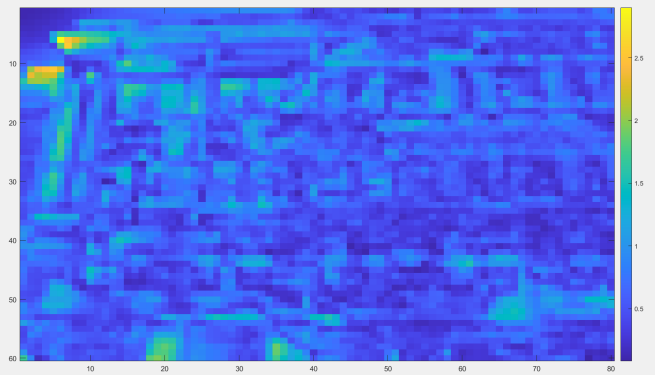

Fig. 11. conspiquity map of intensity

By overlaying the eye tracker data on the saliency map, you can visually identify and analyze fixation points. Fixation points are the specific locations where the person's gaze lingers for a certain duration before moving to another area. These fixation points often correspond to areas of high saliency on the map, indicating that the viewer's attention is drawn to visually significant regions.

By examining the relationship between the eye tracker data, fixation points, and the saliency map, you can gain insights into how people perceive and attend to visual stimuli. This analysis can help in understanding which areas of an image or scene attract the most attention and how visual attention is influenced by saliency cues.

1. Part C

Now we use Gazerecorde tool and obtain your personal heatmap on stimulus images(figure 18 and 19). A gaze tracker tool, also known as an eye tracker or eye-tracking system, is a device or software application that captures and measures eye movements and gaze behavior. It allows for the monitoring and analysis of where a person is looking, how long they fixate on specific areas or objects, and the patterns of their eye movements.

Eye tracking technology typically involves the use of infrared sensors or cameras to track the position and movements of the eyes. These sensors can detect the reflection or absorption of infrared light by the cornea or pupil of the eye. By

Fig. 12. Eye tracker data for first subject

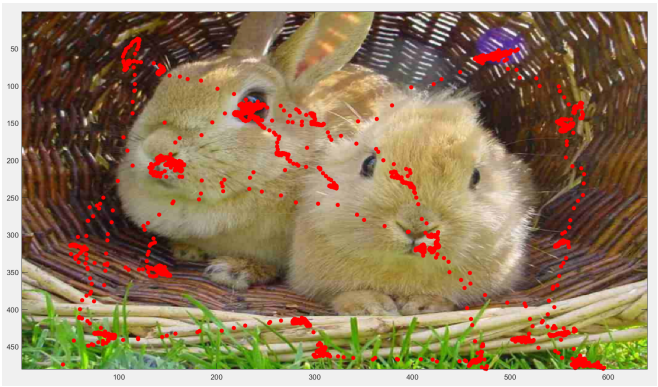

Fig. 13. Eye tracker data for second subject

Fig. 14. Eye tracker data for third subject

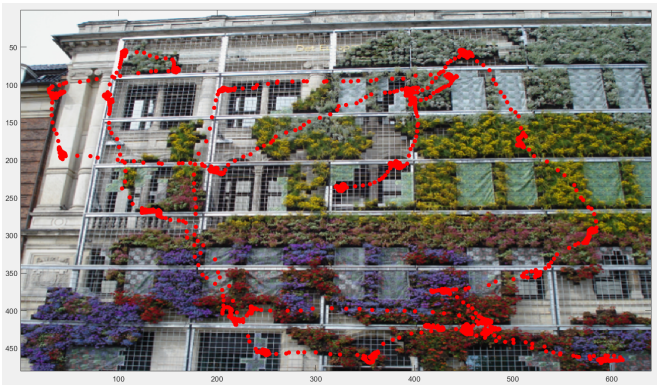

Fig. 15. Eye tracker data for first subject

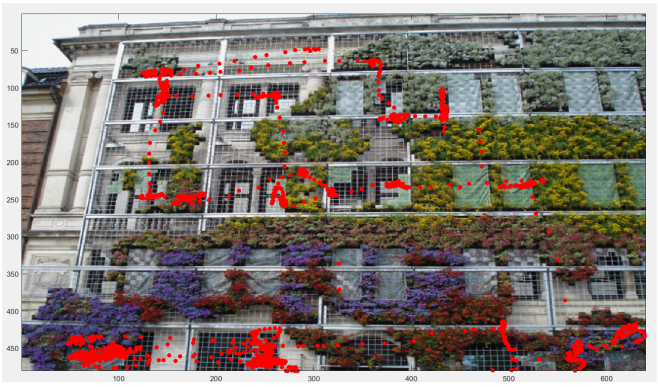

Fig. 16. Eye tracker data for second subject

Fig. 17. Eye tracker data for third subject

tracking the position of the eyes over time, the tool can determine the direction of gaze and provide insights into visual attention and cognitive processes.

Gaze tracker tools find applications in various fields, including psychology, market research, human-computer interaction, usability testing, and medical research. They can help researchers and designers understand how individuals perceive and interact with visual stimuli, user interfaces, advertisements, web pages, and virtual environments. Eye tracking data can provide valuable insights into user behavior, preferences, and decision-making processes.

With the advancements in technology, gaze tracker tools have become more accessible and portable. There are now both hardware-based eye tracking devices, such as standalone eye trackers or integrated eye-tracking systems in virtual reality headsets, as well as software-based solutions that utilize webcams or built-in cameras in laptops and mobile devices. These tools offer opportunities for a wide range of applications and have the potential to enhance various fields of research and design.

2. Part D

NSS (Normalized Scanpath Saliency) is a metric used to evaluate the performance of saliency models in eye-tracking studies. Saliency models aim to predict the regions in an image that are most visually salient or attract human attention. NSS

Fig. 18. GazeRecorder result

Fig. 19. GazeRecorder result

measures how well a saliency model's predictions align with human eye fixation data.

To calculate NSS, the following steps are typically followed:

1. Collect eye-tracking data: Eye-tracking experiments are conducted, where participants' eye movements are recorded while they view a set of images. These eye movements are represented as fixations, which are points where the eyes remain relatively stationary.

2. Generate saliency maps: Saliency models are applied to the same set of images to generate saliency maps, which highlight the regions predicted to be visually salient.

3. Compute normalized scanpath saliency: NSS measures the correspondence between fixations and saliency predictions. It is calculated by averaging the saliency values at the fixation locations and subtracting the mean saliency value across the entire image. This average is then divided by the standard deviation of saliency values across the image. The formula for NSS is as follows:

NSS = (mean(saliency at fixations) - mean(saliency))/std(saliency)

A higher NSS score indicates better alignment between the saliency predictions and the eye fixations.

Receiver Operating Characteristic (ROC) is a graphical representation of the performance of a binary classification model. It is commonly used in machine learning and signal detection tasks to evaluate the trade-off between true positive rate (sensitivity) and false positive rate (1 - specificity) at various decision thresholds.

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) for different classification thresholds. The TPR is the proportion of actual positive samples correctly classified as positive, and the FPR is the proportion of actual negative samples incorrectly classified as positive.

The ROC curve provides a visual representation of how well a binary classifier can distinguish between classes, with the ideal classifier having an ROC curve that reaches the top-left corner of the graph (TPR = 1, FPR = 0). The area under the ROC curve (AUC) is often used as a summary statistic to quantify the overall performance of the classifier. A higher AUC indicates better classification performance, with a maximum value of 1 representing a perfect classifier.

I wrote a code to calculate NSS(normalized scanpath saliency) and ROC(Receiver Operating Characteristic) but it did not work. The code is as below:

fixations = zscore(fixations); saliency_map = im2double(saliency_map); saliency_values = saliency_map(sub2ind(size(saliency_map), fixations(:, 2), fixations(:, 1))); nss_score = mean(saliency_values);

thresholds = linspace(0, 1, 100); true_positives = zeros(size(thresholds)); false_positives = zeros(size(thresholds));

for i = 1:numel(thresholds) threshold = thresholds(i); binarized_map = saliency_map >= threshold; true_positives(i) = sum(binarized_map(fixations(:, 2), fixations(:, 1))); false_positives(i) = sum(binarized_map(:)) - true_positives(i); end sensitivity = true_positives / size(fixations, 1); specificity = 1 - (false_positives / (size(saliency_map, 1) * size(saliency_map, 2))); auc_roc = trapz(false_positives, sensitivity);

disp(['NSS Score: ', num2str(nss_score)]); disp(['AUC-ROC: ', num2str(auc_roc)]);

figure; plot(false_positives, sensitivity); xlabel('False Positive Rate'); ylabel('True Positive Rate'); title('ROC Curve'); grid on;

References

[1] J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y.H. Lai, N. Davis, and F. Nuflo, Modelling Visual Attention via Selective Tuning, Artificial Intelligence, vol. 78, no. 1-2, pp. 507545, Oct. 1995.

[2] E. Niebur and C. Koch, Computational Architectures for Attention, R. Parasuraman, ed., The Attentive Brain, pp. 163186. Cambridge, Mass.: MIT Press, 1998.

[3] B.A. Olshausen, C.H. Anderson, and D.C. Van Essen, A Neurobiological Model of Visual Attention and Invariant Pattern Recognition Based on Dynamic Routing of Information, J. Neuroscience, vol. 13, no. 11, pp. 4,7004,719, Nov. 1993.

[4] C. Koch and S. Ullman, Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry, Human Neurobiology, vol. 4, pp. 219227, 1985.

[5] R. Milanese, S. Gil, and T. Pun, Attentive Mechanisms for Dynamic and Static Scene Analysis, Optical Eng., vol. 34, no. 8, pp. 2,4282,434, Aug. 1995.

[6] S. Baluja and D.A. Pomerleau, Expectation-Based Selective Attention for Visual Monitoring and Control of a Robot Vehicle, Robotics and Autonomous Systems, vol. 22, no. 3-4, pp. 329344, Dec. 1997.

[7] A.M. Treisman and G. Gelade, A Feature-Integration Theory of Attention, Cognitive Psychology, vol. 12, no. 1, pp. 97136, Jan. 1980