UNIVERSITY OF MONS

INSTITUTE OF TECHNOLOGY OF CAMBODIA

Visual Attention: Top-down and Bottom-up Information Relative Importance

Phutphalla KONG

phutphalla.kong@student.umons.ac.be / phutphalla.kong@gsc.itc.edu.kh

A dissertation submitted to the Faculty of Engineering
of the University of Mons, for the degree of Doctor of Philosophy in Engineering Science

Supervisors:
Prof. Bernard GOSSELIN
Prof. Kimtho PO

This thesis was supported by ARES-CCD (program AI 2014-2019)
under the funding of Belgian university cooperation

Wednesday 8th September, 2021

Jury members

Prof. **Pierre MANNEBACK** - University of Mons, President and Reviewer
Prof. **Thierry DUTOIT** - University of Mons, Committee member
Prof. **Bernard GOSSELIN** - University of Mons, Supervisor in Belgium
Prof. **Christophe De VLEESCHOUWER** - Catholic University of Louvain, Reviewer
Prof. **Kimtho PO** - Institute of Technology of Cambodia, Supervisor in Cambodia
Prof. **Sokchenda SRENG** - Institute of Technology of Cambodia, Reviewer
Dr. **Matei MANCAS** - University of Mons, Co-supervisor in Belgium

"**PhD** is not a **MUST**, but it a **PLUS**" – by Phutphalla **KONG**

Acknowledgements

First, I would like to thanks His Excellency Dr. **OM Romny**, director of the Institute of Technology of Cambodia (ITC) for sustainable management of the institute and good cooperation with partner universities at local, regional, and international levels to enhance the quality of the training of doctors, masters, engineers, and technicians program.

Second, I would to deeply thanks Prof. **Bernard**, Prof **Kimtho**, and Dr. **Matei** my supervisors and co-supervisor to give plenty of useful advice and help me during my research. They have used their efforts and experiences to guide me in the correct direction of my research work.

Third, I would like to express my sincere regard and gratitude to Prof. **Thierry** and Dr. **Nathalie** who always help and care me during my stay in Belgium, not only activities in the laboratory but also administration documents. Moreover, I would like to thanks all staffs from the ARES-CCD who help me for official administration documents.

Fourth, I would like to thanks all committee members both in Belgium and in Cambodia that spend their rich time to review my thesis and provide some useful remarkable points in my thesis. They help me to improve my manuscript and its structure.

Fifth, I would like to thank all the colleagues of the lab both in Belgium and Cambodia with which I asked them for help and sharing knowledge together. They also helped me in my eye-tracking experiment.

Last but not least, I would like to warmly thanks my family and my friends for their caring and support, especially my parents who always educate and encourage me in any circumstances.

Abstract

Human visual system is modeled in engineering field providing feature-engineered methods which detect contrasted, surprising, or unusual data into images. This data is "interesting" for humans and leads to numerous applications. Deep learning (DNNs) drastically improved the algorithms efficiency on the main benchmark datasets. However, DNN-based models are counter-intuitive: surprising or unusual data is by definition difficult to learn because of its low occurrence probability. In reality, DNNs models mainly learn top-down features such as faces, text, people, or animals which usually attract human attention, but they have low efficiency in extracting surprising or unusual data in the images.

In this thesis, we propose a model called DeepRare (DR) family model including DeepRare2019 (DR19) and DeepRare2021 (DR21) which uses the power of DNNs feature extraction and the genericity of feature-engineered algorithms. This algorithm is an evolution of a previous version, DR19. DR21 1) does not need any training and uses the default ImageNet training, 2) it is fast even on CPU, 3) our tests on four very different eye-tracking datasets show that DR21 is generic and is always in the within the top models on all datasets and metrics while no other model exhibits such a regularity and genericity. Finally DR21 provides 4) explanation and transparency on why parts of the image are the most surprising at different levels despite the us of a DNN-based feature extractor and 5) it is tested with several network architectures such as VGG16 (V16), VGG19 (V19) and MobileNetV2 (MN2). DeepRare2021 code can be found at VisualAttention-RareFamily.

Keywords: Visual attention prediction, Top-down information, Bottom-up information, Object detection, Face detection, Text detection, Saliency, Rarity, Eye tracking, Deep features, Odd one out, Visibility.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

A model of visual attention is a description of the observed and/or predicted behavior of human and non-human primate visual attention (scholarpedia). A computational model of visual attention is an instance of a model of visual attention, and not only includes a formal description for how attention is computed, but also can be tested by providing image inputs, similar to those an experimenter might present to a subject, and then seeing how the model performs by comparison. In computer vision, a visual attention modeling is an automatic method or technique for predicting a probability of the image pixels attended by an average of human viewers. The basic image features (such as colors, orientations, textures and so on) provide the so-called "bottom-up" information. The other features (such as the image contexts and the viewers memory, experiences, goals, emotions and characteristics), which are more subjective and so called "top-down", can drastically influence people's attention. In other words, these features/elements mainly take into account the elements that might attract a viewer's attention, which include the rareness and surprising features extracted from an image.

According to the research directions taken by the Smart Spaces Group in Information, Signal, Artificial Intelligence (ISIA) Laboratory, a good top-down modeling method for visual attention has been found by using the information related to the emotions, memory, goals and viewing contexts. Due to these facts, a research will therefore focus on two different directions: **(1)** how to improve the performance of a visual attention prediction model using both bottom-up and top-down information and **(2)** how to make generic model for visual attention maps.

This project intends to strength the contacts between the Numediart Institute of the University of Mons (UMONS) and Institute of Technonlogy of Cambodia (ITC) and bring both scientific publications and outcomes which have a lot of practical applications. In Belgium, the Ittention company could benefit directly from the results of the research. In Cambodia, some results can be used on ITC communication optimization to test how potential clients react to the project outcomes. This project also takes advantage of real clients of Ittention feedback on the results to be able to anchor the scientific research in real-life industry needs. In this way, the results will be closer to a rapid use in industry like marketing agencies. This project follows the same structure as the research project: 1) the use of deep learning to detect different kinds of object to improve the bottom-up attention map, 2) acquisition and use of eye-tracking data to personalize and improve the bottom-up attention maps depending

on the type of documents and people observing them, 3) research on the relationship between bottom-up attention, detected objects in the image and people eye gaze on the same image.

1.2 Problematic

There is a growing interest to use human visual attention abilities to prioritize information in computational systems. This is especially the case for computer vision which has to select the most relevant parts in a large amount of data. This is why modeling visual attention, particularly the bottom-up part which is driven by features from the visual scene, has been a very active research area over the past 20 years. Many different models of visual bottom-up attention are now available online. They take as input natural images and output a saliency map which gives the probability for each pixel to catch our attention. As in Fig. 1.1, it describes about result of visual attention by using eye-tracking hardware in order to attract attention. However, nowadays there is few research on top-down information, and especially by adding face and text detection on bottom-up saliency maps. This method is called "top-down approach". Moreover, mixing bottom-up and top-down information both the existing and deep learning models is still lack of research and especially makes it in generic way. For the deep learning models, they cannot explain how to modify or improve results inside the neural networks, just provide the final result. That's why this research topic is proposed in order to get high prediction visual attention in images such as natural, website, and advertisement images in generic and explainable way.

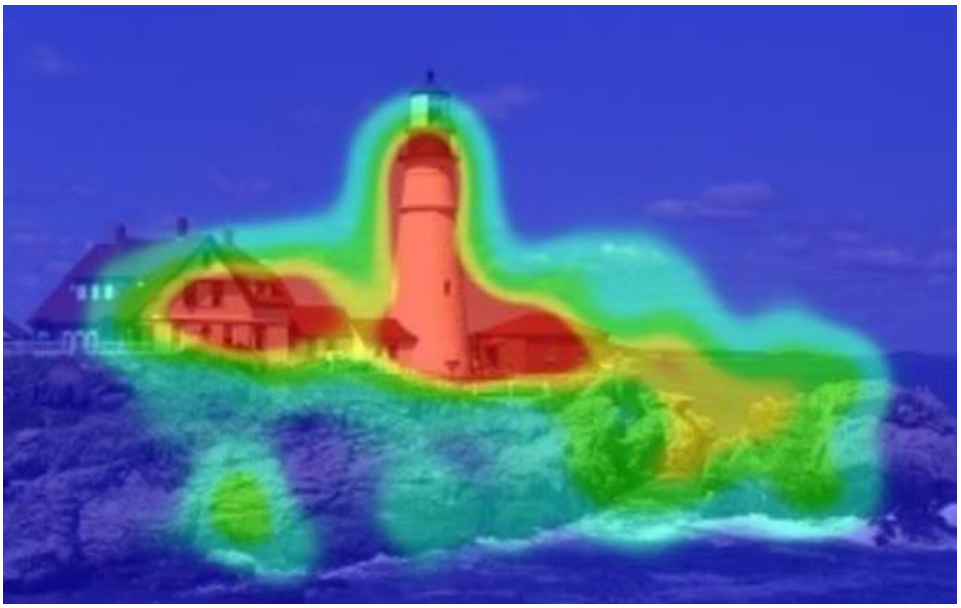


Figure 1.1. An example result of visual attention by using eye-tracking data.

1.3 Objective

What is attention? - **Attention** is the notice taken of somebody or something as interesting or important. In machine learning, attention refers to (1) trainable attention which is a group of techniques that helps a model-in-training effectively notice important things; A trainable attention mechanism is trained while the network is trained and is supposed to help the network to concentrate on key elements of the image and (2) post-hoc attention which is a group of techniques that helps human visualize what an already-trained model thinks is important; A post-hoc mechanism creates a heat map from an already-trained network by including an already-trained model with fixed weights and by providing insight into the model's decisions. What is an attention map? - An **attention map** is a scalar matrix representing the relative importance of layer activations at different two-dimensional spatial locations with respect to the target task. It means that an attention map is a grid of numbers which indicates what two-dimensional locations are important for a task. Important locations correspond to bigger numbers and are normally depicted in red in a heat map, for example, in Fig. 1.1.

There are two main objectives: **1)** the first one focuses on the optimization of existing attention models given a specific category of stimuli and on objects detection and **2)** the second on the increase of the results of attention maps in generic and explainable way by using the advantages of feature extraction and engineering. For the outcome, this project can help other researchers to be able to detect the important points in an image better than not using the techniques in this project and be able to explain why the neural networks provide us with wrong results.

1.4 Solutions

To achieve the objectives, the project is divided into five main tasks:

1. Building the datasets containing different advertisements (i.e., images and videos) and websites (i.e., screenshots and screencasts) and defining the methodology of the dataset's collection and questionnaires. Some preliminary tests will be achieved to check whether the datasets are correct and the data acquired makes sense.
2. Installing and testing the existing models of object detection using deep learning techniques. An optimal method of fusing the results of the object detection and the existing bottom-up attention maps will be found to optimize the current methods.
3. Optimizing the algorithm using the information taken from different eye-tracking datasets. This step will tune attention models to specific documents (such as websites, specific people categories, or others).
4. Working on visual attention of the images based on the bottom-up attention map, the objects detected in the images, and the way people really looks to those images.
5. Working on convolutional neural networks to extract features (top-down information) and combine them with a bottom-up information in order to make a novel generic and explainable model.

1.5 Important Modules

For the first research direction, in order to improve the performance of visual attention model, this research intends to use a combination of the saliency visual features and both face and text features extracted using the existing methods. Then, the obtained experimental results will be evaluated versus the results of eye-tracking model.

For second research direction, the deep learning methods have proven a better performance than the existing methods in many recent articles because they are helpful in detecting the information related to high-level concepts of images (i.e., people, objects, text, logos, etc.). Therefore, for next research improvement, several selected deep learning methods will be tested and then used instead of the existing ones. In addition, the top-down features and object recognition (i.e., text and face) features will be also included in this research in order to increase the result of the attention map in generic way and can explain how the neural networks inside provide with wrong results.

In evaluation, several test sets have been proposed for comparing to the eye-tracking data. Moreover, the datasets from MIT will be also used for evaluating our proposed methods. For this first time, the manual evaluation using real human ability to score the visual attention of the produced images.

In terms of real applications, this research outcomes aim to be used for scoring the visual marketing of the online/offline advertisements or websites.

1.6 Organization of the thesis

This thesis is organized into two main parts. The first part is related to state of the art methods. The second part focuses on a new framework by using both existing and deep learning methods.

Organization of Part I

Part I of this dissertation is dedicated to existing methods for computing visual attention in computer vision. Chapter 1 introduces the general introduction of this research work. Chapter 2 introduces the background of bottom-up and top-down attention models for images which use both existing and deep learning methods.

Organization of Part II

Part II of this thesis addresses the way how to combine bottom-up saliency maps with top-down information. Chapter 3 presents bottom-up attention maps with high-level semantic information by using the existing methods. Chapter 4 focuses on Convolutional Neural Network (CNN) features rarity as an attention cue by using the deep learning methods. Chapter 5 presents the general conclusion by providing the contributions of this thesis and perspective.

1.7 Original Contributions

The main contributions of the present thesis encompass:

- Bottom-up attention maps with high-level semantic information by using the existing methods (Chapter 3).
- CNN features rarity as an attention cue by using the deep learning methods (Chapter 4). A novel model which is called **DeepRare2019 (DR19)** is proposed. This model is generated based on Deep Neural Network, VGG16. Then further experiment is conducted by upgrading the DR19 with considering on different and combined threshold from 0 to 0.9. This model is called **DeepRare2021 (DR21**) which is more tesing Deep Neural Network VGG19 and MobileNetV2. Moreover, first publication paper [34] is accepted on the 1st International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI2018). Second publication paper [33] is accepted on 2018 IEEE International Conference on Image Processing (ICIP2018). Third publication paper [44] is accepted on 2020 IEEE International Conference on Imaging, Vision & Pattern Recognition (IVPR2020). The last model is submitted to an international journal, Signal Processing: Image Communication.

Part I

Background of Visual Attention Modeling

Chapter 2

Background

2.1 Introduction

The **salience** (also called **saliency**) of an item (i.e., an object, a person, a pixel, etc.) is the state or quality by which it stands out from its neighbors. Saliency detection is considered to be a key attentional mechanism that facilitates learning and survival by enabling organisms to focus their limited perceptual and cognitive resources on the most pertinent subset of the available sensory data. Saliency typically arises from contrasts between items and their neighborhood, such as a red dot surrounded by white dots, a flickering message indicator of an answering machine, or a loud noise in an otherwise quiet environment. Saliency detection is often studied in the context of the visual system, but similar mechanisms operate in other sensory systems.

In computer vision, a **saliency map** is an image that shows each pixel's unique quality. The goal of a saliency map is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. For example, if a pixel has a high grey level or other unique color quality in a color image, that pixel's quality will show in the saliency map and in an obvious way (see Fig. 2.1).

When attention deployment is driven by salient stimuli, it is considered to be bottom-up, memory-free, and reactive. Conversely, attention can also be guided by top-down, memory-dependent, or anticipatory mechanisms, such as when looking ahead of moving objects or sideways before crossing streets. Humans and other animals have difficulty paying attention to more than one item simultaneously, so they are faced with the challenge of continuously integrating and prioritizing different bottom-up and top-down influences.

This chapter, we present about state of the art of bottom-up, top-down, and deep learning attention models.

2.2 State-of-the-art Bottom-up Attention Models

Visual attention allows the human visual system to effectively deal with the huge flow of visual information acquired by the retina. Since the years 2000, the human visual system began to be modelled in computer vision and it became part of artificial intelligence: while learning focuses on repetitive data which can easily be modeled, computational attention focuses on abnormal, rare and surprising data which can hardly be learnt. Attention is a product of the continuous interaction between bottom-up and top-down information. While the bottom-up

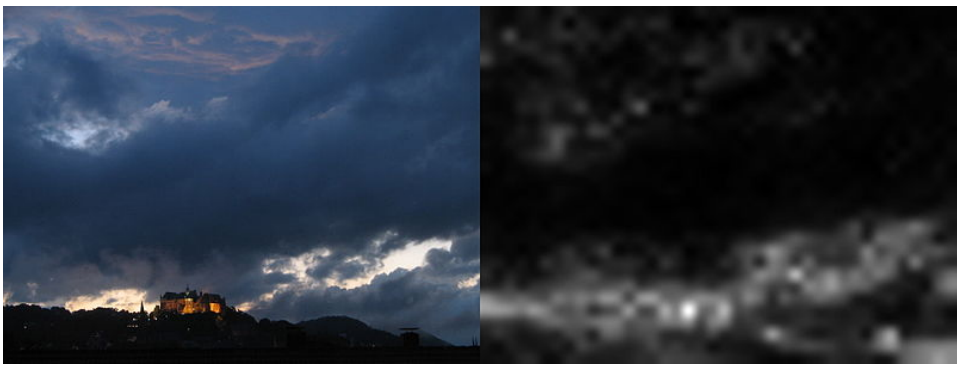


Figure 2.1. An example of saliency map. A view of the fort of Marburg (Germany) and the saliency map of the image using color, intensity and orientation.

information has been extensively investigated through saliency models, top-down influence on visual attention has been less investigated.

Computational visual attention tends to mimic human visual attention and focuses more deeply on the informative and important parts of images. In computer vision, the main approach to the implementation of visual attention includes bottom-up (mainly feature-based information which is known as reflex exogenous reaction) and top-down (learning-based information which refers to reflexive endogenous information). A lot of research was achieved on bottom-up attention models [1, 5, 24, 27, 32, 55, 56] but just only a few on top-down information. It seemed that top-down detectors were not efficient enough to improve results of visual attention models yet.

Saliency model predicts what attracts the attention. The results of such models are saliency maps which is a topographic representation of saliency which refers to visually dominant locations. Such results are obtained by different bottom-up features of an image such as intensity, color, orientation, uniqueness, contrast, etc. We apply RARE algorithm [55] to generate saliency map results as the main results of bottom-up information. This model uses a sequential bottom-up features extraction where first low-level features as luminance and chrominance are computed and from those results medium-level features as image orientations are extracted (see Fig. 2.2). Moreover, The RARE algorithm powerfully predicts human fixations compared with most of the freely available saliency models.

To compare our result with others, we apply other bottom-up attention models beside the RARE model. An Attention based on Information Maximization (AIM) model [4] is derived from a single principle, specifically, that attention seeks to select visual content that is most informative in a formal sense. AIM is compared to a variety of classic visual search paradigms revealing its efficacy in explaining an unprecedented range of effects such as pop-out, search efficiency, distractor heterogeneity, target and distractor familiarity, and visual search asymmetries among others. The model is described (see Fig. 2.3) with sufficient specificity to operate on real images and is revealed to have a greater capacity to predict human gaze patterns than existing efforts. The generality of the definition allows consideration of saliency of arbitrary ensembles of neurons and examples derived from neurons coding for spatiotemporal content and complex stimuli are presented.

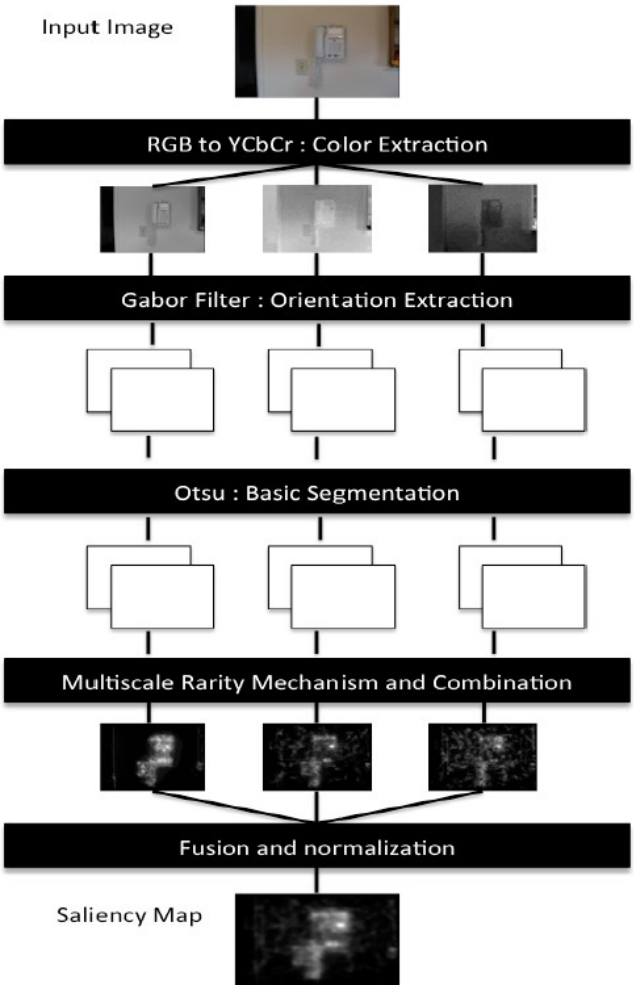


Figure 2.2. Diagram of RARE algorithm.

An Adaptive Whitening Saliency (AWS) model [19] is to ensure the invariance of the behavior that the contribution of an image point to optical variability elicits in the visual system. They proposed a coarse scheme linked to contextual (and data-driven) adaptation mechanisms that appears to produce a coarse figure-ground separation as well as illusory contours. They also made a flowchart of the specific implementation used with RGB images (in Fig. 2.4). The whitening procedure and the bank of filters used for spatial decomposition were exactly the same. The only modified parameters were the size of the input imagevaried as in the other modelsin the first experiment and the number of whitened chromatic components in the second experiment that involved hyperspectral images.

A Graph-based Visual Saliency (GBVS) model [22] consists of two steps: first forming activation maps on certain feature channels, and then normalizing them in a way which highlights conspicuity and admits combination with other maps. The model is simple, and biologically plausible insofar as it is naturally parallelized. They presented a method of computing bottom-up saliency maps (Fig. 2.5) which shows a remarkable consistency with the attentional deployment of human subjects. The method uses a novel application of ideas from

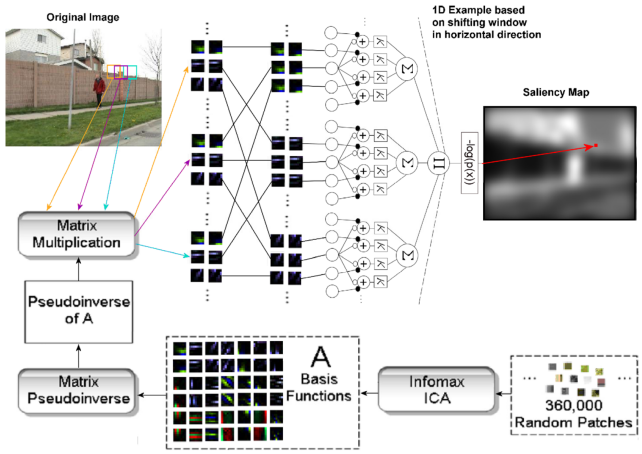


Figure 2.3. The framework that achieves the desired information measure.Shown is the computation corresponding to three horizontally adjacent neighbourhoods with flow through the network indicated by the orange, purple, and cyan windows and connections. The connections shown facilitate computation of the information measure corresponding to the pixel centered in the purple window. The network architecture produces this measure on the basis of evaluating the probability of these coefficients with consideration to the values of such coefficients in neighboring regions.

graph theory to concentrate mass on activation maps, and to form activation maps from raw features. Their model is extensible to multiresolutions for better performance, and it is biologically plausible to the extent that a parallel implementation of the power-law algorithm for Markov chains is trivially accomplished in hardware.

A Boolean Map based Saliency (BMS) model [72] computes saliency maps by analyzing the topological structure of Boolean maps based on a Gestalt principle of figure-ground segregation. The effectiveness of BMS is to use only color channels, while BMS should also be applicable to other feature channels, such as orientation, depth and motion. The pipeline of BMS is shown in Fig. 2.6. Given an image I, a set of Boolean maps B = (B_1, B_2, ..., B_n) is generated. Based on a Gestalt principle of figure-ground segregation, an attention map A_i is computed for each Boolean map B_i. Next, a mean attention map (A) is obtained through a linear combination of the resulting attention maps. Last, some post-processing is applied on the mean attention map to output a saliency map (S).

Furthermore, a bottom-up saliency model which employs region covariances as features (CVS) [18] efficiently encode local structure information. More significantly, region covariances provide a natural mechanism to nonlinearly integrate different features. This allowed their approach to produce especially accurate predictions for natural images containing texture elements or repeating patterns. As low-dimensional representations of image patches, region covariances capture local image structures better than standard linear filters, but more significantly, they naturally provide nonlinear integration of different features by modeling their correlations. Figure 2.7 shows saliency maps extracted at three different scales. As can be seen, as we moved to coarser scales, the model tended to capture the location of the visually most prominent region in the image. The final result shows the combined saliency map ob-

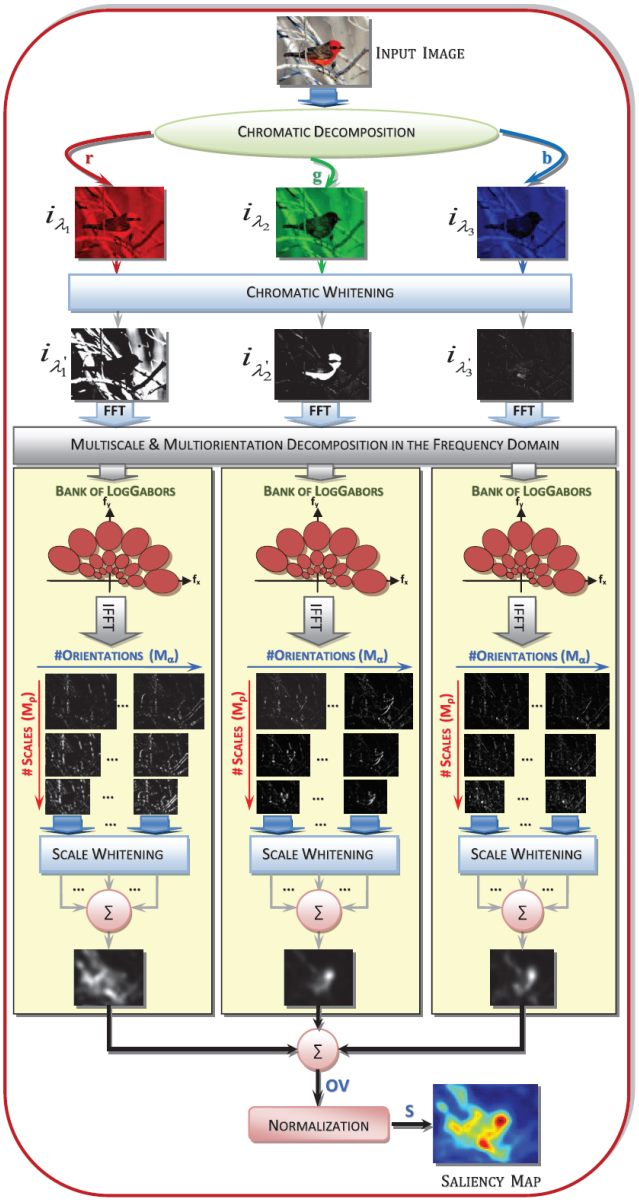


Figure 2.4. Optical variability and saliency estimation from RGB images. The chromatic components (i_1, ..., i_M), are approximated by (r, g, b) components, being r,g, and b the red, green and blue (broadband) components.

tained with the suggested multiscale approach using covariance features. In the master map, the red bell pepper in the image stands out among the surrounding green peppers.

Last but not least, another saliency technique based on center-surround approach (FES) [52] is method which based on estimating saliency of local feature contrast in a Bayesian framework. The distributions needed are estimated particularly using sparse sampling and kernel density estimation. In addition, the nature of method implicitly considers what refereed to as center bias in literature. This method can effectively compute the amount of saliency in images fast

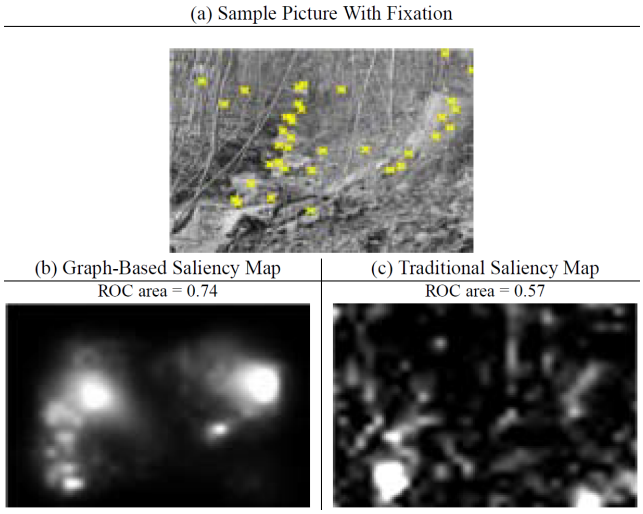


Figure 2.5. (a) An image from the data-set with fixations indicated using x's. (b) The saliency map formed when using (activation,normalization)=(graph (i),graph (iii)). (c) Saliency map for (activation,normalization)=(c-s,DoG).

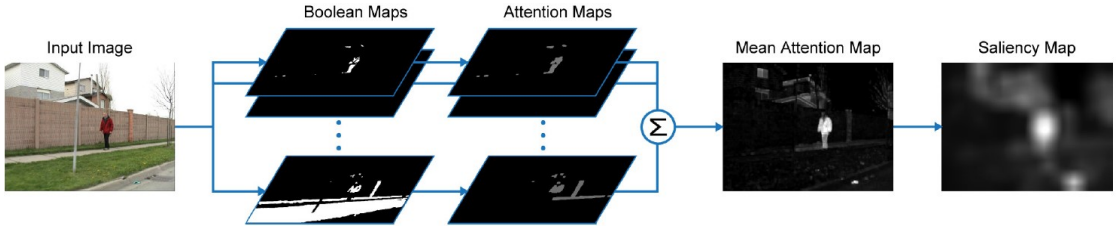


Figure 2.6. The pipeline of BMS.

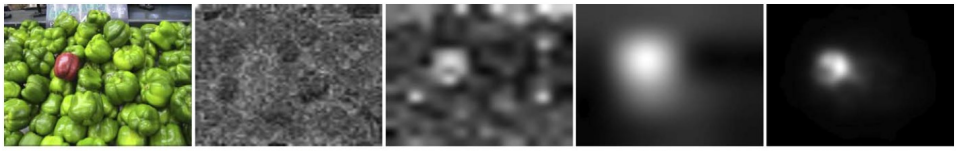
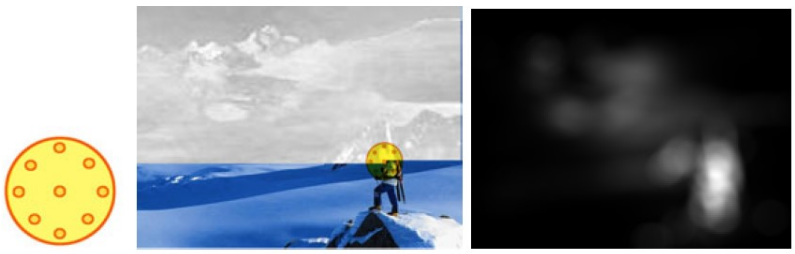


Figure 2.7. (a) Input image. (b-d) Predicted saliency maps obtained at different scales (from the finest to the coarsest). (e) Final saliency map according to the spatial coincidence assumption described in the text.

compared to other similar approaches. This method also use the center utilizing to measure saliency. Figure 2.8 shows an example of such a central pixel and sample pixels around it. This sparse sampling reduces the number of operations required to estimate the center utilizing and increases computation speed.



(a) (b) (c)

Figure 2.8. (a) A pixel and its selected surrounding samples in a window, (b) Procedure of applying a window, (c) A sample saliency map obtained, using the FES method.

2.3 State-of-the-art Top-down Attention Models

There are various kinds of top-down information which can be used in addition with bottom-up saliency [43] such as location-based, contextual-based or object-based models. The combination of face detection and low-level saliency provides already results improvements in [9]. The linear combination was weighted to give to faces the same weight that each one of the three bottom-up conspicuity maps (orientation, color, intensity) which means that the face map global weight was quite low. This helped the authors to deal with false positives from the face detector used at that time which was not optimal.

In [2], the author showed that the high-level features such as faces and people can enhance the model performance, but there was no any precise information related to the relative importance of those features. The author also stated that using a bad object detector could clearly decrease the model performance if it produces too many false positives. In [30], the authors dealt with the importance of people and cars for saliency detection. In [35], the authors introduced the idea of the use of object symmetry as top-down attention in images. In [71], target object features from the Pascal VOC object database are learned using a CRF-modulated dictionary. The saliency maps were really focused on the objects with a very high weight.

2.3.1 Face Acquisition

To generate top-down results of face detection, we apply a face alignment method [31] which is entitled "One Millisecond Face Alignment with an Ensemble of Regression Trees". This paper addressed the problem of face alignment for a single image as seen in Fig. 2.9. They propose a general framework based on gradient boosting for learning an ensemble of regression trees that optimizes the sum of square error loss and naturally handles missing or partially labelled data. In addition, they create two baselines **1)** is based on randomized ferns with random feature selection (EF) and **2)** is a more advanced version of this with correlation based feature selection (EF+CB) in order to accurately benchmark the performance, an ensemble of regression trees (ERT). This method provides us with good results although it misses a few face and occurs false detection.

Moreover, to generate face detection by using deep learning method, we applied a method from dlib library [63], a Convolutional Neural Network (CNN) based face detector. Dlib is a

Figure 2.9. Selected results on the HELEN dataset. An ensemble of randomized regression trees is used to detect 194 landmarks on face from a single image in a millisecond.

toolkit for making real world machine learning and data analysis applications in C++. While the library is originally written in C++, it has good, easy to use Python bindings. It loads a pretrained model and uses it to find faces in images. Moreover, the frontal face detector in dlib works really well. It is simple and just works out of the box. The CNN model is much more accurate than the HOG based model (in Fig. 2.10).

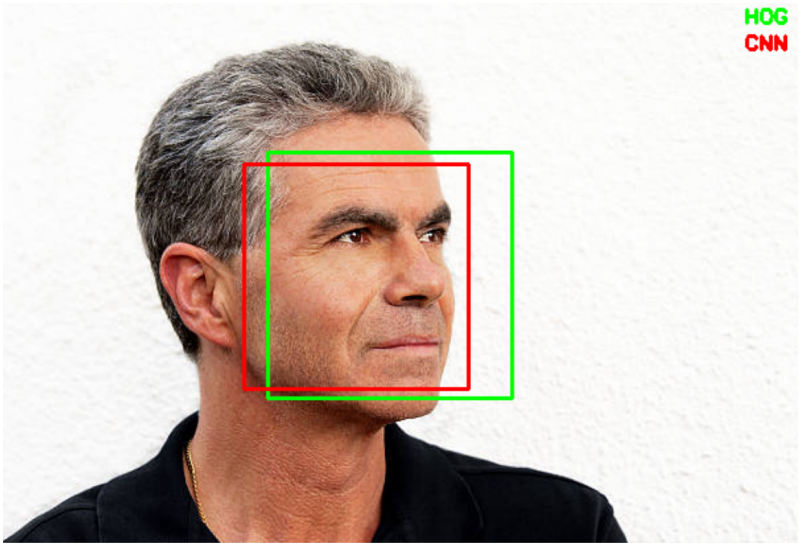


Figure 2.10. Comparison results between CNN and HOG based model (from Arun Ponnusamy).

2.3.2 Text Acquisition

To generate top-down results of text detection, we apply a scene text localization method [48] which is entitled "Real-Time Scene Text Localization and Recognition". This paper presented

an end-to-end real-time scene text localization and recognition method as seen in Fig. 2.11. The real-time performance is achieved by posing the character detection problem as an efficient sequential selection from the set of Extremal Regions (ERs). The ER detector is robust to blur, illumination, color and texture variation and handles low-contrast text. There are two important stages of classification to obtain texts in a scene **1)** estimate the probability of each ER being a character using novel features calculated with O(1) complexity (only ERs with locally maximal probability are selected) **2)** improve the classification using more computationally expensive features. This method provides us with bad results because we don't use real-time which helps to improve overall results from OCR.

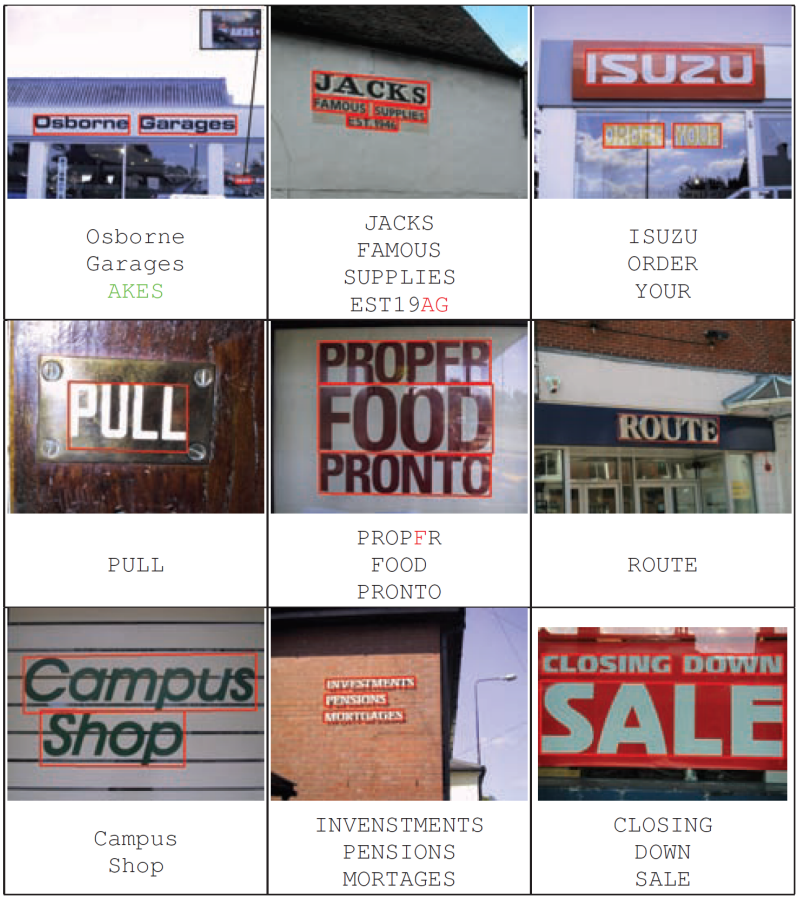


Figure 2.11. Text localization and recognition examples on the ICDAR 2011 dataset. Notice the robustness against reflections and lines passing through the text (bottom-left). Incorrectly recognized letters marked red, text recognized by the proposed method but not present in the ground truth marked green.

Furthermore, to generate text detection by using deep learning method, we applied a Connectionist Text Proposal Network (CTPN) model [66] that accurately localizes text lines in natural image. The CTPN is an efficient text detector that is end-to-end trainable. They propose an in-network Recurrent Neural Network (RNN) layer that connects sequential text proposals elegantly, allowing it to explore meaningful context information. These key technical developments result in a powerful ability to detect highly challenging text, with less

false detections (in Fig. 2.12). The CTPN is able to handle multi-scale and multi-language efficiently (e.g., Chinese and Korean).

Figure 2.12. CTPN detection results several challenging images, including multi-scale and multi-language text lines. Yellow boxes are the ground truth.

2.3.3 Object Acquisition

To generate object detection by using deep learning method, we applied, a real-time object detection system (YOLO9000) [51] that can detect over 9000 object categories. They focus mainly on improving recall and localization while maintaining classification accuracy. Their method used a hierarchical view of object classification that allows them to combine distinct datasets together. In addition, they also propose a joint training algorithm that allows them to train object detectors on both detection and classification data. First they improve upon the base YOLO detection system to produce YOLOv2, a state-of-the-art, real-time detector. Then they use their dataset combination method and joint training algorithm to train a model on more than 9000 classes from ImageNet as well as detection data from COCO. In brief, they apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities (in Fig. 2.13). However, our method doesn't use real-time detection and customizes into three categories: animal, person, and transportation.

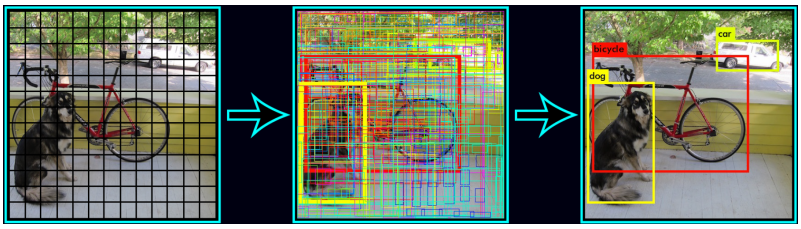


Figure 2.13. YOLO9000. YOLO9000 can detect a wide variety of object classes in real-time. The network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

2.4 State-of-the-art Deep Neural Network Attention Models

After an arrival Deep Neural Network (DNN)-based models, most researchers have switched their research direction to focus more on obtaining an end-to-end DNN saliency model which naturally integrates top-down information. Since 2014, DNNs have changed the saliency paradigm. The deep features were first used in eDN model [67]. Then, DeepGaze1 model [37] showed that the DNN features trained on object recognition were very useful for saliency detection. This finding seems logical as objects apparently represent the regions of interest in images. Since then, a variety of models used fine-tuned mixes of features from several deep learning models which naturally incorporated top-down information (i.e., face and text) during the learning process.

However, in [39], the authors showed that the importance of bottom-up attention was underestimated by DNN-based models. In their experiments, a simple bottom-up model could outperform a state-of-the-art DNN model when the images contained less top-down information. This demonstrated that DNNs too much neglected the bottom-up aspect of visual attention, and they were mostly trained to detect the attractive top-down objects rather than detect saliency itself. Moreover, they could not easily adapt to images in a different context from their training set and they had the structural issue to provide a result that could not really be explained in an explicit way.

In [34], the authors showed that, compared to old detectors which were not accurate enough, current detectors (i.e., face detectors), when mixed to bottom-up saliency maps provide significantly better visual attention results. It is therefore possible to integrate the top-down information into classical bottom-up attention models in a hand-crafted way.

To make comparison results between ours and others by using deep learning methods, we used several methods such as LSTM-base saliency attentive model (SAM-ResNet) [13], semantic gap in saliency prediction (SALICON) [25], and so on. In SAM-ResNet model, gaze maps are computed with a feed-forward network, and it can predict accurate saliency maps by incorporating neural attentive mechanisms. The core of our solution is a Convolutional LSTM that focuses on the most salient regions of the input image to iteratively refine the predicted saliency map. Moreover, to tackle the center bias present in human eye fixations, their model can learn a set of prior maps generated with Gaussian functions. In Fig. 2.14, it shows overview of saliency attentive model. After computing a set of feature maps on the input image through a new architecture called Dilated Convolutional Network, an Attentive Convolutional LSTM sequentially enhances saliency features thanks to an attentive recurrent mechanism. Predictions are then combined with multiple learned priors to model the tendency of humans to fix the center region of the image. During the training phase, we encourage the network to minimize a combination of different loss functions, thus taking into account different quality aspects that predictions should meet.

Saliency in Context (SALICON) [25] is an ongoing effort that aims at understanding and predicting visual attention. They focus on narrow the semantic gap with an architecture based on Deep Neural Network (DNN). It leverages the representational power of high-level semantics encoded in DNNs pretrained for object recognition. Two key components are fine-tuning the DNNs with an objective function based on the saliency evaluation metrics, and integrating information at different image scales. In addition, the architecture of SALICON (in Fig. 2.15) consists of a DNN applied at two different image scales. The last convolutional

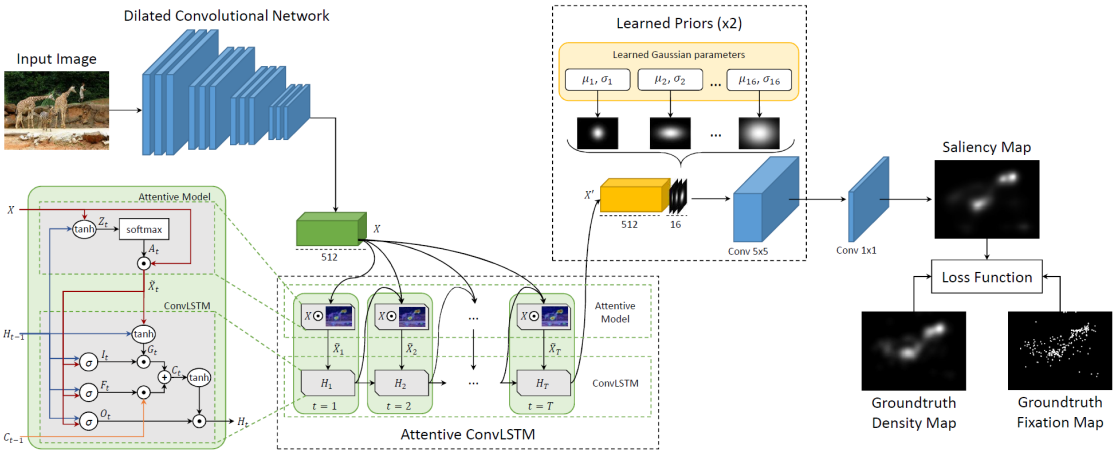


Figure 2.14. Overview of Saliency Attentive Model (SAM).

layer in the pretrained network feeds a randomly initialized convolutional layer with one filter that detects the salient regions. The parameters are then learnt end-to-end with back-propagation. They use objective functions to optimize some common saliency evaluation metrics. Moreover, OpenSALICON (oSALICON) [65] is an open source implementation of the SALICON saliency model.

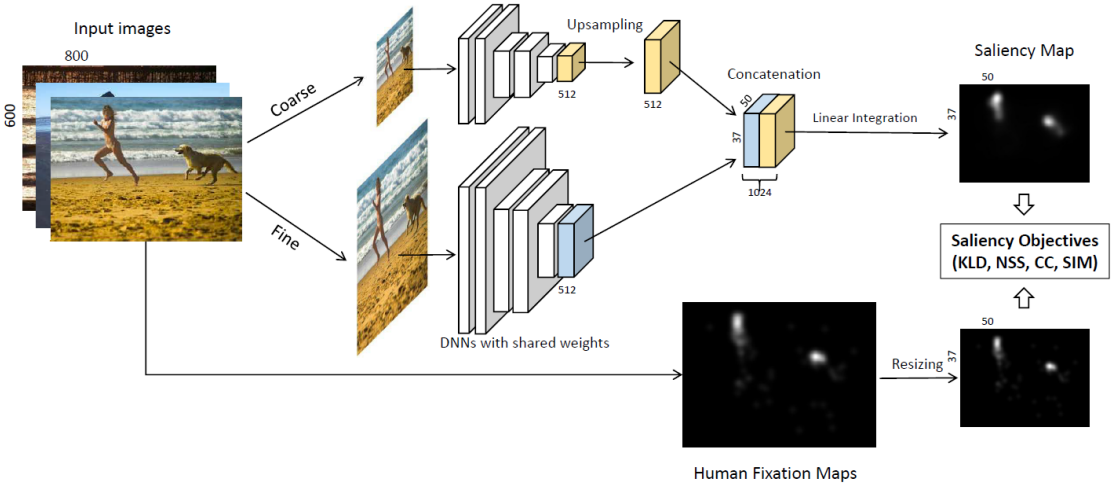


Figure 2.15. Learning of the DNN architecture for saliency prediction.

Furthermore, a deep feature-based saliency model (DeepFeat) [40] is developed to leverage the understanding of the prediction of human fixations. Traditional saliency models often predict the human visual attention relying on few level images cues. Although such models predict fixations on a variety of image complexities, their approaches are limited to the incorporated features. They aim to provide an intuitive interpretation of convolutional neural network deep features by combining low- and high-level visual factors. In Fig. 2.16 presents example deep features of nine representative images from layers 1, 10, 20, 30, 40, and 49 of a residual network.

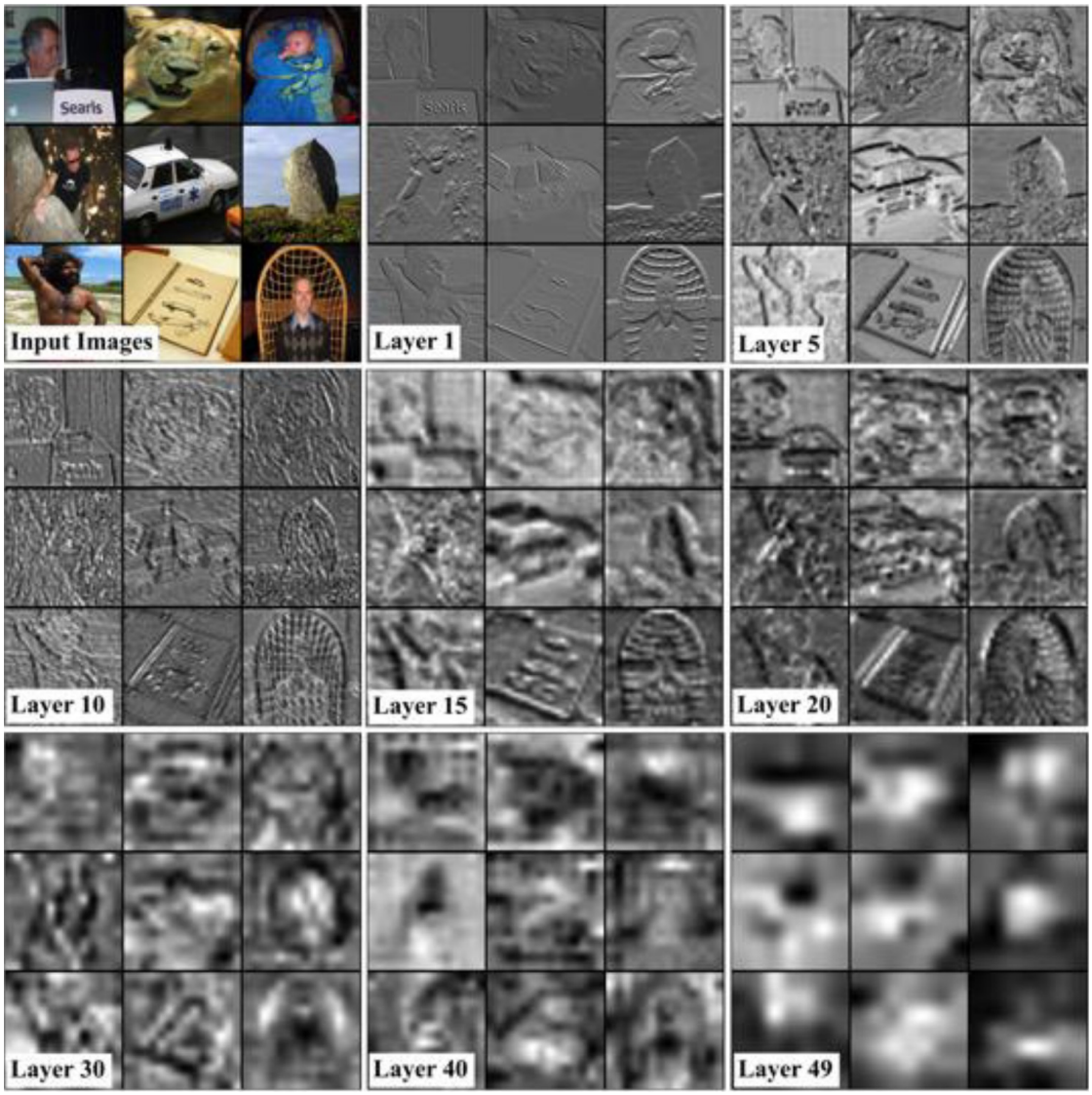


Figure 2.16. Visualization of example features of layers 1, 10, 20, 30, 40, and 49 of a deep convolutional neural network. In each layer visualized, one convolution response image is selected randomly and presented.

They formalize DeepFeat as a fusion of bottom up and top down visual factors using a simple combination strategy. The architecture of the DeepFeat can be visualized in Fig. 2.17. A bottom up visual cues are represented by a CNN pretrained features. For the purpose of bottom up computation, the fully connected layer of the CNN is removed. So, two scales of the deep features are exploited, fine and coarse scales. The fine scale is original size of the extracted deep feature.

In brief, they proposed a deep feature-based saliency model, codenamed DeepFeat, which combines bottom up and top down visual factors obtained from pre-trained deep features. To validate the performance of the DeepFeat model, they investigated four different implementations of the Deep-Feat model using four evaluation metrics over the MIT1003 dataset.

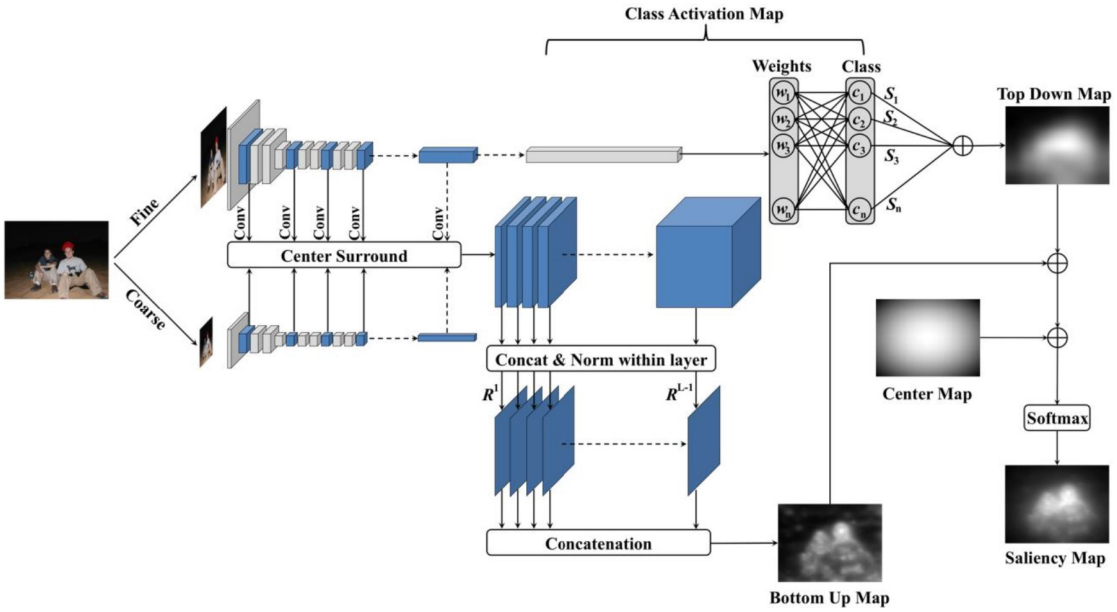


Figure 2.17. Architecture of the DeepFeat Model. The architecture consists of a combination of bottom up and top down features. The bottom up features are computed using two scales of a CNN. The top down map is computed using the class activation map of the full-scale CNN. The result of combining the bottom up and top down map is weighted by a center bias map.

Another model called SCAFI [62] which predicts saliency based on semantic and contrast-aware feature integration. SCAFI is a heuristic model which shares the same inspiration of SCA but achieves much better performance by directly pooling the semantic information out of the VGG net. Compared to traditional deep model, SCAFI is learning free and highlights both the semantic interests and the bottom up contrast. The architecture of their dynamic feature integration model is illustrated in Fig. 2.18 (Part II). After preprocessing, the RGB input image is fed into the pre-trained VGG network. Then they collect feature maps from all the convolutional layers except the final three fully connected layers. Each filter is corresponding to a single feature map generated based on the output of its Relu layer. All feature maps are resized to match the spatial resolution of the input.

In Fig. 2.19 shows some examples of the extracted feature maps. It clearly shows that the VGG features can effectively highlight the semantic objects, e.g. car wheels and faces, in each input image. As visualized in Fig. 2.20, feature maps extracted from different layers describe the image contents at different scales and in different views. The feature maps of the shallow layers contain more structural details such as edges and textures, while those from the deep layers highlight more about the object and object parts with specific semantics.

In brief, they proposed a heuristic framework to integrate both semantic-aware and contrast-aware saliency which combines bottom-up and top-down cues simultaneously for effective eye fixation prediction. Experimental results on 5 benchmark datasets and artificial images demonstrate the superior performance and better plausibility of the proposed SCAFI model over both classic approaches and the recent deep models.

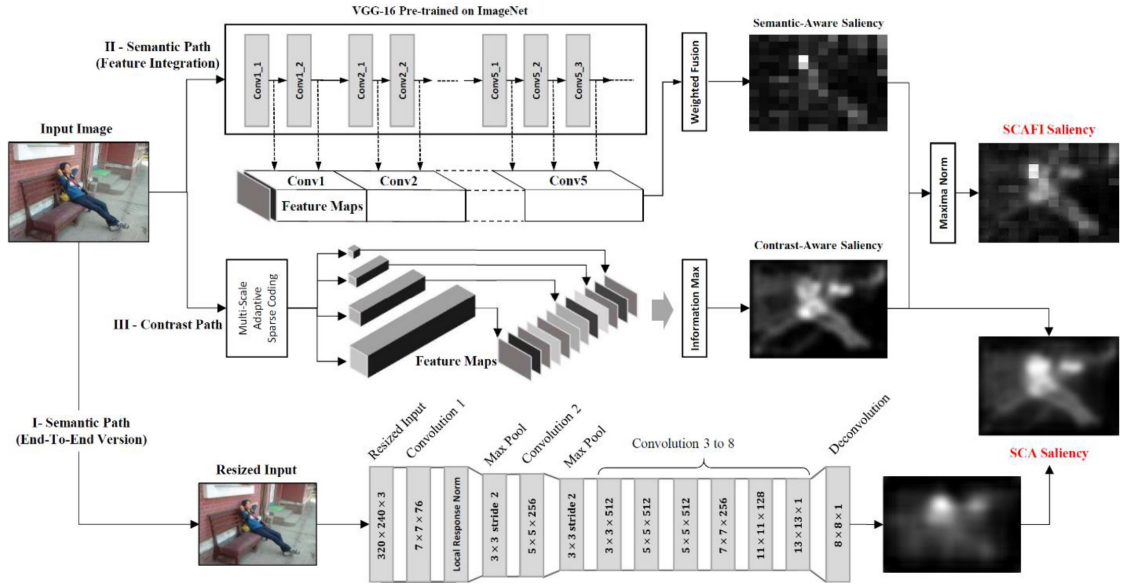


Figure 2.18. Semantic and Contrast-Aware Saliency Model. Part I : An End-to-End deep neural network customized as an alternative version of the semantic-aware saliency (SAS). Part II: A deep feature integration module as the default SAS pathway, which directly collects and aggregates 2D feature maps from the convolutional layers of pre-trained VGG net. Part III: A contrast-aware saliency (CAS) pathway that discovers high contrast salient patterns within the image context based on multi-scale sparse representation and information maximization. The temporal outputs of the SAS and CAS pathways are all resized to the original image size and then integrated using maxima normalization. The integration of I and III corresponds to their previous model SCA, while the combination of II and III results in their new model SCAFI.

In addition, a model called ML-Net [14] predicts saliency maps exploiting a non-linear combination of features coming from different layers of the network. We also present a new loss function to deal with the imbalance issue on saliency masks. An overview of the ML-Net architecture is presented in Fig. 2.21. Saliency prediction can benefit from both low level and high level features. That's why they build a saliency prediction model which combines features extracted at multiple levels from a Fully Convolutional Neural Network (FCN). Since the role of this network in their model is that of extracting features, instead of predicting a saliency map, they call this component feature extraction network. An encoding network is then designed to weight and combine feature maps extracted from the FCN, and training is performed by means of a loss function which tackles the problem of imbalance on saliency maps.

Next, an ensemble of Deep Networks (eDN) [68] makes no assumptions about what lower-level features (color, contrast, etc.) or higher-level concepts (faces, cars, text, horizon) attract the eyes. They allow the hierarchical models to learn such complex patterns from gaze-labeled natural images and follow an entirely automatic data-driven approach that performs a large-scale search for optimal features. In Fig. 2.22, for each image in the training set, they randomly pick 10 salient samples from the top 20% salient regions and 10 non-salient samples from the bottom 70% salient areas of the empirical saliency map. At these selected locations,

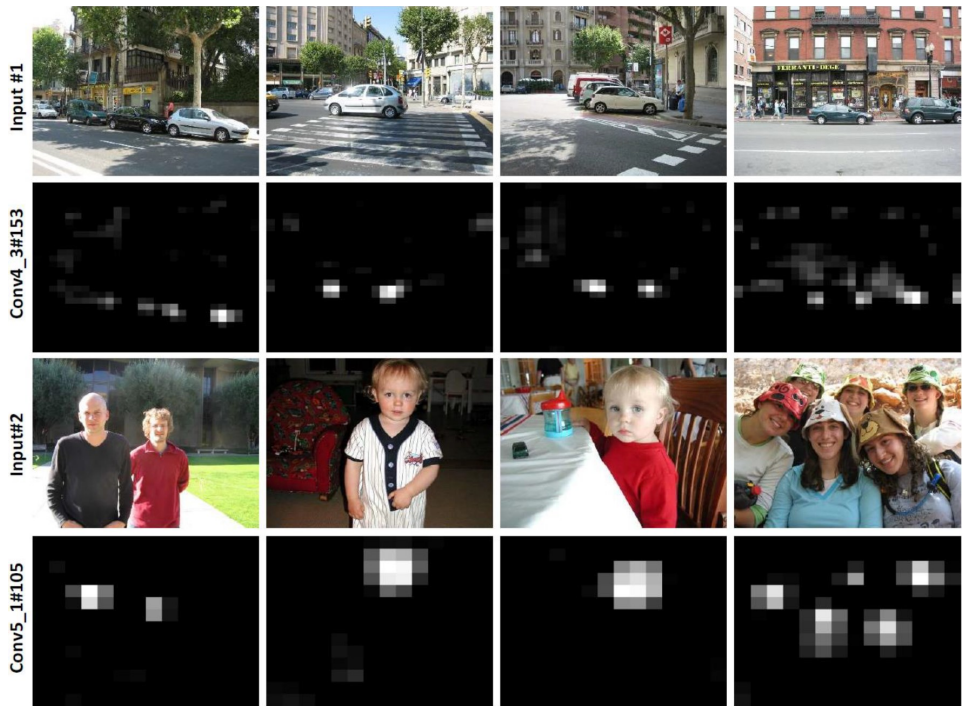


Figure 2.19. Examples of the feature maps extracted using pre-trained VGG network. It clearly shows that the VGG features can effectively highlight important objects with certain semantics. e.g. car wheels (#1) and faces (#2).

features are extracted from the image and normalized (over the entire training set) to zero mean and unit variance. Finally, the labeled feature vectors are fed into an L2-regularized, linear, L2-loss SVM, which is trained to predict for each location in a new test image its probability of fixation.

Then a model called DeepGaze II (DGII) [38] is developed to predicts where people look in images. This model uses the features from the VGG-19 deep neural network trained to identify objects in images with no additional fine-tuning (rather, a few readout layers are trained on top of the VGG features to predict saliency). This model is also a strong test of transfer learning. Moreover, information gain explained is able to quantify precise differences between models, and shows the clear improvement gained by DeepGaze II. The architecture of DeepGaze II is shown in Fig. 2.23, but they do not retrain the VGG features. While this reduces the model space, it also greatly reduces the number of parameters that must be learned from data, reducing the chance of overfitting. However, a low-level intensity contrast features (ICF) [39] model uses simple intensity contrast features to achieve better performance than all models that do not use pre-trained deep features. The ICF model is isotropic (does not even have access to orientation filters) which makes its performance improvement relative to earlier models even more remarkable.

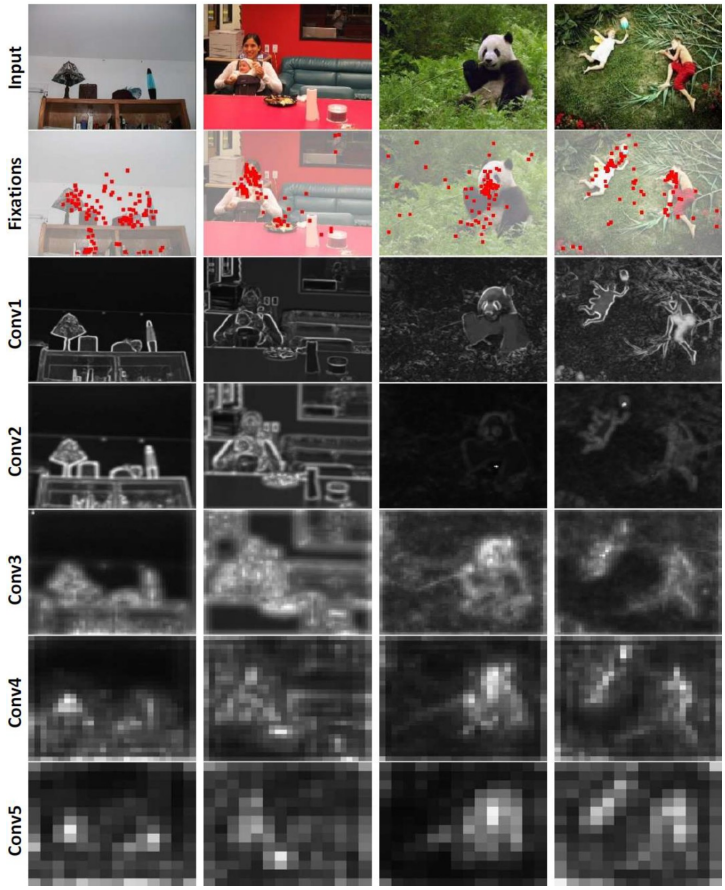


Figure 2.20. Feature maps extracted from different layers of the pre-trained VGG network. The shallow layers contain more structural details while the deep layers highlight more about the object and object parts with specific semantics.

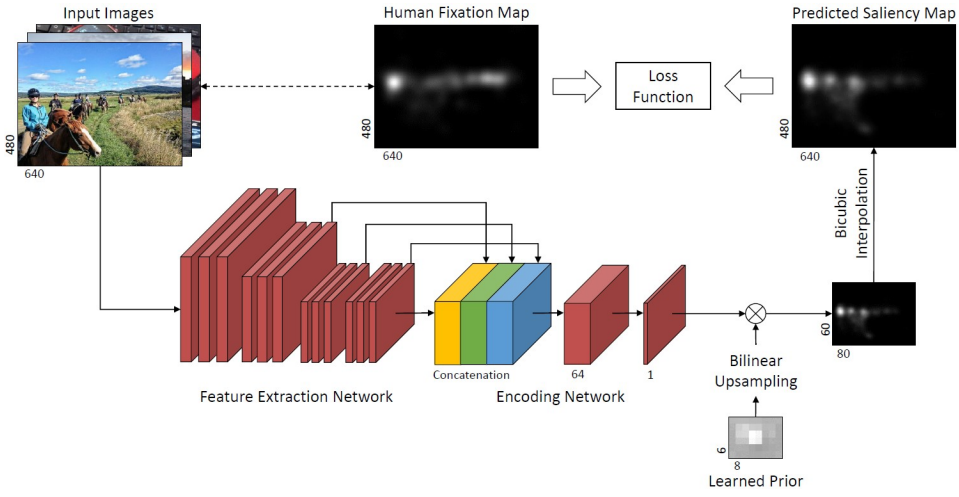


Figure 2.21. Architecture of ML-Net.

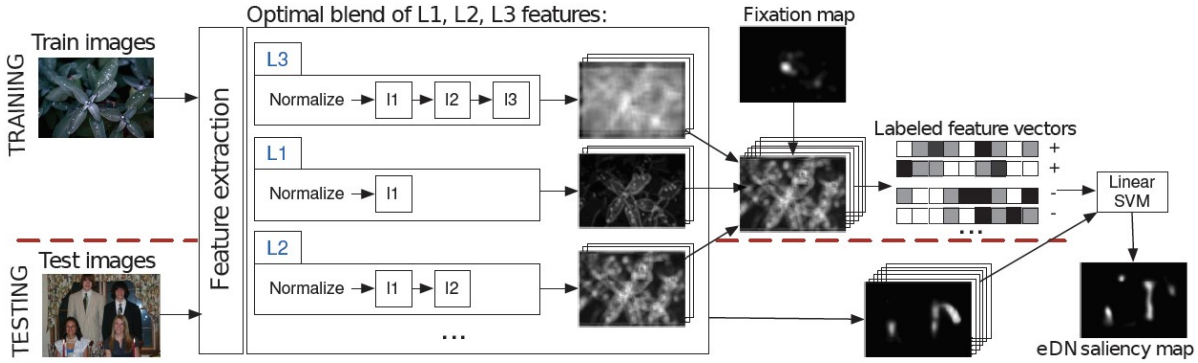


Figure 2.22. Schematic diagram of eDN pipeline. Good L_i multilayer feature extractors are found by guided hyperparameter search (not shown) and combined into an optimal blend. Resulting feature vectors are labeled with empirical gaze data and fed into a linear SVM.

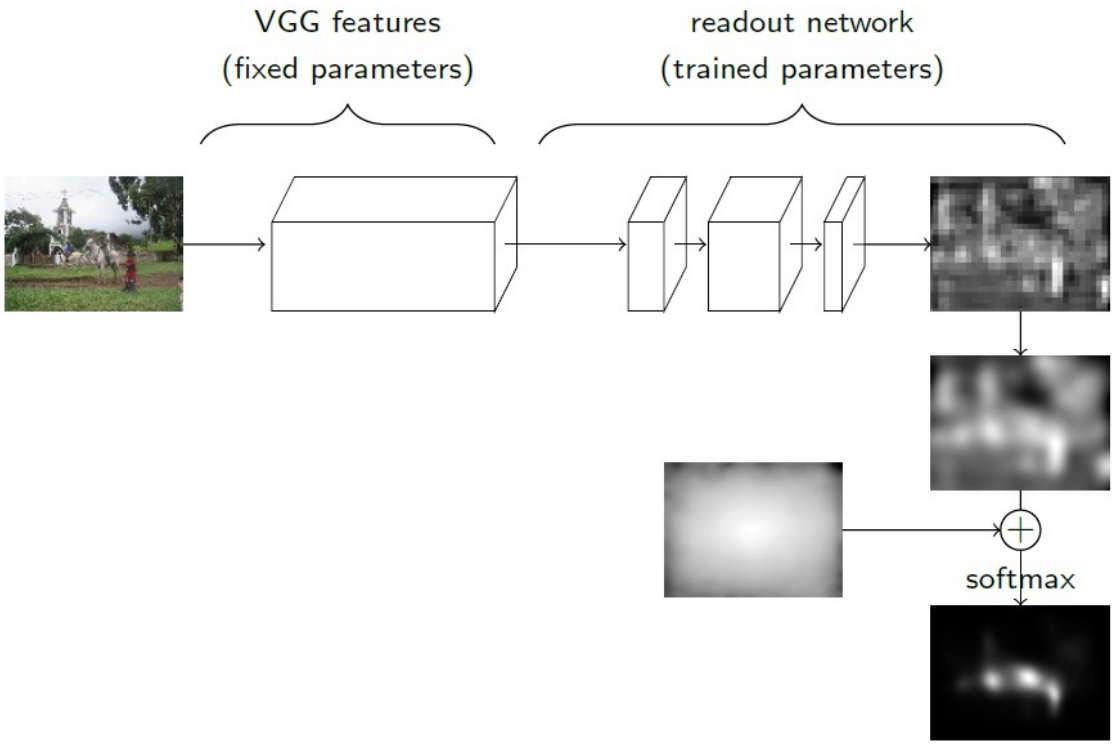


Figure 2.23. The architecture of DeepGaze II. The activations of a subset of the VGG feature maps for a given image are passed to a second neural network (the readout network) consisting of four layers of 1x1 convolutions. The parameters of VGG are held fixed through training (only the readout network learns about saliency prediction). This results in a nal saliency map, which is then blurred, combined with a centre bias and converted into a probability distribution by means of a softmax.

2.5 In Brief

The summaries of this chapter are:

- State-of-the-art bottom-up attention models: mainly focus on feature-based information (reflex exogenous reaction). There are a lot of research which were achieved on this model.
- State-of-the-art top-down attention models: mainly focus on learning-based information (reflexive endogenous information). They combined face detection and low-level features to deal with false positive from the face detector which was not optimal. Moreover, the high-level features such as faces and people can improve the model performance.
- State-of-the-art deep neural network (DNN) attention models: mainly focus more on obtaining and end-to-end DNN saliency models which naturally integrates top-down information. A variety of models used fine-tuned mixes of features from several deep learning models which naturally incorporated top-down information (i.e., face and text) during the learning process.

Part II

Top-down and Bottom-up Information Relative Importance

Chapter 3

Bottom-up Attention Maps with High-level Semantic Information

3.1 Bottom-up Attention Maps with Text Detection

3.1.1 Objective

As it is the first experiment of this research, we try to understand some fundamental knowledge for visual attention modeling such as the methods for saliency visuals and traditional text detection. First of all, we review some state-of-the-art bottom-up saliency models (in section 2.2). Then we go further on some state-of-the-art top-down saliency models (in section 2.3). Finally, we combine bottom-up and top-down information together in order to obtain higher attention. The main goal of these combination methods is to prove that there is a new way or technique that could provide a better visual attention model than the saliency-based visual attention modeling. As the result, we can get some good points of view in order to go to next experiment.

3.1.2 Method

We intend to understand the basic concept of saliency visuals by using an existing MATLAB code from RARE model [55]. In the experiment, an initial image seen at the left side of Fig. 3.1 were selected. As a result, its corresponding saliency map result (generated as a heat map) is given as depicted at the right side of the same figure. In the heat map image, the red and blue areas refer to the highest and lowest attention of the given image, respectively.

For text detection part, the existing code written in Python from [48] was used in this experiment. As seen in Fig. 3.2, the left-side image represents the input image, while the text in the bounding boxes at the right-side image represents the result obtained after the text detection. Next, both the black and white and Gaussian filters (up to 121 kernels is used to make the output results more feasible) are applied and then produce the results depicted in Fig. 3.3.

In addition, we propose a workflow for combining bottom-up and top-down information as seen in Fig. 3.4. According to Fig. 3.4, the general workflow of this proposed method is described as below:

- Input: An image is used as input

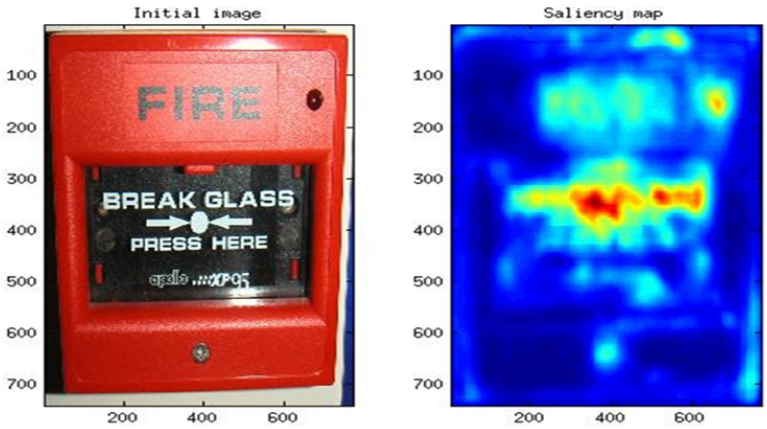


Figure 3.1. An example result of bottom-up saliency map. (left) Input image, (right) heat map.

Figure 3.2. An example result of traditional text detection model. (left) Input image, (right) text detection result (white rectangular boxes).

Figure 3.3. An example result of black and white and Gaussian filtering. (first image) Result of text detection, (second image) result of black and white filtering that white area is detection, and black area is background, (last image) result of Gaussian filtering.

- Output: A heat map
- Process:

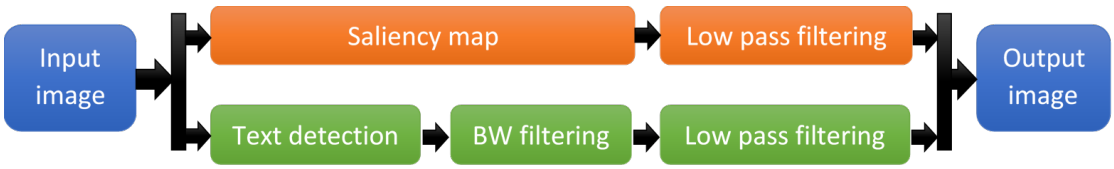


Figure 3.4. Workflow of a method that combines both saliency map and text detection features together.

- An initial image is taken as input and we use saliency-based method to produce an attention map in which the location or pixels where human pay most attention on are displayed.
- The same initial image is used again to detect the location or pixels or zones where all the textual information are found in the image.
- Next, the black and white filter is used to convert the detected area/surface to black&white surface/area/background.
- The low-pass filter (i.e., Gaussian filter) is then applied for smoothing the detected textual areas. This allows the transformed images to be more natural, feasible and readable to the users.
- Finally, an output image (as heat map image) are established after combining the results obtained from both saliency and text detection methods. Three different combination methods have already described above.

According to the observation, it shows that the results given by text detection (TD) method could provide more extra features to be added to those given by bottom-up saliency (SM) method. Therefore, three different proposed techniques has been proposed by combining both obtained results (from SM and TD methods) altogether as follows:

$$RoSM + RoTD \tag{3.1}$$

$$RoSM * RoTD \tag{3.2}$$

$$(w * (RoSM * RoTD)) + ((1 - w) * (RoSM)) \tag{3.3}$$

where $RoSM$ and $RoTD$ indicate the results obtained from SM and TD methods, respectively. Moreover, the parameter w represents the weight for balancing the *(RoSM * RoTD)* and *(RoSM)*.

3.1.3 Results

Equation 3.1 is the simplest method to get the result. But, some cases it provides us with good result because it is not complicated. However, after we apply this method, we can get suitable result because we can get a bit higher attention than saliency alone (in Fig. 3.5). This method, we only add the result from saliency map and text detection without deleting

or modifying the raw result. For equation 3.2, it provides us with areas which have the same

Figure 3.5. An example result of addition between saliency map and text detection. (first image) result of saliency map, (second image) result of Gaussian filtering on text detection, (last image) result of addition.

attention. For the rest attention, it will automatically delete it or regard as low attention in the image. This method we apply the algorithm the same as addition. The different point is that we change from addition to multiplication. As the result, we observed that the result of text detection has high frequency than the result of saliency alone. Thus, this method doesn't make sense for high attention (see Fig. 3.6). In equation 3.3, we can manage the

Figure 3.6. An example result of multiplication between saliency map and text detection. (first image) result of saliency map, (second image) result of Gaussian filtering on text detection, (last image) result of multiplication.

final result by modifying value of weight which is in range [0-1]. When we compare this result to previous results above, this method provides us with better result because it keeps attention from saliency map and give more attention when texts appear in the image (see Fig. 3.7). According to the evaluation, a global result of our proposed method is shown in Fig. 3.8.

3.1.4 Discussion

We tested with a dataset of ICDAR2013 [64]. Among our proposed methods here, the last method provided us with the best result because it has a weight to manage the result of saliency map and text detection. However, there is a problem caused by the text detection algorithm and some false positive results. The selected text detection method does not provide

Figure 3.7. An example result of weighting between saliency map and text detection. (first image) result of saliency map, (second image) result of Gaussian filtering on text detection, (last image) result of weighting.

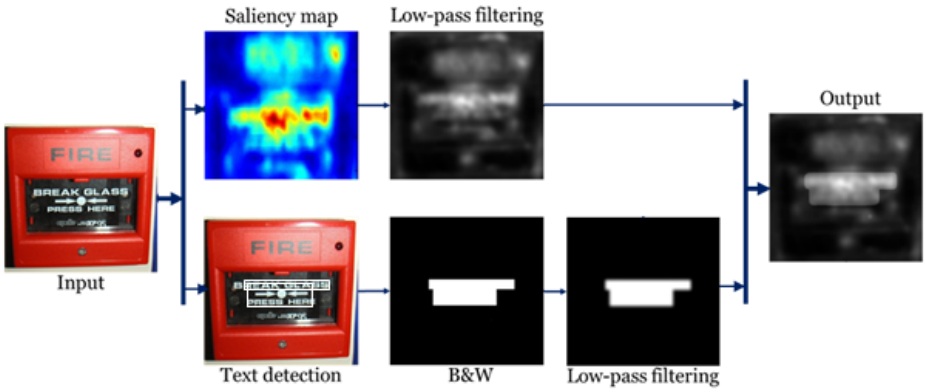


Figure 3.8. A global process of the combination between bottom-up and top-down information.

good or enough results yet because many non-textual objects in the images have been detected. Therefore, another better text detection method should be used to improve the whole system performance.

Moreover, the third proposed combination method seems provide better result than two others, however the value assigned to the weight w is not good yet. Therefore, the tuning process for finding the best value of w will be done in the next step.

3.2 Bottom-up Attention Maps with Text and Face Detection

3.2.1 Objective

This experiment part intends to study the influence of object-based (face and text) top-down information on bottom-up saliency maps. We propose a simple yet effective fusion scheme that can be applied on any bottom-up saliency model depending on the object detector effectiveness and the object size. The evaluation results show that it is possible to highly improve classical bottom-up saliency models with the arrival of better object detectors. In the future, such attention models can become as effective as deep learning based attention models while keeping them more generic and avoiding underestimating bottom-up features.

The objective of this part is to show how to simply add top-down information to any bottom-up saliency models in a generic way. This part is mainly focused on traditional methods of text and face detection.

3.2.2 Method

Several simple fusion algorithms are made (see details in section 3.2.4.2) which can easily add to bottom-up saliency maps. As the result, it demonstrates that if the detector is not good enough, it is better not to use it at all and only use the bottom-up information. Moreover, detection result using a general purposed object detector may include too many objects which are less likely to attract visual attention. If the detector is good, a simple average can be almost as good as a more complex weighted average. When the detector becomes very good, then the weighted average really makes sense. This is even more the case when several top-down features are mixed to bottom-up and some might be more important than others. Workflow of this method is seen in Fig. 3.9. Here, we add one more feature, fact detection, on previous experiment because in a scene image if there is one or more human faces, viewers absolutely see the face(s). This experiment is similar to the previous experiment. However,

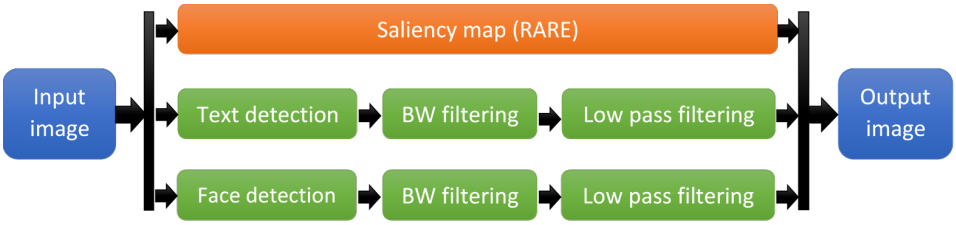


Figure 3.9. Workflow of our proposed model that combines saliency maps with text and face detection.

we don't use Gaussian filtering on the result of saliency map, and face feature is added on this workflow. So, it means that our proposed method is combined from bottom-up saliency map with text and face detection. Black and White (BW) filtering refers to white area as text or face detection and black area as the background. We use the OSIE dataset [69] to test our proposed method.

3.2.3 High-level Features and Detectors

Low- and high-level information are both important to predict human gaze accurately [39, 69]. In [69], the relative importance of different features was used to evaluate the model performance, which was computed by a linear Support Vector Machine (SVM) classifier. In terms of importance, it shows that face, text, and gaze direction are the three main features. In addition, color, orientation, or intensity still have an interesting influence especially when human faces and text were absent [39]. However, in [69], the result cannot be easily applied to any bottom-up attention model, while this research intends to be able to use them in a generic way with any model.

For the object detectors, we focus on **1)** face and **2)** text. Inside these two important features, the research tries to understand the importance of the size (big face and text versus small face

and text). The Object and Semantic Images and Eye-tracking (OSIE) dataset [69] provides us with a set of images and the manually segmented masks for text in Fig. 3.10 (second image) or face in Fig. 3.10 (third image) along with the eye tracking fixation map in Fig. 3.10 (last image).

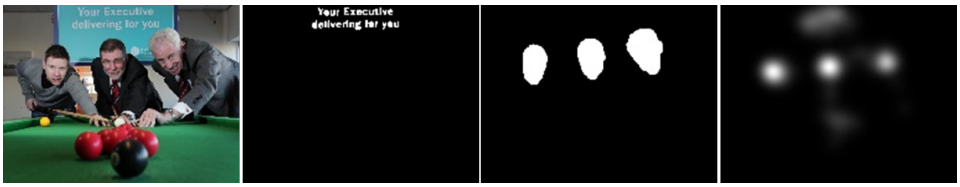


Figure 3.10. Text and face features extraction and eye-tracking fixation map from the OSIE dataset. (first image) Input image, (second image) text features, (third image) face features, and (last image) eye-tracking fixation map.

3.2.3.1 Text Features

Based on the OSIE dataset, we only select the images containing text in Fig. 3.11. When checking the corresponding eye-tracking maps, big text seems much more interesting than small text (around twice more interesting). This consideration pushed us to check the difference between big in Fig. 3.11 (second image) and small in Fig. 3.11 (third image) text. Then we used the OSIE masks to separate text regions between big text (more than 29 pixels in height) and small text (less than 29 pixels in height). This threshold depends of course on the image size, but all the images in the database are of the same size here. The thresholds are taken according to empirical experiment.

Figure 3.11. Our defining big and small text features and eye-tracking fixation map from the OSIE dataset. (first image) Input image, (second image) big text features, (third image) small text features, and (last image) eye-tracking fixation map.

3.2.3.2 Face Features

In the same way, this step uses the manually segmented masks for the images containing face in Fig. 3.12. Even if the difference in terms of eye-tracking maximum is less obvious between big faces and small faces than between big text and small text, this work separates big faces (used a threshold of 76 pixels in height) from small faces (less than 76 pixels in height) the same way as text. Again, the size of the images is always the same here, but for other datasets this threshold should be computed relatively to the mage size to be used on other datasets.

Here this work only takes into account the frontal faces as heads viewed from rear or from the side have less chances to be correctly detected by an automatic face detector.

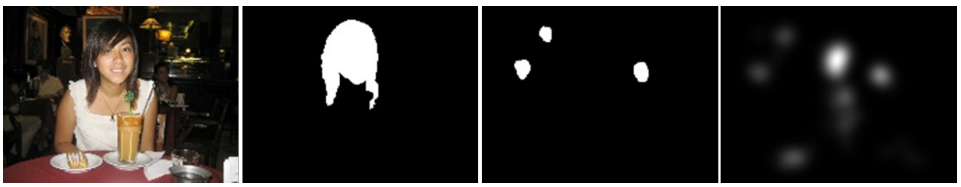


Figure 3.12. Our defining big and small face features and eye-tracking fixation map from OSIE dataset. Big and small face features are defined by us as the text features. (first image) Input image, (second image) big face features, (third image) small face features, and (last image) eye-tracking fixation map.

3.2.3.3 Object Detectors

First we define a perfect detector which is simply the human-based masks already segmented in [69].

For face detection in Fig. 3.13, we use a state-of-the-art face detector based on the DLIB library [63], more detail in section 2.3.1. We used the classical Histograms of Oriented Gradients (HOG) feature followed by a SVM classifier which has a good face detection rate [15]. On this detection, we added the face template approach based on a cascade of classifiers from [31] which exhibited good results for frontal faces, with few false positives.

For text detection in Fig. 3.14, we use an older approach which is integrated into the OpenCV library [48], more detail in section 2.3.2. This detector used Extremal Regions (ERs) which are robust to several image transformations. A second step is used in the algorithm: OCR helps to improve overall results. However, we did not use any OCR results. For real-life images, this detector results are poor with both missed and false detection.

Figure 3.13. Applying a traditional face detection method. (first row) Input images, (second row) result of face detection. The results contain either big faces (brighter), either small faces, and either both big and small faces.

Figure 3.14. Applying a traditional text detection method. (first row) Input images, (second row) result of text detection inside white bounding boxes, and (last row) converting text detection areas into white to indicate text features. The results contain both big (brighter) and small texts.

3.2.4 Experiment

The experiment intends to provide us with a clear view on how top-down information effects bottom-up information by adding text and face detection. It can be divided into two questions: **1)** how to extract weights for different kind of top-down information and **2)** how to mix the top-down information to bottom-up in a simple way.

To do so, we choose the top-down information (see Fig. 3.15, bottom-right), while the bottom-up saliency comes from the RARE model [55] (see Fig. 3.15, bottom-left). This approach was purely bottom-up (no additional centered Gaussian or learning-based information), and it considered both local information and global information through a rarity approach. By considering the MIT saliency benchmark [7], this model is bellow most of the DNN-based models.

3.2.4.1 Weights for Face and Text

While in [69], the features weight is computed by the means of a classifier, we choose here to use the experimental data that we have in the OSIE dataset to extract a meaningful individual weight for each feature of interest. For that purpose, we decided to measure the average maximum eye gaze attractivity on all the OSIE images for big and small text and face masks. As it can be seen in the schema in Fig. 3.16, the eye-tracking map is multiplied by the binary mask which will provide the eye-tracking intensity on the object of interest. Then the maximum of these values is averaged over all the images in the dataset providing a weight for the given feature. For weight of big text, after 75 images we are stabilized between 0.75 and 0.78, and for weight of small text, after 75 images we are stabilized between 0.31

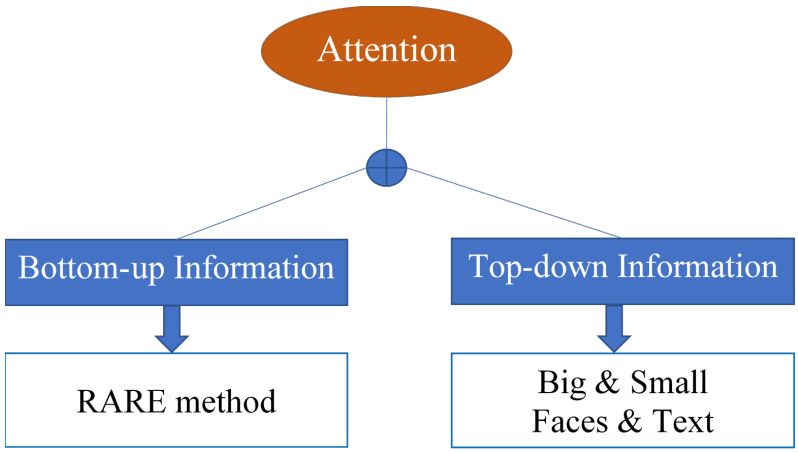


Figure 3.15. Components of bottom-up and top-down attention used in our experiment.

and 0.34. For weight of big face and small face, after 75 image we stabilized between 0.81 and 0.84 and between 0.64 and 0.67, respectively. As a result, between 75 images and 100 images are enough to get stable weights which do not depend a lot on the images we add.

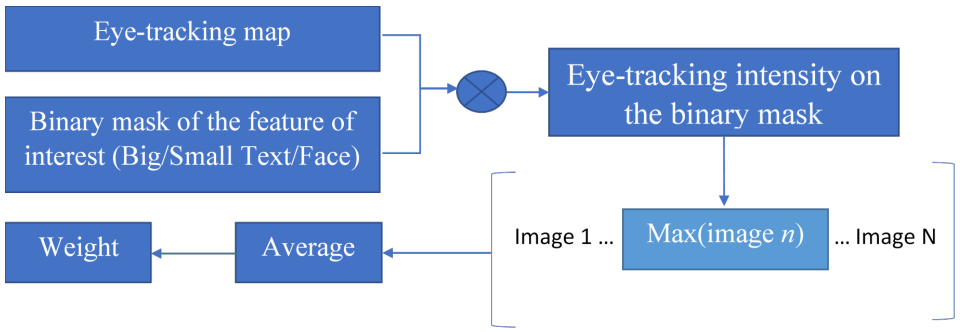


Figure 3.16. Object binary mask is used along with the eye-tracking map to extract the maximum eye-tracking value for the object. This value, averaged on all the images will provide a weight for a given object.

3.2.4.2 Top-down and Bottom-up Fusion

Once the weight was computed for one of the objects among small text (ST), big text (BT), small faces (SF), and big faces (BF), the question is how to make a fusion between this information and the bottom-up saliency map.

First, as described in Fig. 3.17, for each feature, we split the small and big masks and then smooth them in order to obtain and image close to the fuzzy bottom-up saliency map.

We made linear combinations between bottom-up information (saliency maps) and top-down information (text and face detection). We generated results of saliency maps using RARE [55] and **1)** text detection using both [48] and the masks as described in section 3.2.3.1 and **2)** face detection using both [31] and the masks as described in section 3.2.3.2.

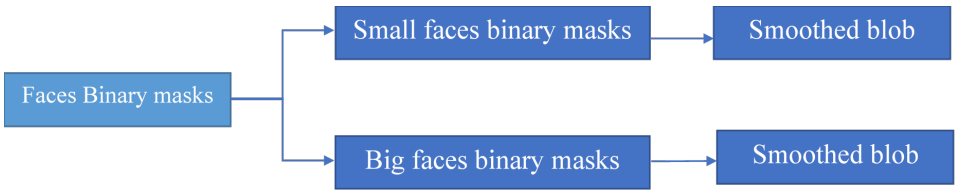


Figure 3.17. For face and text, the binary masks are split between big and small masks and then low-pass filtered to provide a smoothed result before being fused with the bottom-up saliency map.

To make a simple fusion between bottom-up saliency maps (SM) and top-down information (face alone, text alone or both), we used linear combinations which are easy to implement. The weights were either the same for text (big and small) and face (big and small), either different by using the results that we obtained in previous section which are given the following formulas:

$$(SM + ATF)/2 \tag{3.4}$$

$$(SM + BTF)/2 \tag{3.5}$$

$$(SM + STF)/2 \tag{3.6}$$

$$SM + (wSTF/wBTF) * STF + BTF \tag{3.7}$$

where SM is bottom-up saliency maps computed from RARE [55], ATF is either all text (big and small), either all face (big and small), either all text and face depending on the experiment, BTF is either big text, either big face, either all big text and big face depending on the experiment, STF is either small text, either small face, either all small text and small face depending on the experiment. $wSTF$ and $wBTF$ use the weights found in section 3.2.4.1 for either small text or small face ($wSTF$) and for either big text or big face ($wBTF$).

To also test the impact of the detector accuracy, our experiment is divided into three different parts: perfect detector, good detector, and bad detector. For good and bad detector, we use the state-of-the-art detector [31] and [48], respectively. While for the perfect detectors, we use the our masks which are generated from the OSIE dataset [69] as shown in Fig. 3.17 for both face and text.

3.2.5 Results

3.2.5.1 Top-down Feature Weights

To get ideal weights for big and small text and face, we used the method described in Fig. 3.16. As a result, we obtained a weight of big text (wBT) = 0.7871, of small text (wST) = 0.3221, of big face (wBF) = 0.8159, and of small face (wSF) = 0.6457. We can see that the

difference between big text and small text is more important than big face and small face. Big face is a little more important than big text, but the difference is not very significant.

3.2.5.2 Perfect Detector

For the perfect detector, experiment is conducted into three parts. In the first part, we combine the bottom-up saliency map (SM) with text feature alone, then second part with face feature alone, and finally with both text and face features.

We use several metrics to evaluate the bottom-up attention model object-based top-down attention by making correlation between some different results (from section 3.2.4) and the eye-tracking Fixation Maps (FM). To investigate on the role of different top-down information, RARE [55] is used as bottom-up model to generate saliency maps (SM). The same saliency metrics as in the MIT300 dataset evaluation [8] were used. For those metrics, there are Correlation Coefficient (CC), Kullback-Leibler Divergence (KLD), Normalized Scanpath Saliency (NSS), Similarity (SIM), and Area Under the ROC curve from Judd (AUCJ). Those metrics provide some complementarity and are well described in [54]. The smallest values represent the best results in KLD metric. For the other metrics, higher values are the best.

The results are summarized given the three different experiments in Table 3.1 (SM with text alone), Table 3.2 (SM with face alone), and Table 3.3 (SM with both text and face). The first line corresponds to the comparison between the bottom-up saliency map (SM) and the eye tracking fixations map (FM). The second line corresponds with the comparison of all features (all text (AT) in Table 3.1, all faces (AF) in Table 3.2 and all text and face (ATF) in Table 3.3). The third line is a comparison between FM and big text (BT) in Table 3.1, FM and big face (BF) in Table 3.2, and between FM and big text and face (BTF) in Table 3.3. The fourth line in Table 3.1 and Table 3.2 represent the comparison between FM and small text (ST) and FM and small faces (SF), respectively. The final line is the comparison of the weighted fusion for big and small text (wBST) in Table 3.1, for big and small face (wBSF) in Table 3.2 and all big and small text and face (wTF) in Table 3.3. A first global remark is that all metrics are very coherent, and they provide almost the same relative rank for all measures.

Correlation	**Metrics**				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
SM, FM	0.4683	1.0591	1.5364	0.4364	0.8365
AT, FM	0.5042	1.0140	1.7013	0.4514	0.8452
BT, FM	0.5058	1.0151	1.7008	0.4504	0.8444
ST, FM	0.4666	1.0587	1.5420	0.4378	0.8372
wBST, FM	**0.5061**	**1.0127**	**1.7081**	**0.4517**	**0.8454**

Table 3.1. Result between bottom-up saliency map and text detection. The smallest values represent the best results in KLD metric. For the other metrics, higher values are the best.

In Table 3.1, it indicates that adding information about small text brings nothing to the result because result of small text is a little less good than the bottom-up saliency map alone in some metrics. On the other hand, adding big text provides an important and improvement result in the CC metric compared to result when adding both small and big text (AT). For all metrics, the use of the weights provides the best results of all.

Correlation	**Metrics**				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
SM, FM	0.4683	1.0591	1.5364	0.4364	0.8365
AF, FM	0.5352	0.9790	1.8334	0.4581	0.8494
BF, FM	0.5277	0.9876	1.7890	0.4553	0.8478
SF, FM	0.4762	1.0514	1.5851	0.4396	0.8379
wBSF, FM	**0.5354**	**0.9786**	**1.8348**	**0.4582**	**0.8494**

Table 3.2. Result between bottom-up saliency map and face detection.

In Table 3.2, it indicates that adding information about small face brings this time a small improvement to the bottom-up saliency map alone and is never negative. However, adding big face provides an important result improvement. Moreover, adding both big and small face also brings improvement. The best case, for all metrics, the use of the weights provides the better results of all.

In Table 3.3, we do not compute the result for small text and face since it is always smaller than big text and face. We just keep here the best combinations: ATF for all text and face, BTF for only big face and text, and the weighted text and face (wTF). While ATF is always a little better than BTF, the weighted version is even better.

Correlation	**Metrics**				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
SM, FM	0.4683	1.0591	1.5364	0.4364	0.8365
ATF, FM	0.5687	0.9287	1.9787	0.4719	0.8595
BTF, FM	0.5632	0.9395	1.9382	0.4684	0.8567
wTF, FM	**0.5691**	**0.9273**	**1.9824**	**0.4723**	**0.8597**

Table 3.3. Result between bottom-up saliency map and text and face detection.

3.2.5.3 Imperfect Detector

Here we decide to use two state-of-the-art detectors which imply some misdetections or false detections (especially for the text detector). We don't combine result for both text [48] and

face detection since the text-related results are very bad (see Table 3.4). However, the results from face detection are good because facial landmarks approach [31] can detect the frontal face well although it misses some faces in a scene. For faces, we use a state-of-the-art face detector based on the dlib library [63]. We use the classical Histograms of Oriented Gradients (HOG) feature followed by a classifier which has a good detection rate for faces and was introduced by [15].

Correlation	**Metrics**				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
SM, FM	**0.4683**	**1.0591**	**1.5364**	**0.4364**	**0.8365**
AT, FM	0.4092	1.1523	1.3416	0.4193	0.8132
BT, FM	0.4071	1.1555	1.3306	0.4184	0.8130
ST, FM	0.4620	1.0641	1.5226	0.4364	0.8347
wBST, FM	0.4101	1.1513	1.3445	0.4196	0.8134

Table 3.4. Result between bottom-up saliency map and text detection (bad text detector).

Table 3.4 shows that the misdetections and even more the false detections of the text detector seriously decrease the results compared to the bottom-up saliency map alone. This text detector is not good enough to be used to add top-down information. For all metrics, results of the saliency map alone are better than all text, big text, and weighted.

Correlation	**Metrics**				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
SM, FM	0.4683	1.0591	1.5364	0.4364	0.8365
AF, FM	**0.5264**	**0.9968**	**1.8122**	**0.4523**	**0.8471**
BF, FM	0.5180	1.0047	1.7508	0.4504	0.8455
SF, FM	0.4763	1.0518	1.6003	0.4381	0.8381
wBSF, FM	0.5258	0.9975	1.8126	0.4520	0.8471

Table 3.5. Result between bottom-up saliency map and face detection (good face detector).

Table 3.5 shows that the face detector, which has a better quality, can provide good improvement of the bottom-up saliency map. However, the difference between a simple average fusion (AF) and the weighted version (wBSF) is not significant. For some metrics such as CC and KLD, it is even better to just make the global average instead of using the face weights.

3.2.6 Discussion

We tested with the OSIE dataset [69]. We tested several object detectors (bad, good, and perfect), and we demonstrated that if the detector is not good enough, it is better not to use it at all and only use the bottom-up information (see Table 3.4). If the detector is good, a simple average can be almost as good as a more complex weighted average (Table 3.5). When the detector becomes very good, then the weighted average really makes sense (see Table 3.1, Table 3.2, and Table 3.3). This is even more the case when several top-down features are mixed to bottom-up and some might be more important than others.

The size of the top-down object is very important. This was more the case with text where the difference in terms of eye fixations between big text (titles) and small text (description) is very important. Indeed, people provide attention to text because of its cognitive content. While for titles, the cognitive load needed is very small, for blocks of smaller text, they will less attract attention, especially if the beginning of the text (the most attended) has no important information. There is still a difference between big face and small face, but this difference is smaller. If we just consider big text and big face, the weights values of them are almost the same. It is an interesting result as previous results [69] do not consider the difference between big and small text or big and small face. The result was polluted by small text which really decrease a lot the overall text importance.

An important result improvement can be obtained by using classical bottom-up attention models to which we can add easily the higher level detected objects. Resulting models will approach novel DNN-based attention approaches while they keep generality. They are also well responding to bottom-up features which are underestimated by DNNs [39]. In addition to that, classical models can have a behavior which can be explained while DNNs provide results without letting any chance to the programmers to explain why exactly their model works well or not. Being able to explain the reaction of an algorithm might be critical especially for security applications. That is why, for our future work, we will go deeper in the object-based top-down features which can be extracted and the optimal mix with bottom-up saliency maps.

3.3 Bottom-up Attention Maps with Object Detection

3.3.1 Objective

The objective of this experiment is to investigate the importance of bottom-up versus top-down attention. First, we enrich with top-down information classical bottom-up models of attention. Then, the results are compared with DNN-based models. Our provocative question is: “do deep-learning saliency models really predict saliency or they simply detect interesting objects?”. We found that if DNN saliency models very accurately detect top-down features, they neglect a lot of bottom-up information which is surprising and rare, thus by definition difficult to learn.

3.3.2 Method

Several fusion algorithms are made (see details in section 3.3.4) which can mix the bottom-up saliency maps and top-down information. All of top-down information is derived from deep-learning methods. For dataset to be used in this step, there are two datasets such as the OSIE dataset [69] and the MIT300 dataset [7]. For the OSIE dataset usage, we can test and get results from our side. However, for the MIT300 dataset usage, we can also test on our side, and then after getting the results of saliency maps, we have to send those result to MIT evaluation team in order to get our final results of attention. Finally, we can compare our proposed method to other methods both classical and deep-learning. Workflow of this method is seen in Fig. 3.18. For this part, object detection is added more from the previous experiment. The presenting of objects in a scene image is very important if there is no human face or text. Moreover, the centered Gaussian is also applied in this part because it plays an important role for natural images. The OSIE and MIT300 datasets contain mainly natural images, so a centered Gaussian function is the best choice.

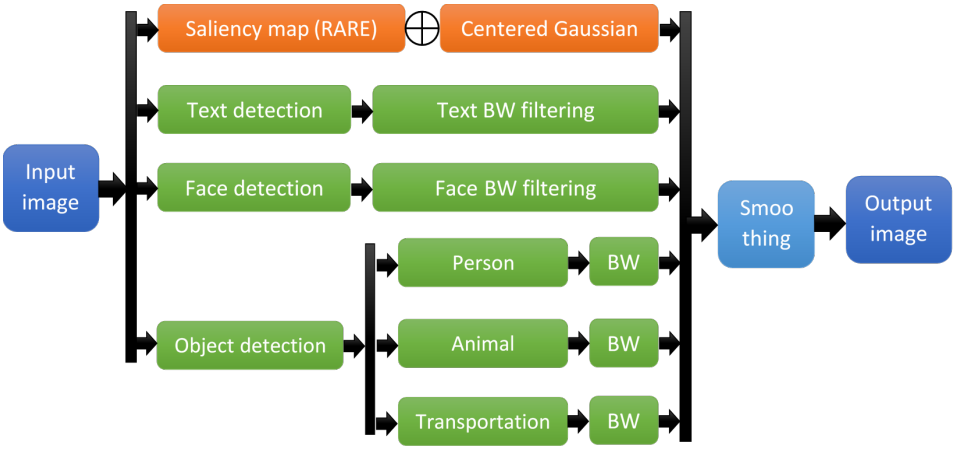


Figure 3.18. Workflow of our proposed method that combines saliency maps with text, face, and object detection.

3.3.3 Bottom-up and Top-down Information

Traditional bottom-up models use image features (i.e., luminance, chrominance, texture) to detect locally contrasted or globally rare regions, more details in section 2.2. This research work proposes a naive yet generic top-down information framework that can be added to any bottom-up saliency model. In [34], the authors demonstrated that an object detector could bring remarkable improvement result to saliency maps on condition that such detector is **1)** "good enough" (especially with few false positives) and it is **2)** "specific enough" (general-purpose object detectors may include objects less likely to attract visual attention [10]).

Here, we use a bunch of existing detectors for face, text, person, animal, and transportation detection. The current Deep Neural Network (DNN)-based object detectors have become very good and, based on [34], we hypothesize that the use of this set of good detectors bringing specific top-down information will improve the results of overall saliency maps. All those

detectors are used on all the images to keep our method generic and the final results include issues due to false positives or false negatives.

To get quantitative results, two different datasets are used. The first one is the OSIE dataset [69] containing more than 700 images along with the eye-tracking and object segmentation. This dataset is also used for our generic top-down framework parameters tuning. The second dataset is the MIT300 dataset [7] containing 300 images, which is used for comparing our proposed model with various state-of-the-art visual attention models.

3.3.3.1 Face Detection

The face detection algorithms available in [63] were used in our study, more details in section 2.3.1. The first algorithm uses the Histogram of Oriented Gradients (HOG) features combined with a linear classifier (SVM), while the second one uses a Convolutional Neural Network (CNN). The CNN-based face detector outperforms the HOG-based detector on the OSIE dataset especially on the badly exposed faces (see Fig. 3.19).

Figure 3.19. Comparison results between HOG and CNN-based face detectors. (first image) Input image, (second image) result of HOG-based face detector, and (last image) result of CNN-based face detector.

3.3.3.2 Text Detection

Connectionist Text Proposal Network (CTPN) [66] is used as text detector in our framework, more details in section 2.3.2. The CTPN detects a text line in a sequence of fine-scale text proposals directly in convolutional feature maps. The sequential text proposals are connected by a recurrent neural network, outcoming in an end-to-end trainable model. Moreover, CTPN works reliably on multi-scale and multi-language texts without any additional post-processing step (see Fig. 3.20).

3.3.3.3 Object Detection

A state-of-the-art real-time object detection from [51] were used, which could detect over 9000 objects in a reliable way, , more details in section 2.3.3. Many different detection classes are available, but only three categories (i.e., person, animal and transportation) were selected and used in our experiments. As a result, these classifiers provide three different maps with binary masks for persons, animals and transportation (see Fig. 3.21).

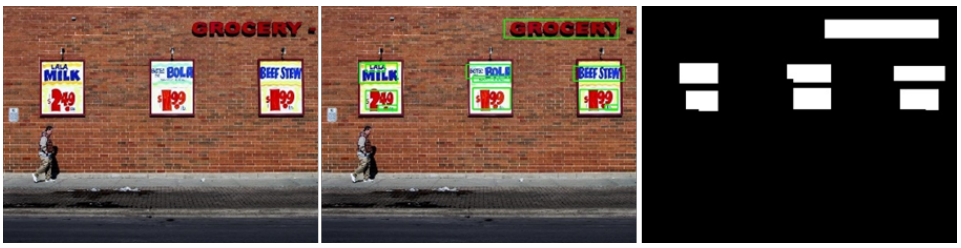


Figure 3.20. Result of text detection. (first image) Input image, (second image) text detection (green bounding-boxes), and (last image) binary text masks.

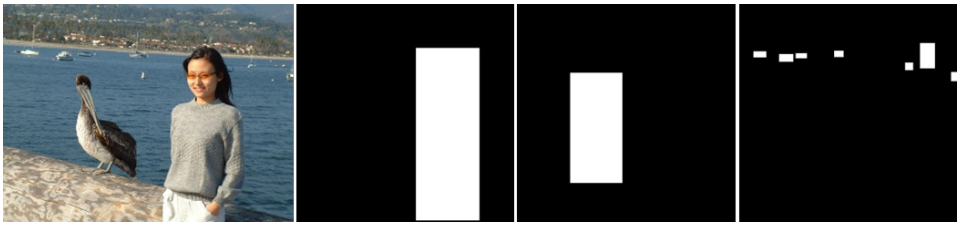


Figure 3.21. Result of object detection. (first image) Input image, (second image) person detection, (third image) animal detection, and (last image) transportation detection (here the small boats in the back are detected).

3.3.3.4 Context-based Top-down Information

Besides the three detection models mentioned above, a centered Gaussian function is also added into the image because it plays an important role for natural images. The image context (Gist) is immediately detected by the viewer [49]. In [42], the author showed the difference between natural image context (where the eye gaze focuses in the center), a website context (where it is attracted more towards the top-left corner and an advertising context (where the behavior is in between the previous two). The OSIE and MIT300 datasets contain mainly natural images, so a centered Gaussian function is the best choice.

3.3.4 Mixing Bottom-up and Top-down Information

A straightforward combination of the bottom-up model, the centered Gaussian function and the different detectors for face, text, person, animal, and transportation were implemented. The binary masks which are the outputs of all the detectors are smoothed to better fit into the saliency map (SM). The combination begins with the centered Gaussian (Eq. 3.8). Then, the object detectors are added (Eq. 3.9, Eq. 3.10, Eq. 3.11) and mixed together in Eq. 3.12. Finally, the faces and text, which have more impact are added only at the end (Eq. 3.13). These equations are as follows:

$$CSM = (a * SM * CG^b) + (1 - a) * SM \tag{3.8}$$

$$CTSM = (Tra * CSM) + CSM \tag{3.9}$$

$$CASM = (Ani * CSM) + CSM \tag{3.10}$$

$$CPSM = (Per * CSM) + CSM \tag{3.11}$$

$$COSM = (CTSM + CASM + CPSM)/3 \tag{3.12}$$

$$FAPTTX = (COSM + F + w * T)/3 \tag{3.13}$$

where a, b are two parameters (found to be a=0.75 and b=4), SM is the bottom-up saliency map, CG is the centered Gaussian image, Tra, Ani, Per, F, T are the smoothed masks of transportation, animal, person, face, and text detection, respectively. w is a weight set to 0.6. The a, b, and w parameters were found as to optimize the results on the OSIE dataset. At the end, the final saliency map is optimized by blurring as stated in [53].

3.3.5 Top-down versus Bottom-up Influence

We use the same metrics as in section 3.2.5.2 to evaluate our results. Table 3.6 shows, on the OSIE dataset, the metric values between the eye-tracking fixation maps and the model output. This output is 1) bottom-up saliency maps (SM) alone on the first line, 2) SM with face detection (F) on the second line, 3) SM with text detection (TX) on the fourth line, 4) SM with animal detection (Ani) on the sixth line, 5) SM with person detection (Per) on the eights line, 6) SM with transportation detection (Tra) on the tenth line and 7) SM with centered Gaussian (CG) on the twelfth line. Results are computed on subsets of images: 279 images with faces, 425 images with text, 138 images with animals, 484 images with persons, 98 images with transportation and 700 images with centered Gaussian. In Table 3.6, the results in terms of CC metric shows that the faces influence is definitely higher than SM with a 0.15 (0.5631 - 0.4179 = 0.15) improvement measured on the 279 images in each of which having at least one face. Besides, it is quite interesting to see the result given by text detection TX (with 0.5478 - 0.4637 = 0.08 difference). The centered Gaussian and the animals' detection come after with a 0.04 difference. Strangely, person detection is less useful than animal detection with only 0.01 of difference on the CC metric. This is probably because the eye gaze will only focus on small parts of the body while the face has already been taken by the face detector. Finally, the transportation detector has a negative effect on the result with 0.02 differences. This shows that the objects like cars, buses, bikes, and so on are not really attended or only on very specific parts of these objects. The results for the other metrics are similar to the CC metric. The result with **bold-fonts** represents the best result in comparison.

To check how general our framework is, we test it on four different bottom-up saliency models such as AIM [4], AWS [19], GBVS [22], and RARE [55]. Table 3.7 shows, for each model, the results of the bottom-up saliency map alone (SM) and the SM added with our framework (**FAPTTX**). RARE model has the best results but they are very close to AWS. GBVS and AIM are less good. One can see that the best improvement is achieved for AIM which for example gains about 0.22 in CC metric and seems to be the one capturing the less top-down attention. AWS and RARE both improve at 0.16 (CC metric). Finally, the GBVS

Maps (images)	Metrics				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
SM (279)	0.4179	1.1548	1.4118	0.4115	0.8291
F (279)	**0.5631**	**0.9390**	**1.8914**	**0.5165**	**0.8525**
SM (425)	0.4637	1.0492	1.4626	0.4390	0.8311
TX (425)	**0.5478**	**0.9011**	**1.7870**	**0.4995**	**0.8544**
SM (138)	0.4754	1.1183	1.7178	0.4202	0.8516
Ani (138)	**0.5111**	**1.0425**	**1.8565**	**0.4716**	**0.8629**
SM (484)	0.4587	1.0971	1.5700	0.4262	0.8412
Per (484)	**0.4699**	**1.0594**	**1.6185**	**0.4626**	**0.8433**
SM (98)	**0.5152**	**0.9998**	**1.8336**	**0.4471**	**0.8636**
Tra (98)	0.4902	1.0135	1.7608	0.4748	0.8579
SM (all)	0.4683	1.0597	1.5364	0.4364	0.8365
CG (all)	**0.5001**	**0.9738**	**1.6231**	**0.4679**	**0.8472**

Table 3.6. Results using RARE model (OSIE dataset) on the number of images (on a total of 700) where at least an object is detected.

only improves about 0.13 (CC metric) probably because it already has the centered Gaussian included in the bottom-up model.

Models	Maps	Metrics				
		CC	**KLD**	**NSS**	**SIM**	**AUCJ**
AIM [4]	SM	0.3251	1.5241	1.0717	0.3454	0.7733
	FAPTTX	0.5392	1.1186	1.7311	0.4070	0.8496
AWS [19]	SM	0.4583	1.1171	1.4855	0.4268	0.8219
	FAPTTX	0.6161	0.8313	2.0290	0.4995	0.8708
GBVS [22]	SM	0.4380	1.0880	1.3496	0.4250	0.8159
	FAPTTX	0.5608	0.9379	1.8104	0.4828	0.8488
RARE [55]	SM	0.4683	1.0597	1.5364	0.4364	0.8365
	FAPTTX	**0.6235**	**0.8162**	**2.0868**	**0.5192**	**0.8719**

Table 3.7. Correlation result using several models (OSIE dataset).

3.3.6 DNN-Based versus Bottom-up Models

The new DNN-based attention models occupy the first places on benchmarks such as the one of MIT300. In this part, we check what happens when we come up with a bottom-up model augmented with our top-down framework.

3.3.6.1 Qualitative Comparison

For the DNN-based model, we use SAM-ResNet [13] and Salicon [25] which are two state-of-the-art DNN-based models. We notice that they provide better results than our proposed approach especially if the scene contains humans (in Fig. 3.22). However, they provide poorer results than RARE model if the scene is complex with unknown objects (in Fig. 3.23).

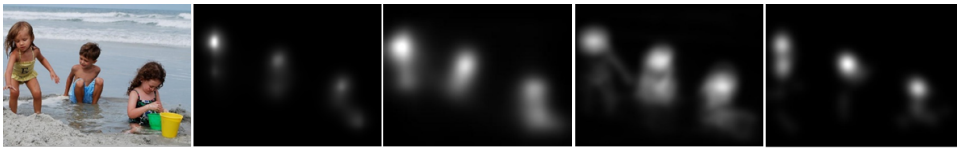


Figure 3.22. Results where DNN-based models are better than bottom-up models. (first) Input image, (second) Result of SAM-ResNet, (third) Result of Salicon, (fourth) Result of our model and (last) Eye tracking map.

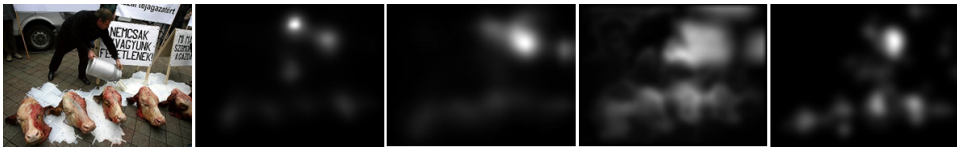


Figure 3.23. Results where DNN-based models are less good than bottom-up models. (first) Input image, (second) Result of SAM-ResNet, (third) Result of Salicon, (fourth) Result of our model and (last) Eye tracking map.

3.3.6.2 Quantitative Comparison

The OSIE dataset: While the DNN-based models are overall better than the proposed model, we looked more in details on the images. Our experiment shows that on the OSIE dataset RARE bottom-up model alone is better than SAM-ResNet for 5.7% of the images and RARE augmented with our generic framework is better than SAM-ResNet on 14.3% of the images. If we only take the images where our model has a CC metric higher than 0.05 compared to SAM-ResNet (which will boost the model of about 10 places on a benchmark such as MIT300), our approach is still much better than SAM-ResNet on 9% of the images. We used here the CC metric as it is one which is not favorable to our approach (see Table 3.10 showing that KLD is the best metric for our model). It appears that DNN-based models might be inferior in learning bottom-up data while they are better in detecting objects usually salient (top-down information). Moreover, we compare our method with SAM-ResNet method

by using the OSIE dataset in Table 3.8. However, the SAM-ResNet result is better than ours because in the SAM-ResNet method, they have already added top-down information.

Models	Metrics				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
SAM alone	**0.7713**	**1.3726**	**3.1023**	**0.6510**	**0.9026**
SAM+FAPTTX	0.7546	1.6081	2.8205	0.6226	0.8943
Ours (FAPTTX)	0.6235	1.8162	2.0868	0.5192	0.8719

Table 3.8. Comparing result between SAM-ResNet and ours by using the OSIE dataset.

The MIT300 dataset: According to MIT300 benchmark [7], our model has the best results compared to all bottom-up models in Table 3.9. It is still surpassed by some DNN-based models in Table 3.10, but a lot of those models are now less good than ours. This shows that a bottom-up model, simply augmented with some top-down information can be better than all the other bottom-up models and even better than number of other DNN-based models depending on the metrics. For example, for the KLD metric by this date, our model is better (lower KLD value) than about 18 DNN-based models while less good than only 7 DNN-based models. More details on all DNN-based models in Table 3.10 can be found in the MIT300 benchmark [7].

Models	Metrics				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
Ours	**0.6166**	**0.7179**	**1.6762**	**0.5472**	**0.8388**
BMS [72]	0.55	0.81	1.41	0.51	0.83
OS [11]	0.54	0.84	1.41	0.51	0.82
GBVS [22]	0.48	0.87	1.24	0.48	0.81

Table 3.9. Comparing result between bottom-up models and ours by using the MIT300 dataset.

3.3.7 Discussion

The influence of person and transportation detection is marginal or even negative, because the viewer gazes probably focus on small parts of persons or cars but not everywhere on their bounding boxes. Concerning bottom-up information, we showed that for almost 6% of the images in the OSIE dataset, a bottom-up model alone can better predict the gaze than DNN-based models. This means that the bottom-up information still remains important and should not be neglected in visual attention, especially in the complex and crowded images where it is hard to identify faces.

Models	Metrics				
	CC	**KLD**	**NSS**	**SIM**	**AUCJ**
DSCLRCN	0.8	0.95	2.35	0.68	0.87
SALICON	0.74	0.54	2.12	0.6	0.87
SAM-ResNet	0.78	1.27	2.34	0.68	0.87
Ours	**0.6166**	**0.7179**	**1.6762**	**0.5472**	**0.8388**
SalNet	0.58	0.81	1.51	0.52	0.83
eDN	0.45	1.14	1.14	0.41	0.82
GoogLeNet	0.49	0.99	1.26	0.45	0.81
JuntingNet	0.54	0.96	1.43	0.46	0.80

Table 3.10. Comparing result between DNN-based models and ours by using the MIT300 dataset.

Finally, we showed that mixing a bottom-up model with our naive top-down information framework leads on the MIT300 saliency benchmark to the best results among all bottom-up models and overtakes number of DNN-based models especially on KLD which measures the probability distribution resemblance with eye-tracking. Considering the fact that DNN-based models results cannot be explained and that they seem to neglect bottom-up information, future work is to see how a DNN-based model can be mixed with bottom-up models to consider both the good top-down detection of DNN-based models and the necessary bottom-up attention from traditional models.

3.4 Building a New Retail Dataset

3.4.1 Objective

This data collection is our newly assigned task which allows to generate our own dataset/database and new experimental results. To achieve this task, we conducted some data collection by taking 200 photos from various supermarkets (retail images) located in both Cambodia and Belgium. We captured those photos from different locations or countries because we would like to:

- Extract different information and memory from the collected photos
- Make the eye-tracking data or maps (ground truth fixation maps)
- Conduct new experiments using this newly collected dataset

It was not easy to collect such dataset because a few different process were needed before and after capturing the photos. First, we had to prepare the camera with the same settings and then go to visit several supermarkets in Phnom Penh, Cambodia as well as Mons, Belgium. We observed the available products in each market and then searched for the scenes we wanted to take the photos for the research. It took us quite long for this process because we had to check very carefully the quality of all the captured photos in order to avoid getting the

redundant, blurred or low-quality photos/images. For resolution of each image, we decide to determine as 2048 in width by 1152 in height. This resolution is quite good for our training and testing because it has enough information from each image. It means that it is not too small or big. Finally, we choose only 100 good quality of images from each country. We do not take images which are blur or redundant. Two examples of retail images captured at a supermarket in Phnom Penh are depicted in Fig. 3.24 and 3.25. Another example, as seen in Fig. 3.26 and 3.27, two retail images token at a supermarket located in Mons are displayed.

Figure 3.24. General ingredient products in a supermarket in Phnom Penh, Cambodia.

Figure 3.25. General skin care products in a supermarket in Phnom Penh, Cambodia.

3.4.2 Method

After the data collection from supermarkets, the newly collected dataset are composed of few hundreds photos, which is then decomposed into two different datasets such as training and testing datasets. Each dataset contains 100 images in total, which comprise of 50 retail

Figure 3.26. General meat products sold in supermarket in Mons, Belgium.

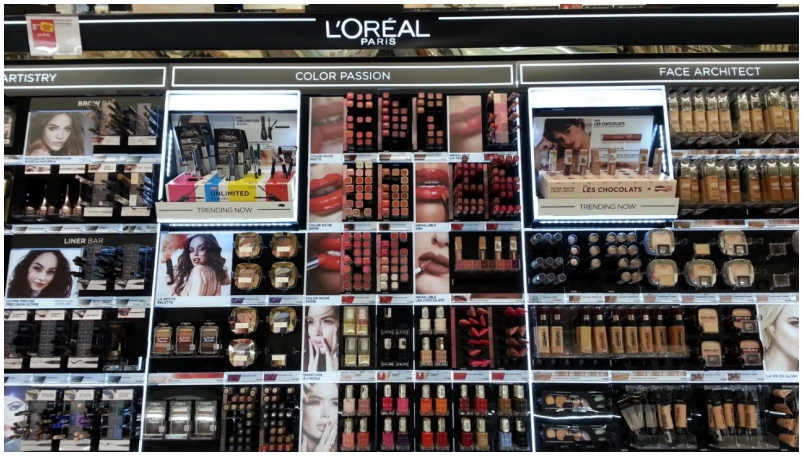
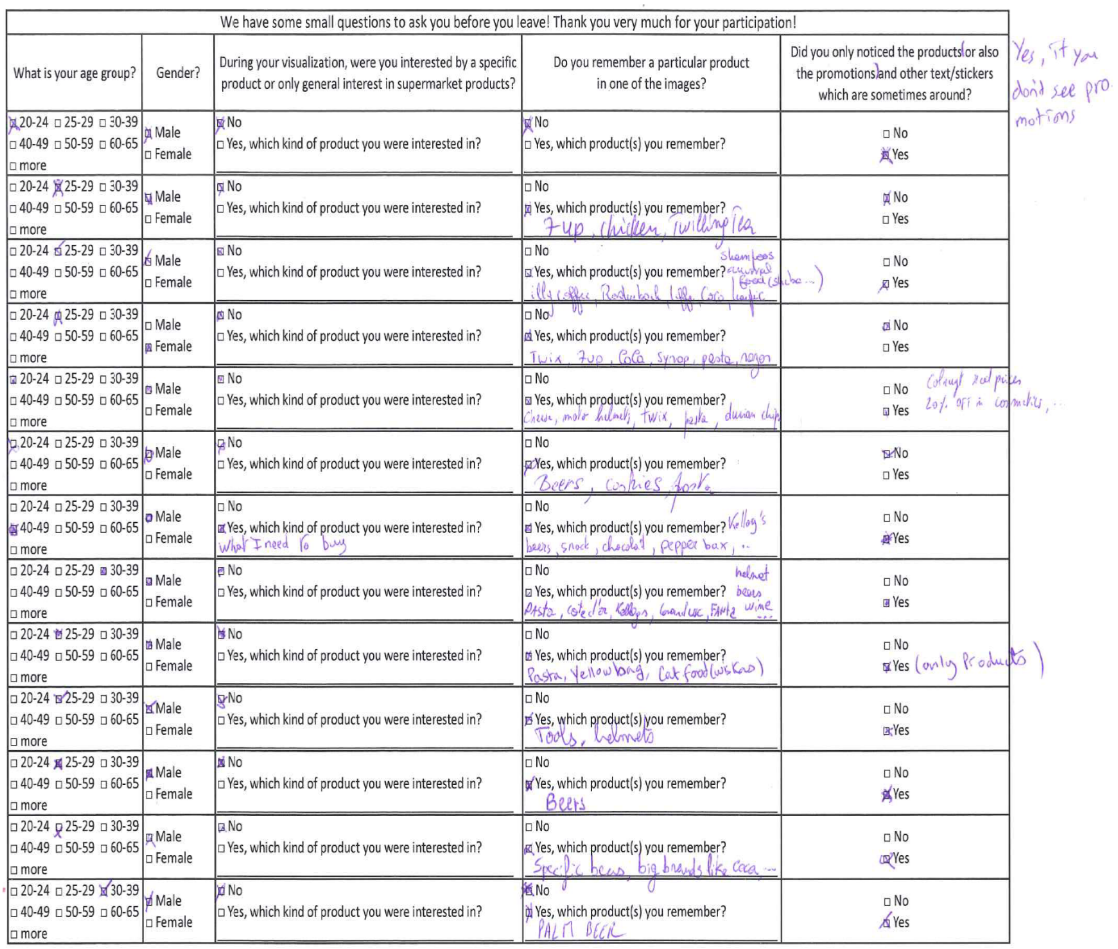


Figure 3.27. General cosmetic products sold in supermarket in Mons, Belgium.

images taken in Cambodia and another 50 images taken in Belgium. The photos are merged like this in order to make ease for the viewers who see these images during doing eye-tracker experiment. Then, we find some people to conduct our new eye-tracker experiments using those datasets. To set up eye-tracker experiments and obtain this data, we consider on several steps such as:

- Create 5 questions to know about the viewers' memory and attention (see Fig. 3.28)
- Make 2 projects on Gazepoint software (Gazepoint Analysis UX Edition v4.3.0)
- Each project contains 200 images: 100 retail images (display 5 seconds) and 100 blank images (display 2 seconds)
- Display images in sequence, one retail image (see Fig. 3.29) and one blank images
- Calibrate each viewer with display screen for getting accurate viewing pixels in image
- Get a green point: good calibration, otherwise it is bad calibration (see Fig. 3.30)

- Maintain viewers' head position and just move their eyes after calibrating
- Take distance between eye-tracker and viewer about 0.5 meters in horizontal
- Involve 14 viewers per project, analyze the projects, and combine these two projects into a dataset



We have some small questions to ask you before you leave! Thank you very much for your participation!

What is your age group?	Gender?	During your visualization, were you interested by a specific product or only general interest in supermarket products?	Do you remember a particular product in one of the images?	Did you only noticed the products or also the promotions and other text/stickers which are sometimes around?
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember?	□ No □ Yes (Yes, if you don't see promotions)
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? 7up, chicken, Twilling Tea	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? shampoos, animal food (...) illy coffee, [illegible]	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? Twix, 7up, CoCa, Syrop, pesto, [illegible]	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? Cheese, moto helmet, Twix, pasta, durian chips	□ No □ Yes (Colruyt red prices 20% off in cosmetics, ...)
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? Beers, cookies, pasta	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in? what I need to buy	□ No □ Yes, which product(s) you remember? Kellog's beers, snack, chocolat, pepper box, ..	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? helmet beers Pasta, cote d'or, Kellogs, [illegible], FANTA wine ...	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? Pasta, Yellow bag, Cat food (wiskas)	□ No □ Yes (only Products)
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? Tools, helmets	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? Beers	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? Specific beers, big brands like coca ...	□ No □ Yes
□ 20-24 □ 25-29 □ 30-39 □ 40-49 □ 50-59 □ 60-65 □ more	□ Male □ Female	□ No □ Yes, which kind of product you were interested in?	□ No □ Yes, which product(s) you remember? PALM BEER	□ No □ Yes

Figure 3.28. Creation 5 questions for viewers after the eye-tracker experiments.

We add bland images in order to refresh the viewers' eyes and reset last view position from each image. So, in total there are 200 images per dataset, and it took about 12 minutes per person. Finally, we get all fixation pixels from all viewers.

When we have enough data, we start to do eye-tracker experiment. We create two projects to get viewers' fixation points. We request some people in the ISIA Laboratory and some students from department of translation. We decide to take 14 people for each project and have 5 questions after finish doing each experiment. After we finish doing eye-tracker experiment, we start to analyze these projects. We make a group of 14 viewers for each image in order to get all fixation (viewing) points. Moreover, Gazepoint software can generate several values, but we choose only three values for our experiment such as image id, fixation point of gaze x, and y axis (see Fig. 3.31).

Figure 3.29. Displaying one retail image during the eye-tracker experiments.

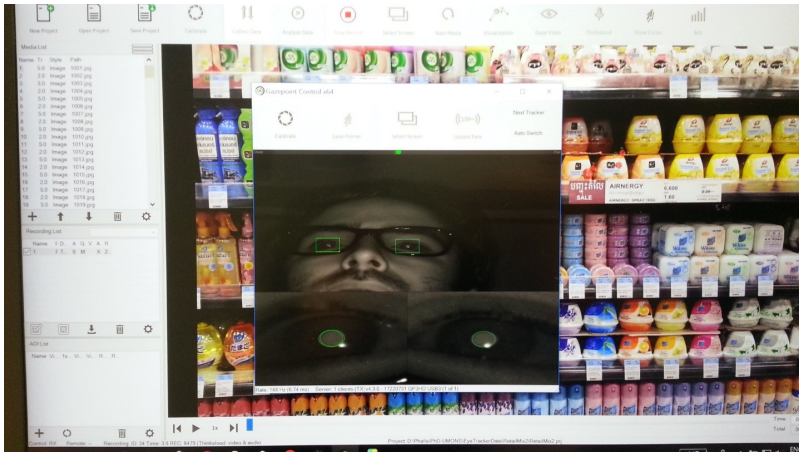


Figure 3.30. Obtaining good calibration on eye-tracker experiment.

This software is also able to generate a heat map by grouping 14 viewers of each image (see Fig. 3.32). The red color means that it has highest viewing or attention from each image. Then it follows by the yellow, green, and blue color.

Although we get data from the eye-tracker experiment, we cannot directly use it as ground truth density and fixation map. It is because of the software is not stable to generate the correct fixation points for all images. From fifth image, the software will generate the fixation at the center of image, and it is not correct compared to real viewers' seeing in images. Thus, firstly we must transform this data into visible images as fixation and saliency maps. Then we compare manually them to heat maps from eye-tracker. We must make sure that our ground truth generation is the same as viewer points. From each image, we generate two different images such as fixation image (see Fig. 3.33) and ground truth image (or saliency map) (see Fig. 3.34). We use the MATLAB programming to generate those results. While we have all 200 ground truth density and fixation maps, we start to retrain one of famous deep learning

	A	B	C
1	MEDIA_ID	FPOGX	FPOGY
2	0	0.26047	0.54802
3	0	0.28475	0.53502
4	0	0.34275	0.60639
5	0	0.57618	0.60308
6	0	0.10266	0.90216
7	0	0.14469	0.88689
8	0	0.43426	0.83691
9	0	0.52154	0.80333
10	0	0.75515	0.81867
11	0	0.83621	0.85087
12	0	0.62473	0.83163
13	0	0.58657	0.88586
14	0	0.49609	0.88211
15	0	0.42816	0.77315
16	0	0.47194	0.72721
17	1	0.48593	0.71816
18	1	0.4842	0.7208
19	1	0.43035	0.63054
20	1	0.65294	0.56547
21	1	0.42118	0.55746

Figure 3.31. Fixation points from viewers who do eye-tracker experiment.

model which is the SAM-ResNet model [13]. For this model, the last layer requires the ground truth density and fixation map. That's why we need to generate those maps.

3.4.3 Results

3.4.3.1 Retrain SAM-ResNet Network

To maintain the existing weight from this network, we firstly load the weight. Then we add new weight from our new dataset (200 images) as bottom-up information. This network focuses mainly on top-down information because most of current deep learning networks consider only top-down information, i.e. face detection. However, they neglect on bottom-up information which can be improved the attention.

According to this network, there is an existing weight of Salicon 2015 dataset. In addition, these days there is a new weight of Salicon 2017 dataset. Then we decide to retrain SAM-ResNet network into two new different weights by keeping the old weight and adding new information. In addition, we also train on another dataset, CAT2000.

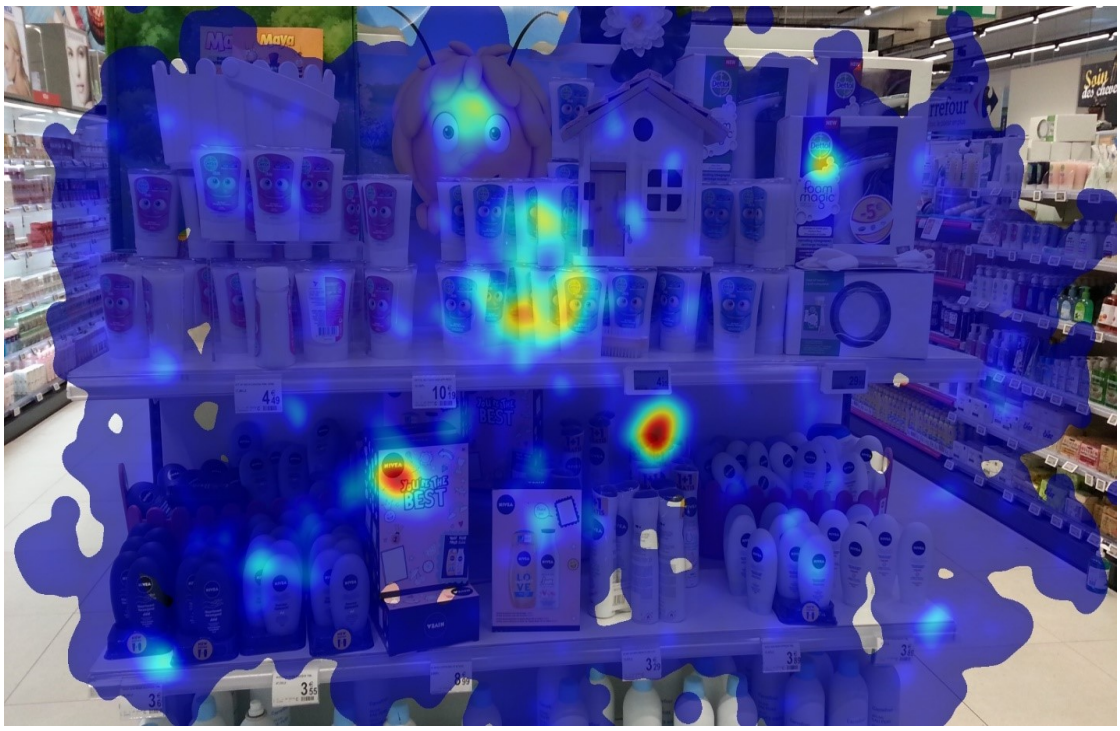


Figure 3.32. An example of heatmap image by grouping 14 viewers.

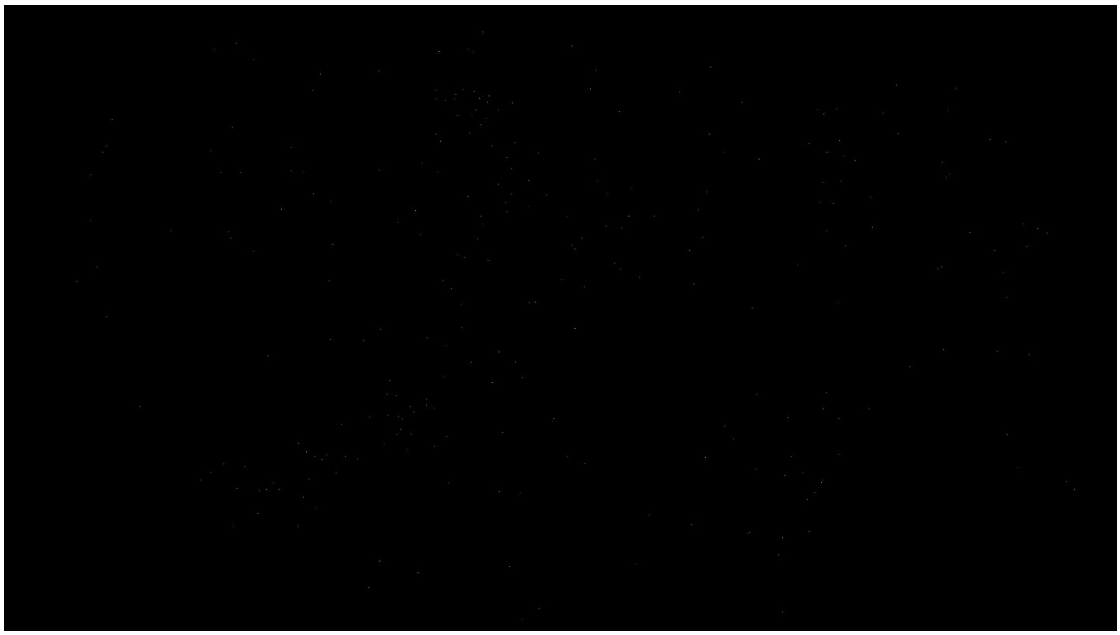


Figure 3.33. Ground truth fixation map.

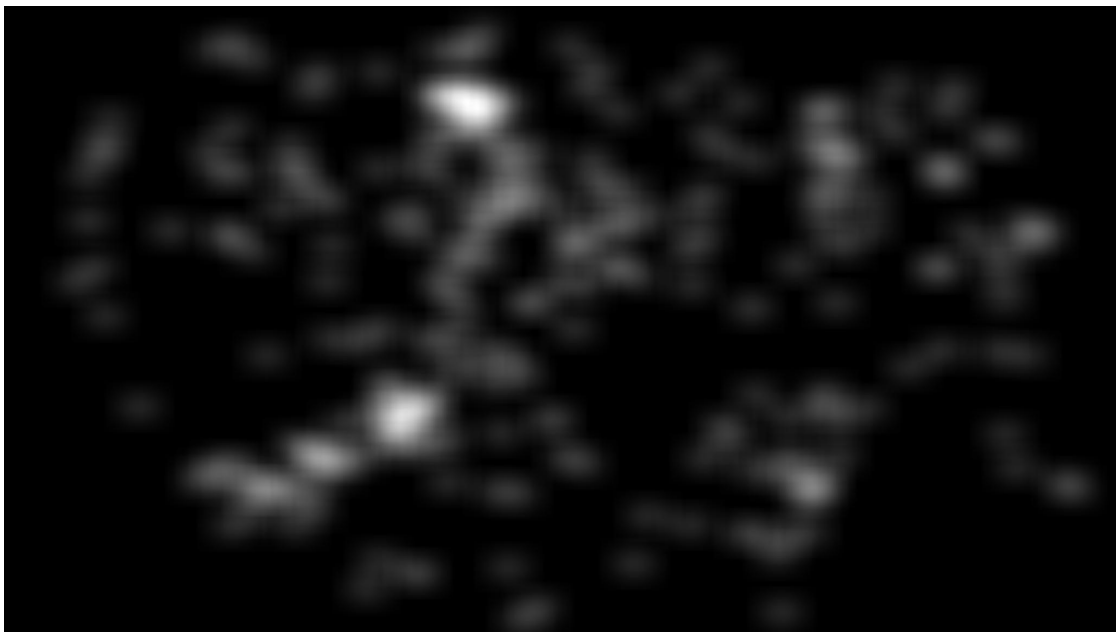


Figure 3.34. Groundtruth density map.

3.4.3.2 Testing Our New Weights

After we finish retraining the SAM-ResNet network, we get four new different weights such as the existing weight by adding bottom-up information in 2015 and 2017 on Salicon and CAT2000 dataset. Firstly, we compare our result to the existing weight of Salicon 2015. Then we follow more testing on the existing weight of Salicon 2017. This is just our initial testing, so we evaluate our result by using only three metrics such as CC, KLD, and NSS. For CC and NSS metric, the higher the better. In contrast, for KLD metric, the lower the better. Result of Salicon 2015 shows that our new weight is better than the existing weight. However, for CAT2000 dataset shows that our new weight is less good than the existing weight, except KLD metric (see Table 3.11). By using the Salicon weight 2017, it indicates that our new weights both retail and CAT2000 dataset are better than the existing weights (see Table 3.12).

Datasets	Metrics		
	CC	KLD	NSS
Retail - original weight	0.4122	0.7565	0.6195
Retail - our new weight	**0.6251**	**0.5254**	**0.9558**
CAT2000 - original weight	0.6486	1.1867	1.8063
CAT2000 - our new weight	**0.6445**	**0.7636**	**1.6845**

Table 3.11. Comparing result between ours and Salicon weight 2015.

Datasets	Metrics		
	CC	KLD	NSS
Retail - original weight	0.4502	0.6554	0.6709
Retail - our new weight	**0.6233**	**0.5247**	**0.9412**
CAT2000 - original weight	0.6172	1.0872	1.6277
CAT2000 - our new weight	**0.6450**	**0.7285**	**1.6641**

Table 3.12. Comparing result between ours and Salicon weight 2017.

3.4.4 Discussion

Our new dataset can improve the saliency maps while there is no top-down information. However, this dataset is small compared to other dataset in deep learning. The deep learning models detect much on top-down information (faces, text). But, they ignore some important bottom-up information (color, orientation). Although our new cannot publish, we can still show that this bottom-up dataset can affect the result of deep learning models.

3.5 In Brief

The summaries of this chapter are:

- Mixing bottom-up and top-down information (text detection): We presented our proposed model which consists of three combination between bottom-up and top-down information. Due to the facts that this was just the first step of our research, many other activities and experiments are required to reach our research goals. Therefore, we must conduct more experiments using both saliency map and text detection features.
- Mixing bottom-up and top-down information (text and face detection): In this experiment, we showed how to simply add top-down information to any bottom-up saliency models in a generic way. Our work mainly focused on both text and face features.
- Mixing bottom-up and top-down information (text, face, objects detection): The purpose of this experiment is to understand the difference in visual attention computation between classical bottom-up saliency models and DNN-based saliency models and the relative importance of bottom-up and top-down information. Our results show that the influence of the main objects in images is the following: **1)** face detection is the most important, **2)** text detection (with about half of the importance of face detection), **3)** animal detection (about half less important than text detection).
- Building a new Retail dataset: In this experiment, we want to show that deep learning models are good for generating saliency maps, but it neglects bottom-up information. We make our own , which is so called Retail dataset. Moreover, we did experiment on 14 people from eye-tracker to get the ground truth for each image. Then we retrain the SAM-ResNet network by using Salicon weight 2015 and 2017. Finally, our results show

that by adding bottom-up information to deep learning models, it can improve saliency maps compared to the fixation maps.

Chapter 4

CNN Features Rarity as an Attention Cue

4.1 Introduction

Human visual system is modeled in engineering field providing feature-engineered methods which detect contrasted/surprising/unusual data into images. This data is "interesting" for humans and leads to numerous applications. The arrival of deep learning (DNNs) let to much more efficient algorithms on the benchmark datasets. However, DNN-based models are counter-intuitive: how surprising or unusual data can be learned while this data is by definition difficult to learn because of its low occurrence probability? In reality, DNNs models mainly learn top-down features such as faces, text, people, or animals which usually attract human attention, but they have low efficiency in extracting surprising or unusual data in the images. In this chapter, we propose a model which uses the power of DNNs feature extraction and the genericity of feature-engineered algorithms. Our tests show that while the results are good on classical saliency benchmarks it is still very good on datasets containing odd objects where DNN-based models provide very poor results even when re-trained. Our model called DeepRare can be found at `https://gitlab.com/matei.mancas/deeprare`.

The human visual system handles a huge quantity of incoming visual information and it cannot carry out multiple complex tasks in the same time on the whole visual field. This bottleneck [3] implies that the human visual system has an exceptional ability of sampling the surrounding world and pay attention to objects of interest. In computer vision, visual attention is modeled through the so-called saliency maps. The modeling of visual attention has numerous applications such as object detection [6], [17], image segmentation [47], [41], image/video compression [26], [20], image re-targeting [46], [59], [45], and so on.

4.1.1 The Age of Feature-engineered Saliency

Since the early 2000, numerous models of visual attention based on feature extraction from images were provided. In this paper, they will be referred as "classical models". While they can be very different, most of them have the same main philosophy: search for contrasted, rare, abnormal or surprising features within a given context. Among those models one may find seminal work of [27] or [56], but also more recent work based on information processing such as AIM [4]. Finally, some models became a reference for classical models such as GBVS [22], RARE [55], BMS [72] or AWS [19].

4.1.2 The Rise of Deep Learning

With the arrival of the deep learning wave, most researchers have focused on Deep Neural Networks saliency (which will be referred as DNN-based in this paper) which triggered a revolution in terms of results on the main benchmark datasets such as MIT benchmark [7] where DNN-based saliency models definitely outperformed classical models. The DNN-based models have been already used in several applications such as image and video processing, medical signal processing, big data analysis, and saliency modeling as well [61], [73], [50], [21], [60]. Some of the DNN-based models became new references such as SALICON [28], MLNet [14] or SAM-ResNet [13].

4.1.3 Trouble into the Deep Learning

However, recently DNN-based models have been criticized for some drawbacks. They underestimate the importance of bottom-up attention [39] which indicates that they were mostly trained to detect the attractive top-down objects rather than detect saliency itself. In [33] the authors found that if saliency models very precisely detect top-down features, they neglect a lot of bottom-up information which is surprising and rare, thus by definition difficult to learn. This shows that saliency cannot be learnt but instead objects which are often attended by human gaze (such as faces, text, bodies, etc.) are learnt and by the way, they are enough to provide good results on the main benchmarks. Recently, [36] introduced two novel datasets, one based on psycho-physical patterns (P^3) and one based on natural odd-one-out (O^3) stimuli. They showed that while DNN-based models are good in MIT dataset on natural images, their results drastically drop on P^3 and O^3. This shows that in addition to not take into account low-level features, DNN-based models are not generic enough to adapt to new datasets of images which are different enough from the train datasets. Finally, we can add the usual DNN black-box drawback which is the impossibility to explain the results. In parallel to the DNN-based models, DeepFeat [40] or SCAFI [62] deal with models where pre-trained deep features are directly used. Those paper will be called "deep-features models" in the paper. Based on the new datasets in [36], we provide a new deep-feature saliency model called DeepRare mixing deep features and the philosophy of an existing classical model [55].

4.2 DeepRare2019 Model

Traditional visual attention models are generic, they generally allow to understand the results obtained are they are good in low-level feature attention. On the other side, DNN-based models provide results which cannot be explained and are sometimes tuned for specific types of datasets, however they are good in higher level and especially top-down information which is most of the time very important in terms of eye-tracking.

We present a model called **DeepRare2019** (**DR19**) which mixes the deep feature extraction advantages and the feature-engineered advantages giving a generic and explainable model. It shows good results whatever the dataset is and considers both low-level and high-level features. **DR19** does not need any training and only uses the default ImageNET training.

4.2.1 Deep Features Extraction

A convolutional network is a great tool for feature extraction. When trained on a general datasets such as ImageNET, the network will extract a complete set of features that one finds in images at several scales (from very low-level in the first layers to very high level in the last ones). We decide here to use a VGG16 network with its default training on ImageNET dataset as a feature extractor. The architecture of VGG16 is shown in Fig. 4.1. There are five blocks of convolutional layer, and we consider them as five groups such as first low-level features (block1), second low-level features (block2), first mid-level features (block3), second mid-level features (block4), and high-level features (block5) as seen in Fig. 4.2.

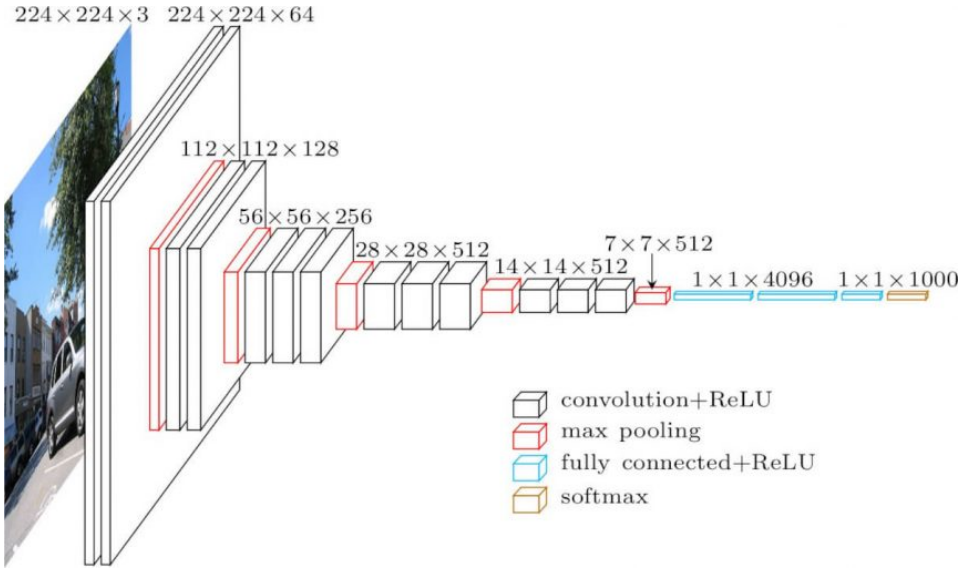


Figure 4.1. The architecture of VGG16.

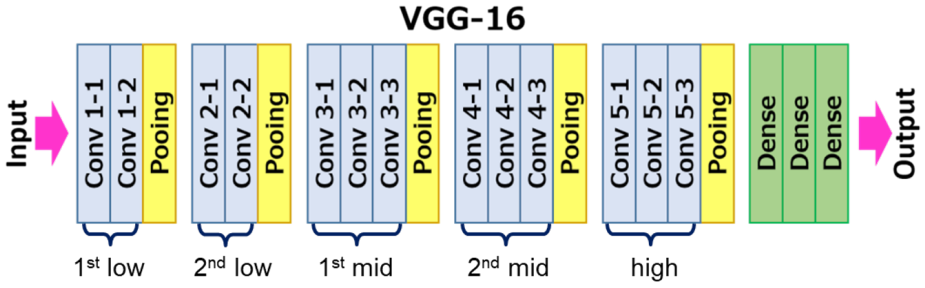


Figure 4.2. Workflow of VGG16. There are five blocks of convolutional layer.

In our implementation, we use the Keras framework to extract any layer and feature map within this layer. We do not use the pooling layers (as they are redundant with the previous convolutional layer) and the final fully connected classification layers. An example for layer 1 can be seen in Fig. 4.3, and it explains more detail of the workflow of VGG16 on how

each block is processed and applied rarity feature. In a VGG16, the convolutional layers are gathered within 5 groups separated by the pooling layers : 1) the first low-level features in layers 1 and 2, then 2) second set of low-level features from layers 4 and 5, after that 3) the first middle-level layers 7, 8 and 9 and 4) the second middle-level layers 11, 12 and 13 and finally 5) the high-level features from layers 15, 16 and 17.

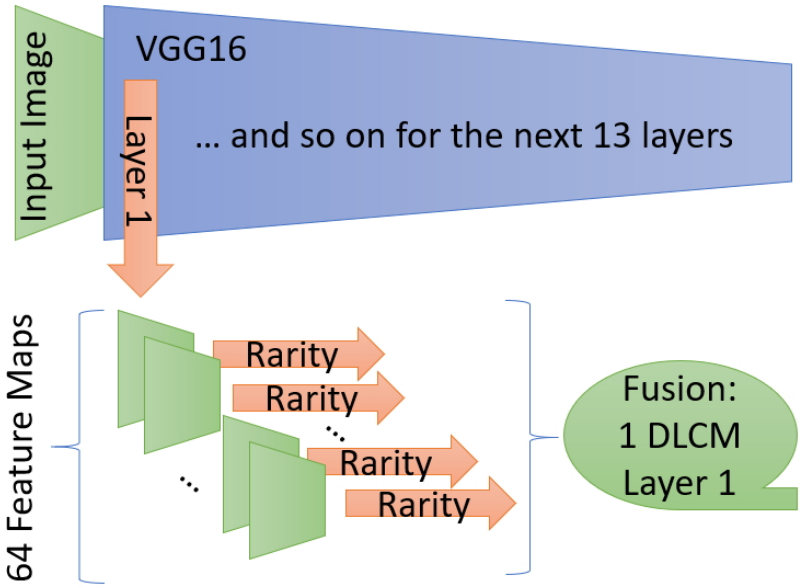


Figure 4.3. Processing for Layer 1. This processing is iterated for all 13 convolutional layers from a VGG16 network.

4.2.2 Rarity of Deep Features and Top-down Information

On each feature map within the layers we compute the data rarity. For that we use the main idea from [55] without the multi-resolution part which is naturally achieved by the VGG16 architecture. A very simple rarity function R based on the histogram of each feature map sampled on a few bins (11 in the current implementation) is used as in equation 4.1.

$$R(i) = -log(p(i)) \tag{4.1}$$

where $p(i)$ is the occurrence probability for the pixels of bin i.

Once the rarity histogram R is computed, the resulting rarity image is reconstructed by backprojection. This image will highlight pixels in the feature map which are rare compared to the other pixels in the feature map. Based on [55], rare pixels are the ones which might attract human attention. Rarity is applied on each feature map of each layer as it can be seen on the 64 feature maps of layer 1 in Fig. 4.3.

4.2.3 Data fusion

Once the rarity of all feature maps is computed, the results need to be fused together. We use a classical map fusion from [26] where the fusion weights depend on the squared difference between the max and the mean of each map. This is applied to all feature maps within each layer leading to 13 deep layer conspicuity maps (DLCM), one for each convolutional layer in VGG16 (see Fig. 4.3 for first layer).

In a second stage, the same fusion method is applied for each of the 5 layer groups arriving to 5 deep groups conspicuity maps (DGCM). Finally, the 5 DGCM are summed up and a top-down face map is added. This face map is the feature map 105 from layer 15 which is known to detect faces [62].

4.2.4 DeepRare2019 Validation

To validate our results, we compare **DR19** to several classical and DNN-based models both in a qualitative and quantitative way on three datasets also used in [36] which are P^3, O^3, and MIT1003. In addition we also compare our results with the DeepFeat [40] model on the MIT1003 dataset.

4.2.4.1 Data and Metrics for Validation

We use 3 datasets namely MIT1003 [30], P^3, and O^3 datasets [36] to validate our results. The MIT dataset has general-purpose real-life images. P^3 dataset evaluates the ability of saliency algorithms to find singleton targets which focuses on color, orientation, and size (without center bias). O^3 dataset depicts a scene with multiple objects similar to each other in appearance (distractors) and a singleton (target) which focuses on color, shape, and size (with center bias).

Concerning metrics, we use measures from [36] such as "number of fixations" (# fix.) defined as the path formed by the saliency maximum followed by the other maxima of the saliency map before reaching the target, the global saliency index (GSI) which measures how well the target mean saliency is distinguished from the distractors, and the saliency ratio. The latter focuses on maximum saliency of the target versus the distractors [57] and the same for the background versus target (MSR_t and MSR_b). We also use standard eye-tracking evaluation metrics from MIT benchmark [7] such as CC, KL, AUC Judd, AUC Borji, NSS, and SIM.

4.2.4.2 Qualitative validation

We compare our model to other models on P^3 and O^3 datasets. According to [36], they observe that most classical models perform better on P^3 than DNN-based models. In contrast, DNN-based models perform better on O^3. DeepRare has a stable behaviour performing well on both datasets (see Fig. 4.4).

Figure 4.5 shows six samples from P^3 dataset which exhibit color, orientation and size differences of the target. While distractors are still visible on **DR19** saliency map, the targets are

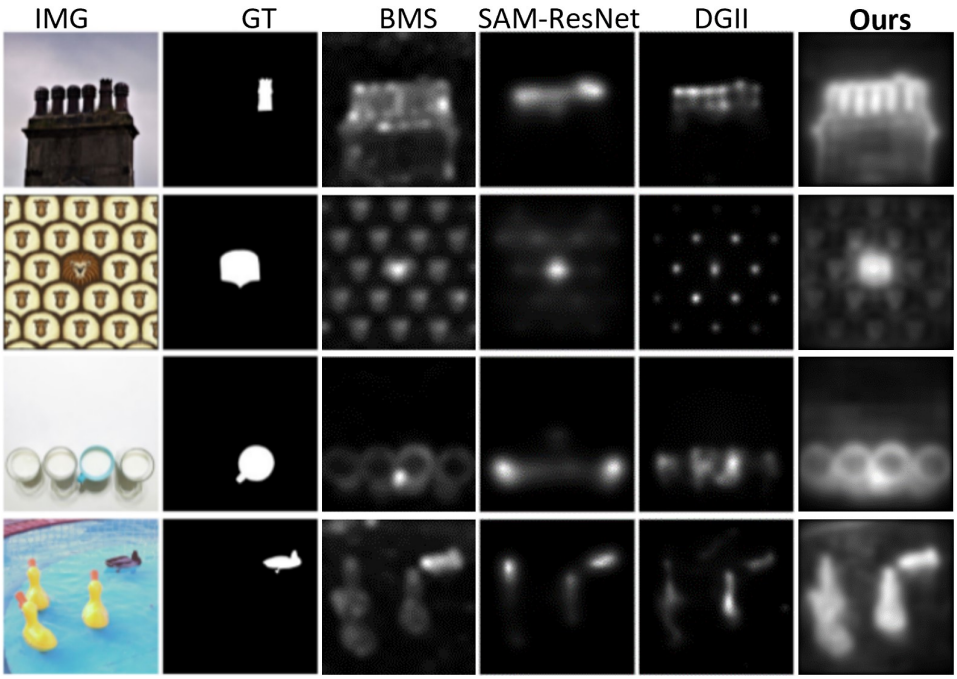


Figure 4.4. Sample images and corresponding saliency maps both the classical and deep models including ours.

always correctly highlighted compared to RARE [55] which works well mainly for colors and two DNN-based models (ML-Net [14] and SALICON [25]) which only work on one sample.

Figure 4.6 and Fig. 4.7 show 3 selected sampled images from O^3 dataset for 4 different target categories. They are difficult and easy for classical and deep model (see Fig. 4.6), classical model performing better and deep model performing better (see Fig. 4.7). In addition, these 2 figures show sample images and saliency maps for a range natural odd-one-out targets. According to [36], the 3 selected sampled images in each category based on average MSR_t: difficult for classical and deep models (MSR_t <1), easy for classical and deep models (MSR_t >2), easier for classical models (MSR_t for classical models is >1 but not for deep ones), and easier for deep models (MSR_t for deep models is >1 but not for classical ones). Again, our model highlights the target more than the other models.

4.2.4.3 Quantitative validation

We compare our model with others on three datasets. First on MIT1003 dataset which shows general-purpose images where learning objects is very important. This dataset is basically one which should provide advantage to DNN-based models which focus on objects (faces, text, etc.).

Second, we use O^3 dataset from [36] which also provides real life images but with odd-out-one regions. The dataset should provide similar difficulty to classical and DNN-based saliency models.

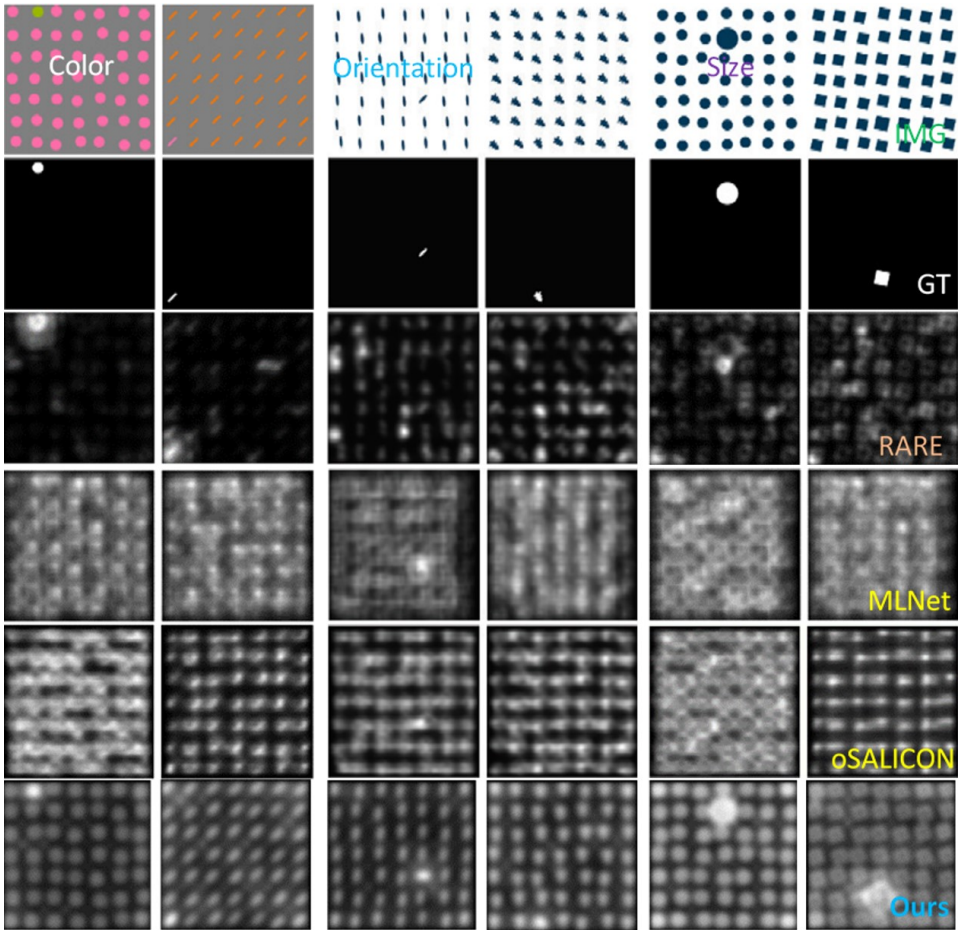


Figure 4.5. Selected samples P^3 dataset. From left to right : target difference in color, orientation, and size. From top to down : initial, ground truth, RARE, ML-Net, SALICON, **DR19**.

Finally, we use P^3 dataset from [36] which shows synthetic psycho-physical rare objects which should work better for classical saliency models.

1). MIT1003 dataset

We summarize in Table 4.1 the results of **DR19** and also results coming from [36] for ML-Net end SALICON where ML-Net was trained with SALICON, P^3 and O^3 datasets and SALICON was trained with OSIE, P^3 and O^3 datasets. For other models (DeepFeat, eDN, GBVS, RARE, BMS, AWS), results are stated in [40]. We remark that our model is less good than SALICON (and probably than newer models such as SAM-ResNet), but equivalent to ML-Net and better than other DNN-based models.

It is also better than DeepFeat and all classical models. While the latest DNN-based models are still better on this dataset than **DR19**, the latter is better than several DNN-based models and all the other classical models.

2). O^3 dataset

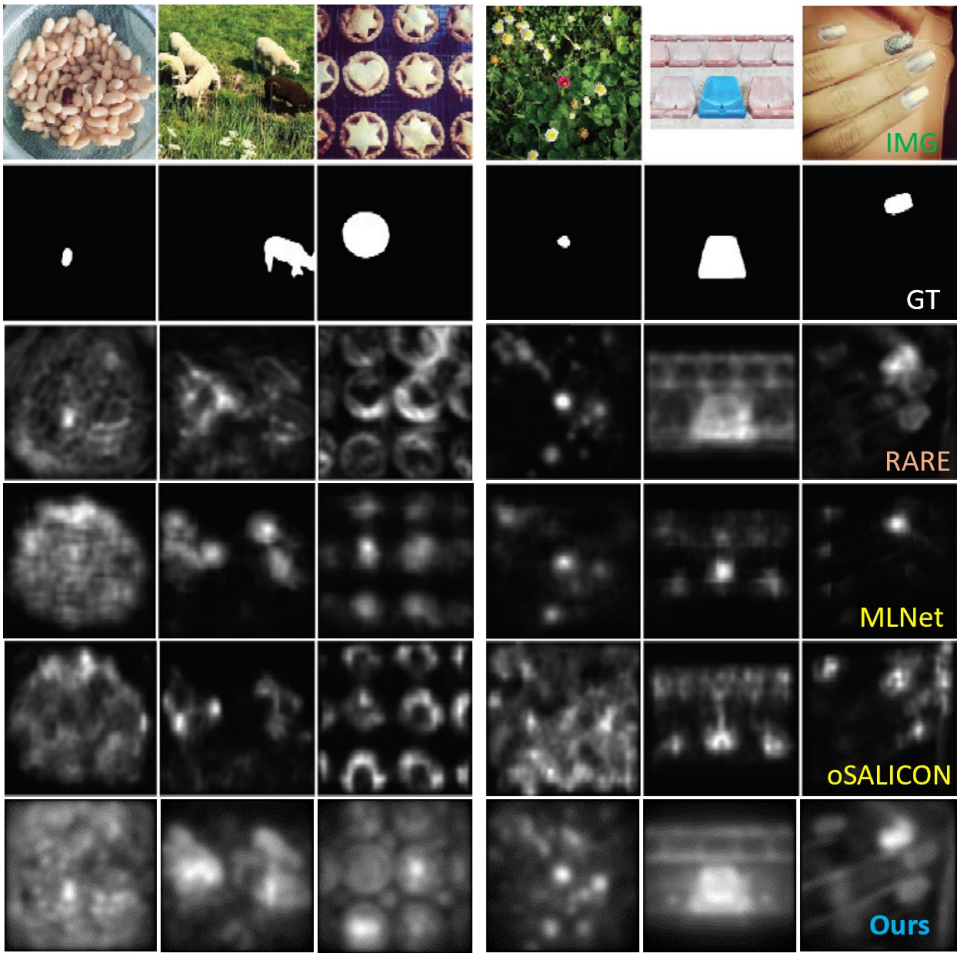


Figure 4.6. Three selected sampled images for a range natural odd-one-out targets. First and last 3 column images, it shows difficult and easy for classical and deep model, respectively.

The O^3 dataset uses the MSR metric defined in [36]. When the MSR_t is higher, it is better as the target is well highlighted compared to the distractors. When MSR_b is lower, it is better, it means that the maximum of the saliency of the target is higher than the one of the background. The first measure will ensure that the target is visible compared to the distractors and the second that it is visible compared to the background.

Table 4.2 shows the MSR from [36] where we added **DR19** at the end splitting the dataset between the images where color is a good discriminator (Color) and the others (Non-color). All models work better for targets where color is an important feature and less well for non-color. For MSR_t for Color our model is less good especially compared to DNN-based models. However we can see that for Non-color images where the models fail much more our model has a remarkable stability being second and very close the the best one (SAM-ResNet). If we take into account the MSR_b, our model clearly outperforms all the others (smallest MSR_b) providing the best discrimination between the target and the background. **DR19** is the only

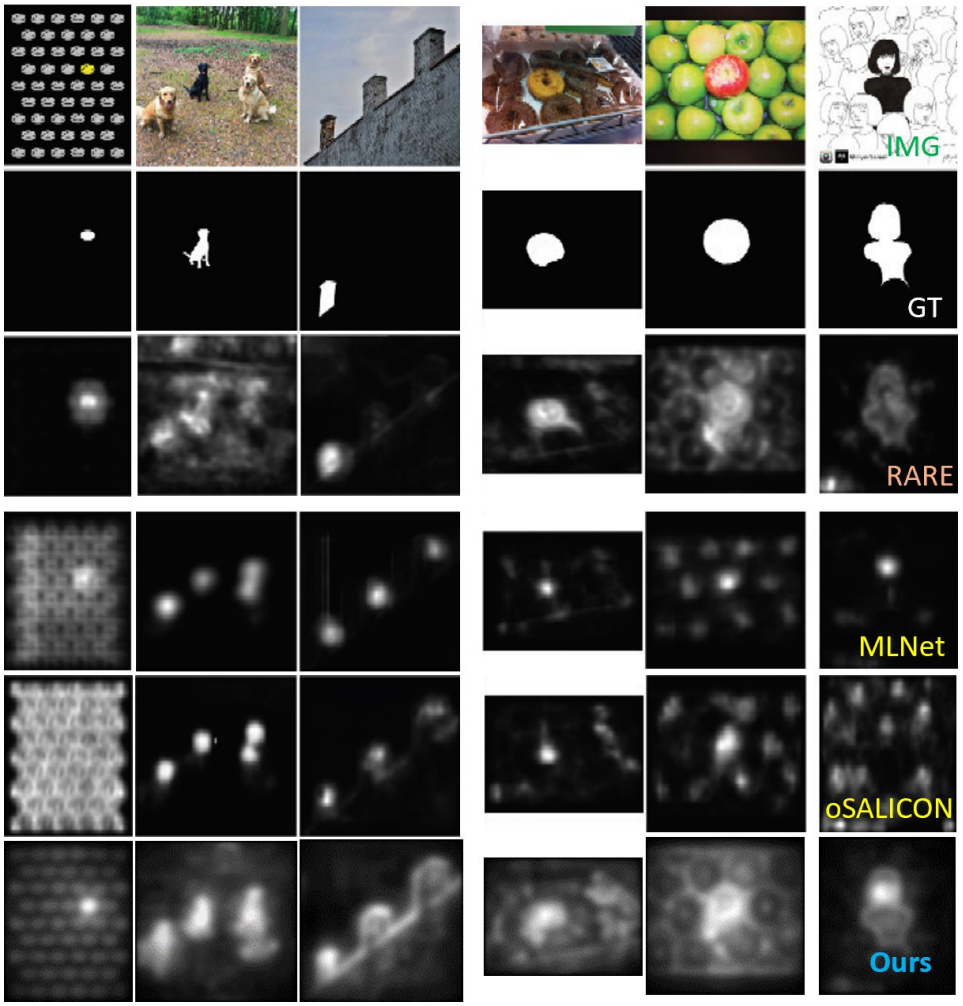


Figure 4.7. Three selected sampled images for a range natural odd-one-out targets. First and last 3 column images, it shows classical perform better and deep perform better, respectively.

model with a MSR_b smaller than 1 which means that in average the maximum of the target saliency is higher than the maximum of the background saliency.

Table 4.3 shows the results of the two DNN-based models also tested in Table 4.1. Our model outperforms both the SALICON and ML-Net models on both MSR_t and MSR_b metrics.

3). P^3 dataset

The P^3 dataset is the one which exhibits the less top-down information and it even does not have any centered bias. Naturally, for this dataset, the DNN-based models perform the worst. We will check here how DeepRare deals with the data.

First we use the average number of fixations (# fix.) and found percentage (% found) metrics. Table 4.4 shows first the results on P^3 for **DR19** compared with SALICON and ML-Net models. Our model outperforms the two DNN-based models and needs much less fixations to discover more of the targets showing here very good results.

Model	AUCJ ↑	AUCB ↑	CC ↑	KL ↓	NSS ↑	SIM ↑
SALICON [25]	0.83	-	**0.51**	**1.12**	**1.84**	**0.41**
DR19	***0.86***	***0.85***	*0.48*	*1.25*	*1.58*	*0.36*
ML-Net [14]	0.82	-	0.46	1.36	1.64	0.35
DeepFeat [40]	0.86	0.83	0.44	1.41	-	-
eDN [68]	0.86	0.84	0.41	1.54	-	-
GBVS [22]	0.83	0.81	0.42	1.3	-	-
RARE [55]	0.75	0.77	0.38	1.41	-	-
BMS [72]	0.75	0.77	0.36	1.45	-	-
AWS [19]	0.71	0.74	0.32	1.54	-	-

Table 4.1. DeepRare2019 results on the MIT1003 dataset. DFeat, eDN, GBVS, RARE, BMS, AWS results are stated in [40] and SALICON and ML-Net results are stated in [36].

Model	Color		Non-color		All targets	
	MSR_t ↑	MSR_b ↓	MSR_t ↑	MSR_b ↓	MSR_t ↑	MSR_b ↓
SAM-ResNet [13]	**1.47**	1.46	**1.04**	1.84	**1.40**	1.52
CVS [18]	1.43	2.43	0.91	4.26	1.34	2.72
DGII [38]	1.32	1.55	0.94	1.95	1.26	1.62
FES [52]	1.34	2.53	0.81	5.93	1.26	3.08
ICF [39]	1.30	2.00	0.84	2.03	1.23	2.01
BMS [72]	1.29	0.97	0.87	1.59	1.22	1.07
DR19	*1.14*	***0.75***	***1.00***	***1.00***	*1.06*	***0.89***

Table 4.2. DeepRare2019 results on O^3 dataset: 3 targets. DNN-based models are in bold.

Model	MSR_t ↑	MSR_b ↓
ML-Net [14]	0.96	0.91
SALICON [25]	0.90	1.26
DR19	**1.06**	**0.89**

Table 4.3. DeepRare2019 results on a whole O^3 dataset.

Figure 4.8 shows that compared to state-of-the-art models (top graph), our model (bottom-graph) ranges between 80% of targets found after 15 fixations to 88% target found after 100 fixations. It is possible to see that even after 15 fixations more than 80% of the targets are found which is much better than all tested DNN-based models and all classical models excepting IMSIG [23] which has equivalent results.

Model	Avg. # fix. ↓	% found ↑
ML-Net [14]	42.00	0.44
SALICON [25]	49.37	0.65
DR19	**16.34**	**0.87**

Table 4.4. DeepRare2019 results on P^3 dataset.

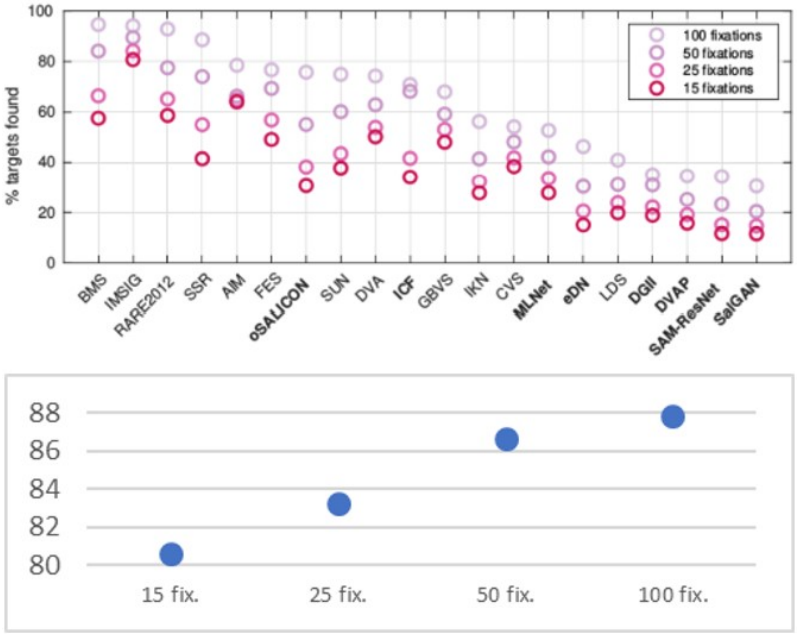


Figure 4.8. Number of fixations vs. % of targets detected. Top graph: results from [36] for state-of-the-art models. Bottom graph: **DR19** model. Labels for DNN-based models are shown in bold.

Figure 4.9 shows the results similar to Fig. 4.8, but it focuses on singleton feature (color, orientation, size). For color feature, the classical model, RARE [55] performs better than other classical and deep models. However, our model performs better than some of the classical models and all of the deep models. It reaches to 87% mark in average.

For the GSI score, Fig. 4.10, Fig. 4.11, and Fig. 4.12 let us compare the three best classical models with the three best DNN-based models on the left and **DR19** results on the right.

For color targets (see Fig. 4.10, right graph) we see that the maximum of GSI score for **DR19** is 0.56 which puts our model under BMS, RARE2012 and IMSIG but much better than all the other models. In addition, the shape of the GSI curve exhibited by **DR19** is coherent from a biological point of view: if the difference between the target color and the distractor color is small, then the model detects less well the target (left-side of the curve) than when the color of the target and background is very different (right-side of the curve). Our model is the only one to provide a biologically plausible GSI curve.

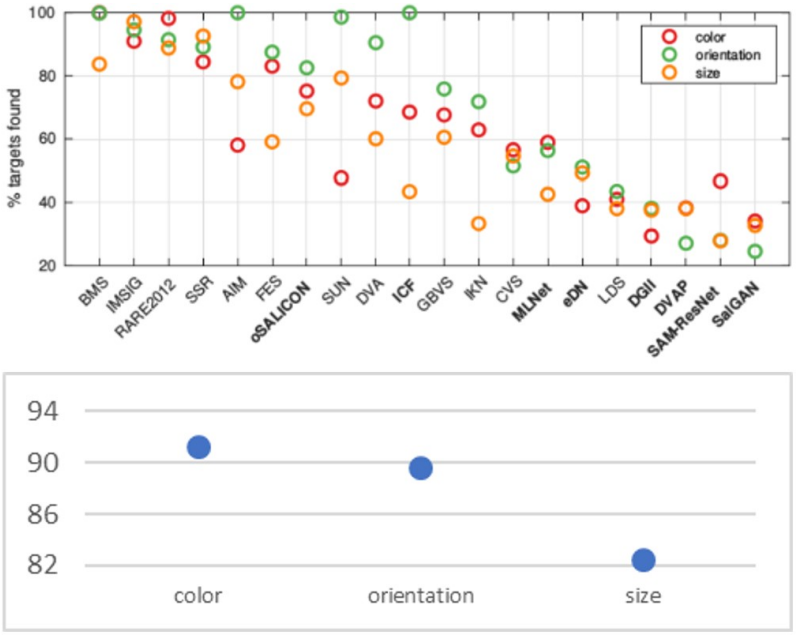


Figure 4.9. Singleton feature vs. % of targets detected. Top graph: results from [36] for state-of-the-art models. Bottom graph: **DR19** model. Labels for DNN-based models are shown in bold.

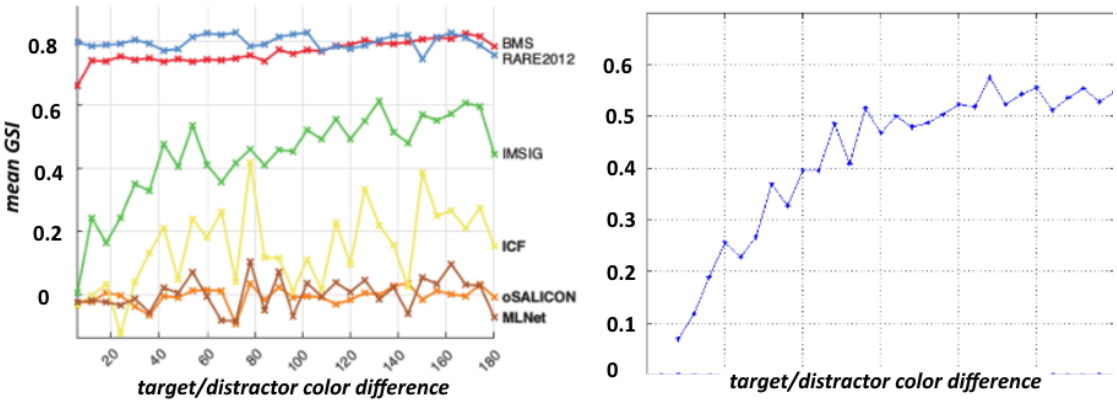


Figure 4.10. The GSI score for color target/distractor difference. Left plot: generated by [36]. Right plot: **DR19** model.

For orientation targets (see Fig. 4.11, right graph) we see that the maximum of GSI score is about 0.22. This makes **DR19** better than any other model in terms of maximum. Also, the shape of the GSI curve exhibited by DeepRare is again coherent from a biological point of view: if the difference between the target orientation and the distractor orientation is small (left-side of the curve), then the model detects the target less well than when target orientation is very different from the distractors (right-side of the curve). Our model is again the only one to provide a biologically plausible GSI curve.

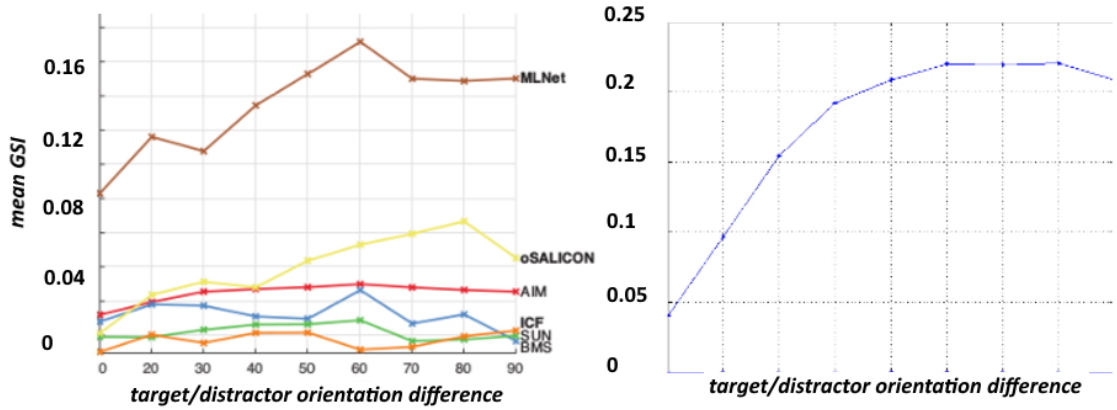


Figure 4.11. The GSI score for orientation target/distractor difference. Left plot: generated by [36]. Right plot: **DR19** model.

For size targets (see Fig. 4.12, right graph) we see that the maximum of GSI score is about 0.25 which makes it close to RARE2012 in terms of maximum GSI. The shape of the GSI curve exhibited by our model is finally again coherent from a biological point of view: if the difference between the target size and the distractor size is small (center of the curve), then the model detects the target less well than when its size is very different (left and right sides of the curve). Our model is again the only one to provide a biologically plausible GSI curve. We can also see a dissimmetry in the curve showing that it is easier for **DR19** to detect target twice bigger than distractors than targets twice smaller than the distractors which is again biologically coherent.

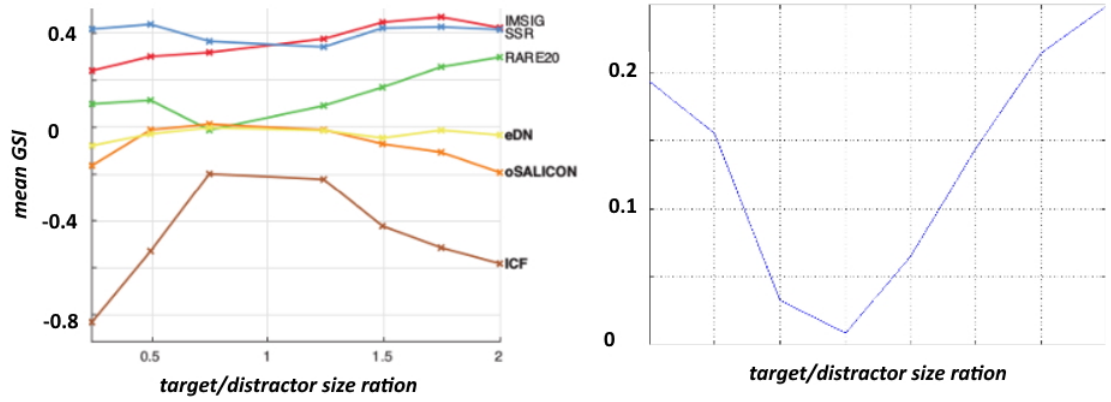


Figure 4.12. The GSI score for size target/distractor ratio. Left plot: generated by [36]. Right plot: **DR19** model.

4.2.5 Discussion

We proposed a novel saliency model called **DeepRare2019** by using the rarity idea of [55] applied on the deep features extracted by the VGG16 network pretrained on the ImageNet dataset. This exhibits several interesting features:

- It needs no training, and the default ImageNet training is enough.
- The model is computationally efficient and is easy to run on CPU at less than one second per image.
- Our approach is very modular, and it is very easy to adapt to any neural network architecture such as VGG19, ResNET50, or MobileNetV2 for adaptation on mobile devices such as smart phones.
- It is possible to check each layer contribution and thus better understand the result contrary to black-box DNN-based models.
- **DeepRare2019** is very generic and stable through all kinds of different datasets where other models are sometimes better but only for one dataset and/or a specific metric but much worse for the others.

We show that this model is definitely the most stable and generic when testing it on 3 very different datasets. It was first tested on MIT1003 where it outperforms all the classical models and most of the DNN-based models. However some DNN-based models, especially the latest ones still provide better results. We then tested **DeepRare2019** on the O^3 dataset, where it outperforms all the models on target/background discrimination. On target/distractor discrimination, other models perform better for Color, but our model is second on non-Color showing its stability again. Finally, on P^3 dataset, our model is first ex-aequo for the target discrimination based on the number of fixations. When computing the average GSI metric our model is the only one to be in the top-three for all the features (color, orientation, size) and the only one to exhibit a GSI plot which is biologically plausible.

While one cannot expect from an unsupervised model such as **DeepRare2019** to be better on MIT1003 dataset than DNN-based models which are trained and tuned on similar data, those DNN-based models are bad or even completely lost on O^3 and P^3 datasets. The other way around, classical models are sometimes better than **DeepRare2019** on the latter datasets, but they perform much worse than **DeepRare2019** on MIT1003 dataset. In addition, they outperform **DeepRare2019** only on specific metrics and never on all the dataset subclasses.

4.2.6 Conclusion

In brief, **DeepRare2019** is always the best or in the top-3 or top-4 best models in all tests we achieved. No other model is capable to be good in all datasets and their subclasses. **DeepRare2019** is definitely the most stable and generic model within the tested saliency models.

All those advantages show that deep-features-engineered models might become a good choice in visual attention field especially when the images they are applied on are not very specific or when specific eye-tracking datasets are not available.

4.3 DeepRare2021 Model

4.3.1 Objective

In this section, we propose an upgrade visual attention model called **DeepRare2021** (**DR21**) which uses the power of DNNs feature extraction and the genericity of feature-engineered algorithms. This algorithm is an evolution of a previous version called **DeepRare2019** (**DR19**) based on a common framework. **DR21** 1) does not need any training and uses the default ImageNet training, 2) is fast even on CPU, 3) is tested on four very different eye-tracking datasets showing that the **DR21** is generic and is always in the within the top models on all datasets and metrics while no other model exhibits such a regularity and genericity. Finally **DR21** 4) is tested with several network architectures such as VGG16 (V16), VGG19 (V19) and MobileNetV2 (MN2) and 5) it provides explanation and transparency on which parts of the image are the most surprising at different levels despite the use of a DNN-based feature extractor.

While in **DR19**, the algorithm is applied only to a VGG16 architecture, **DR21** can be applied to various convolutional architectures. In this part we apply it to a VGG16, VGG19, and MobileNetV2 architectures. While VGG19 is a variant of the VGG16 architecture, MobileNetV2 is very different and it has the advantage to be light in terms of weight and computation which make it usable on embedded devices such as smartphones, etc. The rarity is not computed on all the layers to avoid adding unnecessary information. For VGG16, we do not use (1) the pooling layers (as they are redundant with the previous convolutional layer) and (2) the final fully connected classification layers. For VGG19, the same approach was taken into account. We take layer 1 and 2 for the first low-level features; layer 4 and 5 for the second low-level features; layer 7, 8, 9, and 10 for the first middle-level features; layer 12, 13, 14, and 15 for the second middle-level features; and layer 17, 18, 19, and 20 for the high-level features. For MobileNetV2, we use the same approach as VGG16 and VGG19. However, the architecture is much more complex. We take layer 16 and 18 for the first low-level features; layer 24 and 32 for the second low-level features; layer 41, 50, 59, and 67 for the first middle-level features; layer 76, 85, 94, and 102 for the second middle-level features; and layer 111, 120, 137, and 146 for the high-level features.

4.3.2 Digging into Rare Deep Features

Once we decided the layers which will be taken into account into the model and we computed their rarity, we can go further and select the most rare features in the feature maps. In that aim we decided to apply a threshold on the computed rarity maps. This threshold is applied directly on the rarity of each feature map and varies from 0 (no threshold) to 0.9 (only keeping the 10% most rare features) by steps of 0.1. A binary threshold is first obtained and used as a mask on the feature map to keep only the values within this mask while the rest are set to 0.

In this section we inspect the rare deep features at different scales to understand what this rarity thresholds physically mean. One advantage of **DR21** is that it is possible to investigate at which scale and where the feature rarity is important and thus let us understand how the attention mechanism works and how the image structures are taken into account. In this

section, Fig. 4.13, Fig. 4.14, and Fig. 4.15 are computed with a VGG16 architecture and the 5 groups discussed in section 4.2.1.

In Fig. 4.13 we inspect a simple image with an obvious low-level focus of attention. The initial image (on the left) represents several horizontal blue bars while only one is in red. This red bar is an obvious point of attention based on a low-level feature : the color.

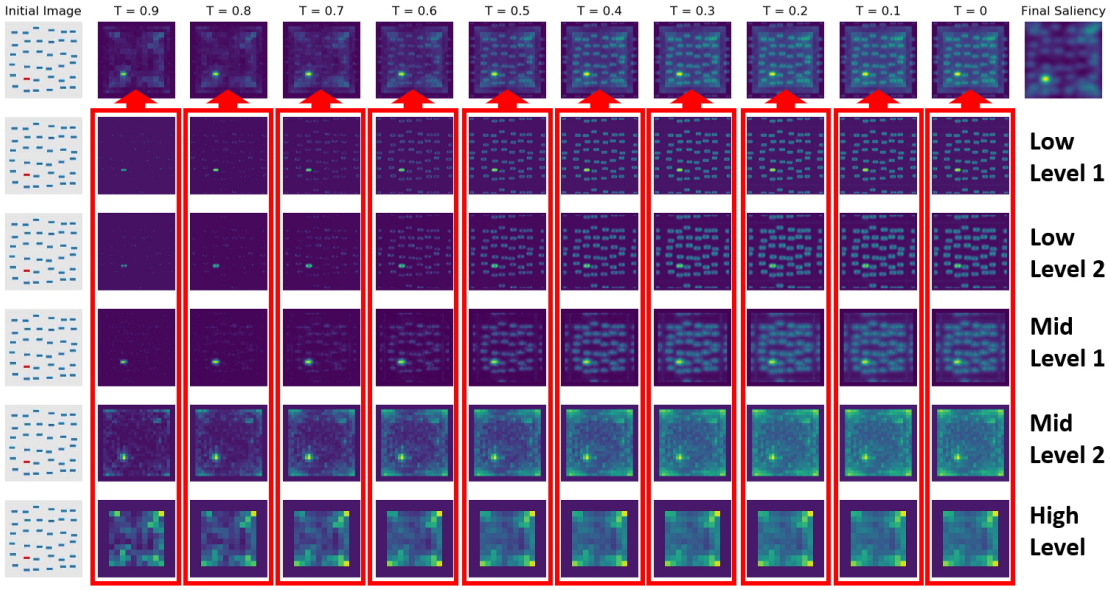


Figure 4.13. Example 1: Detailed maps of different levels (from Low Level 1 to High Level) and different thresholds on feature rarity (from 0.9 to no threshold) within the VGG16 architecture.

From this image, there are 10 columns with different thresholds from T=0.9 which only keep the 10% rarest features to T=0 where no threshold was applied to the rarity feature maps. Lines 2 to 6 represent the features for different levels (5 levels when using a VGG16 architecture) which are already a fusion of the selected layers rarity maps (for the fusion, see next section). The final fusion of the 5 levels can be found on the first line. The post-processing saliency map (see details in section 4.3.4) can be found on top-right of the image.

For the higher threshold (T=0.9) the abnormal region is detected on all levels except the higher level where the edge effects are too important (and can be seen in the corners even when the edges of the image are set to 0). For the low levels (such as level 1 to level 3) only the red pattern appears and the model is very precise and selective on the rare object. When going towards the right with lower thresholds, little by little, the other blue patterns also appear while the red one is still the most highlighted but the distractors around are visible.

In Fig. 4.14 one can see the result for a situation where mid-level (big letters) and high-level features (such as text and people) are the rare features (see initial image on the left). This image has a less obvious attention focus as the one in Fig. 4.13.

For the higher threshold (T=0.9) the abnormal regions are split between mid levels and the higher level. While at the low-levels very few information passes the threshold, for the higher levels text and the person are well highlighted. For the lest level the bigger text and the

person are more highlighted than small text. At smaller thresholds the low levels highlight mostly the posters on the wall based on their colors but not enough he person and the large letters on top right.

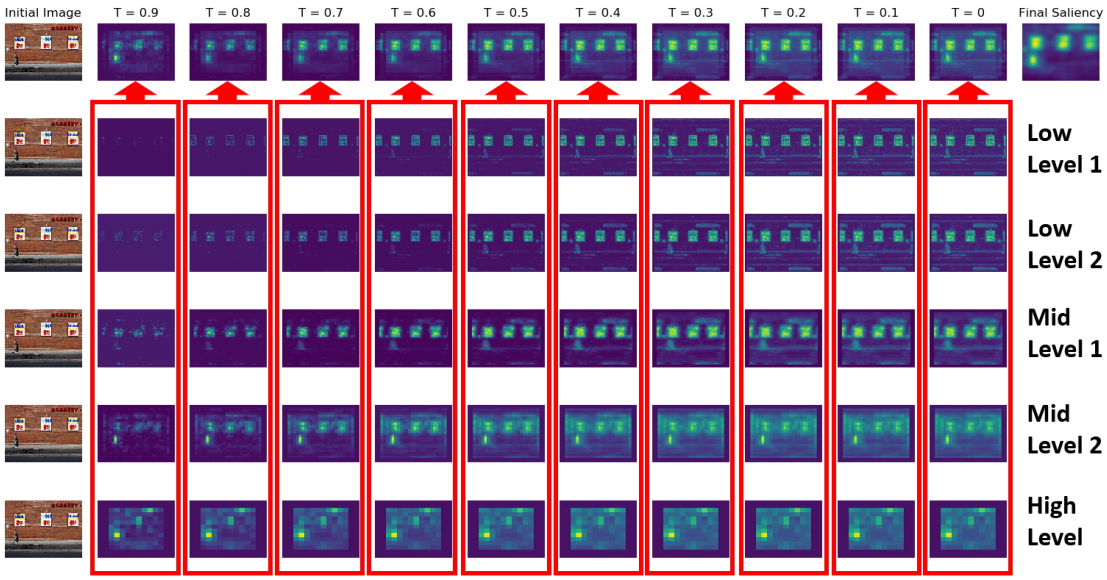


Figure 4.14. Example 2: Detailed maps of different levels (from Low Level 1 to High Level) and different thresholds on feature rarity (from 0.9 to no threshold) within the VGG16 architecture.

In Fig. 4.15 one can see for a situation where high-level features (big cake shape and color) are the rare features. For the higher threshold (T=0.9) the abnormal regions are only detected in the higher level (mid level 2 and especially high level). On all the other levels no interesting feature is highlighted. For small thresholds, for low level 1 and 2 and mid level 1 only edges and object areas are highlighted but the model fails in detecting the different cake. We see that here, the low level feature never detect the abnormal cake whatever the threshold is.

Overall in Fig. 4.13, Fig. 4.14, and Fig. 4.15 mid level 2 and high level provide always better results with a high threshold such as T = 0.9, while lower level feature work better on this high threshold only in specific kind of images with obvious abnormal patterns dues to low-level features. We already understand from here that several thresholds need to be combined to provide better final results.

In [33] the authors showed that top-down information for high-level features such as text, people, animals or transportation had a huge impact on visual attention through the mix of those features with a simple rarity bottom-up approach. But the rarity-based features were only low-level features. In the current section, we use both mid-level and high-level features however we do not add top-down information (except for a weak face detector only added when the VGG16 architecture is used). In the following section we show how the thresholded rarity feature maps from the chosen layers are fused together.

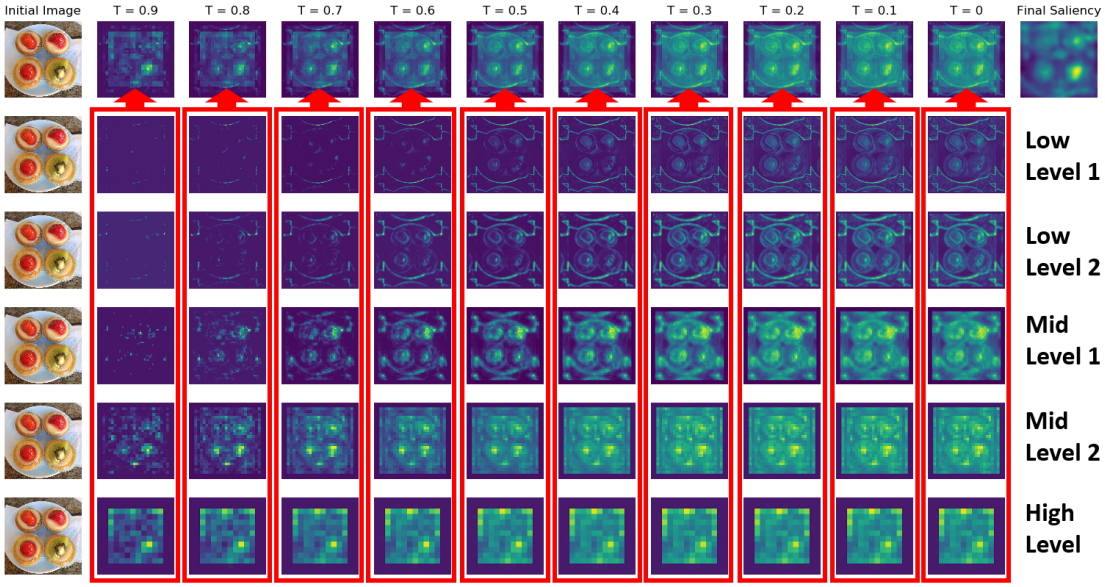


Figure 4.15. Example 3: Detailed maps of different levels (from Low Level 1 to High Level) and different thresholds on feature rarity (from 0.9 to no threshold) within the VGG16 architecture.

4.3.3 Data Fusion

We show here different configurations of thresholds on the layers and check the results for the VGG16 architecture (see Table 4.5 and Table 4.6). The accuracy is here computed by using the correlation metric (CC) between the final saliency map and the real people gaze obtained by using eye-tracking.

VGG16	With face	Without face
Thresholds	CC	CC
0	0.55	0.53
0.9	0.56	0.55
(0+0.9)/2	0.57	0.56
(0.4+0.9)/2	0.57	0.56

Table 4.5. The OSIE dataset. It applies both face and without face features on VGG16 network.

We observe that on two different validation datasets with natural images (the OSIE and MIT1003 dataset) the use of the face improves the results. On the OSIE dataset, the use of the higher threshold (0.9) or no threshold (0) has different effects producing better results on the threholded rariy layers on OSIE (see Table 4.5) and less good resuls on the MIT1003 (see Table 4.6). However, the combination of the thresholds 0 and 0.9 is better in both cases while

VGG16	With face	Without face
Thresholds	CC	CC
0	0.47	0.46
0.9	0.45	0.43
(0+0.9)/2	0.48	0.47
(0.4+0.9)/2	0.47	0.45

Table 4.6. The MIT1003 dataset. It applies both face and without face features on VGG16 network.

the combination between 0 and 0.4 is a little less good on images from MIT1003. These tests show that it always works better to mix the 0 threshold which shows all the data classified by order of rarity and the 0.9 which is the higher threshold which only lets the most rare regions pass. At the end we have the best mi which is to take into account all the rare data (threshold 0) and reinforce the areas with very rare data (threshold 0.9).

4.3.4 Saliency Map Post Processing

Once maps were fused, it is well known [29] that a post-processing of the saliency maps can improve the final results depending on the validation metrics. Indeed, the eye-tracking data which is used for validation leads to rather fuzzy eye-tracking saliency maps, thus the correlation with fuzzy predicted saliency maps will be better. Here we used a gaussian low-pass smoothing filtering approach to optimize the final saliency map with the same parameters as in [33].

In addition of smoothing the data we tested the fact of squaring the data after the smoothing. Table 4.7 and Table 4.8 show the results for the chosen configuration in section 4.3.3 which is the mix of threshold 0 and 0.9 1) not filtered, 2) using the filtering technique from [44] and 3) squared after the filtering technique. We can see that in all cases the filter followed by the square provides the best results. When trying to put the image at power 3 or more, results are less good so we decided to keep as the final post processing scheme the filtering from [44] followed by the squared map.

VGG16	With face	Without face
(0+0.9)/2	CC	CC
No filtered	0.54	0.53
Filtered	0.57	0.56
Filtered + squared	0.59	0.58

Table 4.7. The OSIE dataset. It tests on threshold 0 and 0.9 by considering on without filtering, filtering, and filtering in power 2.

VGG16	With face	Without face
(0+0.9)/2	CC	CC
No filtered	0.43	0.42
Filtered	0.48	0.47
Filtered + squared	0.51	0.50

Table 4.8. The MIT1003 dataset. It tests on threshold 0 and 0.9 by considering on without filtering, filtering, and filtering in power 2.

4.3.5 DeepRare2021 Validation

To validate our results, we compare **DR21** to several classical and DNN-based models including **DR19** both in a qualitative and quantitative way on four datasets also used in [36] which are P^3, O^3, and MIT1003 and OSIE. In addition we also compare our results with the DeepFeat [40] model on MIT1003 dataset.

We use 4 datasets namely OSIE [70], MIT1003 [30], P^3, and O^3 datasets [36] to validate our results. The OSIE dataset contains information at three lavels: pixel-level image attributes, object-level attributes, and semantic-level attributes. The MIT dataset has general-purpose real-life images. P^3 dataset evaluates the ability of saliency algorithms to find singleton targets which focuses on color, orientation, and size (without center bias). O^3 dataset depicts a scene with multiple objects similar to each other in appearance (distractors) and a singleton (target) which focuses on color, shape, and size (with center bias). We decided to use these 4 very different datasets to check how saliency models behave when facing images in different contexts. Concerning metrics, we use all the same metrics which are described in section 4.2.4.1.

4.3.5.1 Qualitative Validation on the Different Datasets

We compare our model to other models on P^3 and O^3 datasets. According to [36], they observe that most classical models perform better on P^3 than DNN-based models. In contrast, DNN-based models perform better on O^3.

Figure 4.16 shows six samples from P^3 dataset which exhibit color, orientation, and size differences of the target. While distractors are still visible on **DR19** saliency map, the targets are always correctly highlighted compared to RARE2012 which works well mainly for colors and two DNN-based models (ML-Net and SALICON) which only work on one sample. **DR21** also spots all the targets but in addition, it highly decreases the distractors influence making the results very close to the ones in line 2 (ground truth).

Figure 4.17 shows images from O^3 dataset for different target categories (easy or difficult). Again, **DR19** highlights the target better than the DNN-based models. **DR19** seems equivalent in average with RARE. **DR21** shows again a much more precise detection eliminating distractors and background information. From a qualitative point of view, on the image in Fig. 4.17, **DR21** is the closest to the second line images (ground truth).

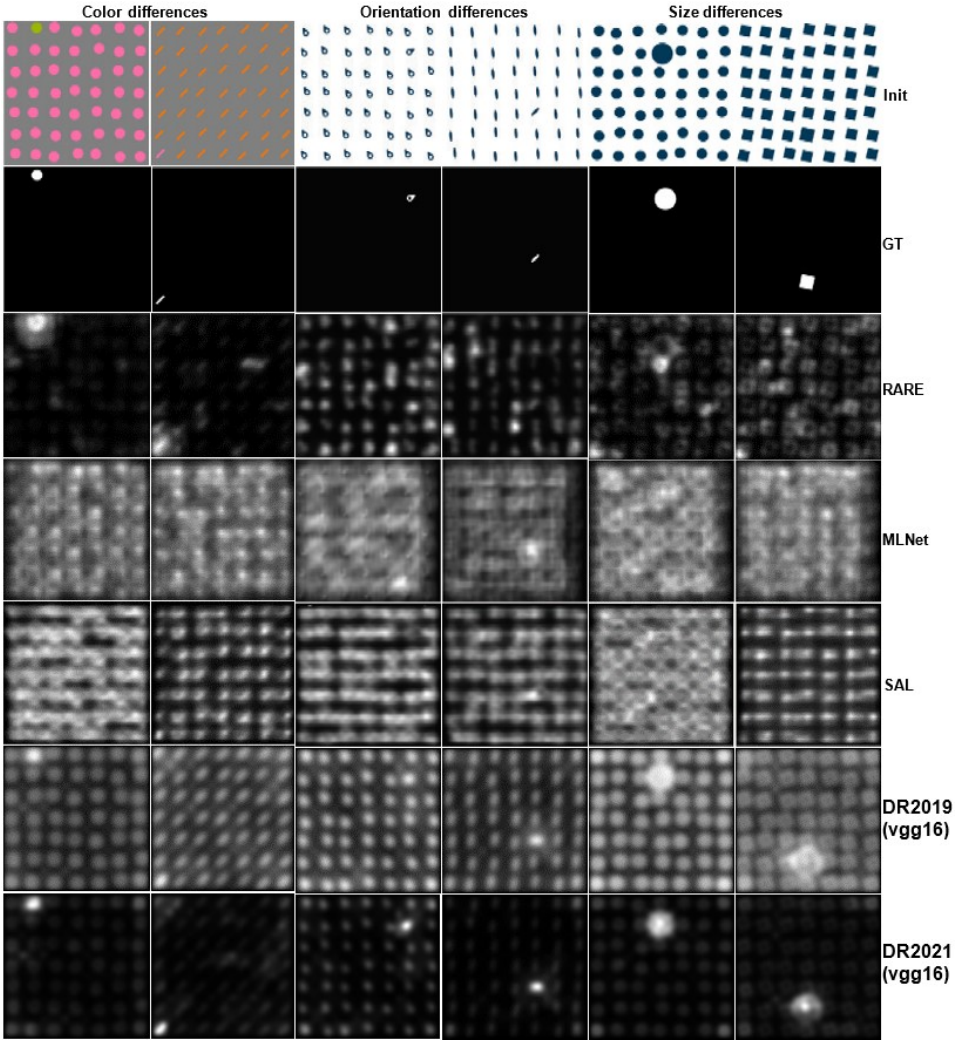


Figure 4.16. Selected samples P^3 dataset. From left to right : target difference in color, orientation, and size. From top to down : initial image, ground truth, RARE, MLNET, SALICON, DR19, and DR21.

Figure 4.18 shows images from MIT1003 dataset. **DR19** always finds the ground truth (GT) focus regions (except for the right image where one GT focus is just in the middle probably due to the centered bias) but it also has details around those focus areas which might decrease its scores on MIT1003. **DR21** is more precise but still keeping the same focus areas. Compared to ground truth (line 2) the focus areas are the same but probably less focused as other DNN-based models which might affect its scores even if those scores should be higher than **DR19**.

Figure 4.19 shows the images from OSIE dataset. **DR19** again spots the main correct salient regions but exhibits a lot of noise or distractors around them with a saliency map less focused as the one of the ground truth (line two). This issue is partially solved by **DR21** which is much more selective but still less than some DNN-based models.

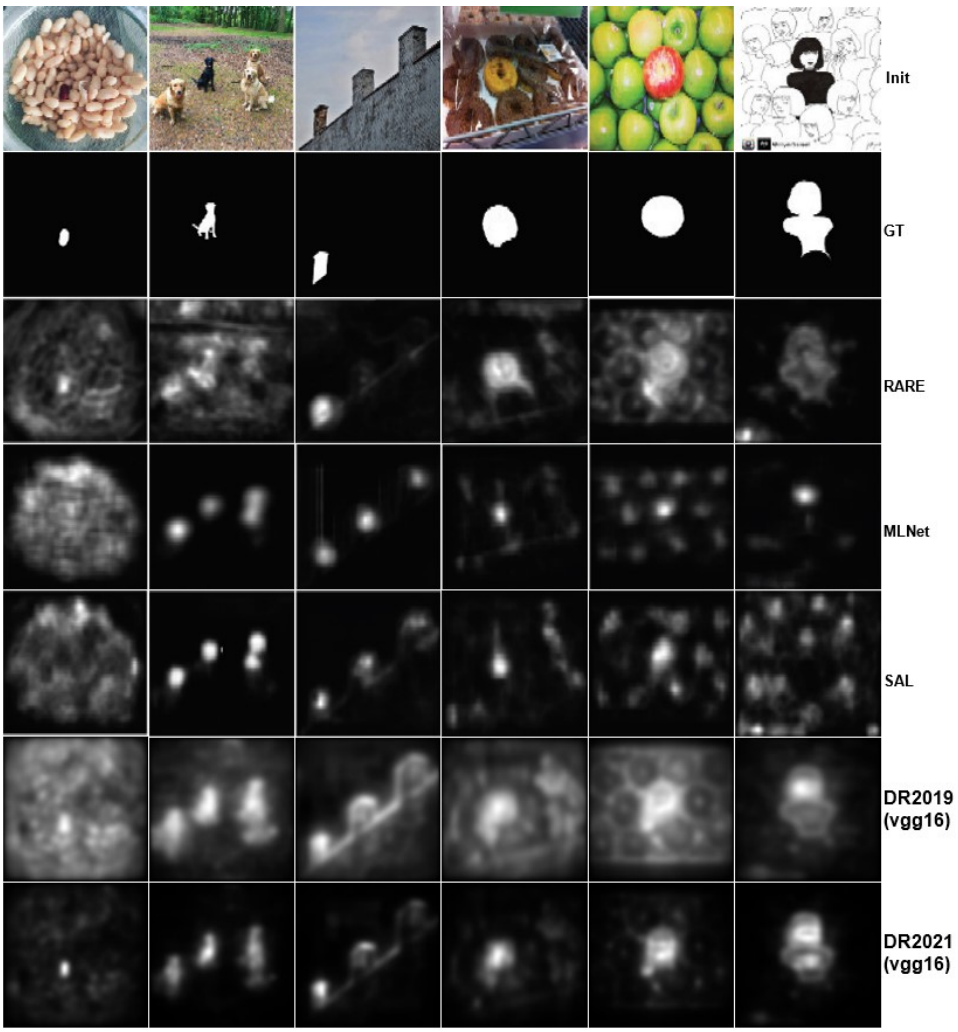


Figure 4.17. Selected samples O^3 dataset. From top to down : initial image, ground truth, RARE, MLNET, SALICON, DR19, DR21.

Overall, the qualitative study reveals that **DR21** spots most of the time the most important regions in all datasets. On the MIT1003 and OSIE dataset the results of **DR21** are in most cases correct. If some DNN-based models are probably better on the MIT1003 or OSIE datasets, one reason is that they are more focused on the salient areas only as the ground truth is. Indeed, DNN-based models were trained on images close the the ones in those two datasets. On the O^3 and P^3 dataset, **DR21** clearly show their superiority on DNN-based models which are sometimes completely lost with very bad results. **DR19** and especially **DR21** exhibits the most stable behavior performing well on all datasets while other models might be good on some images but much less good on others.

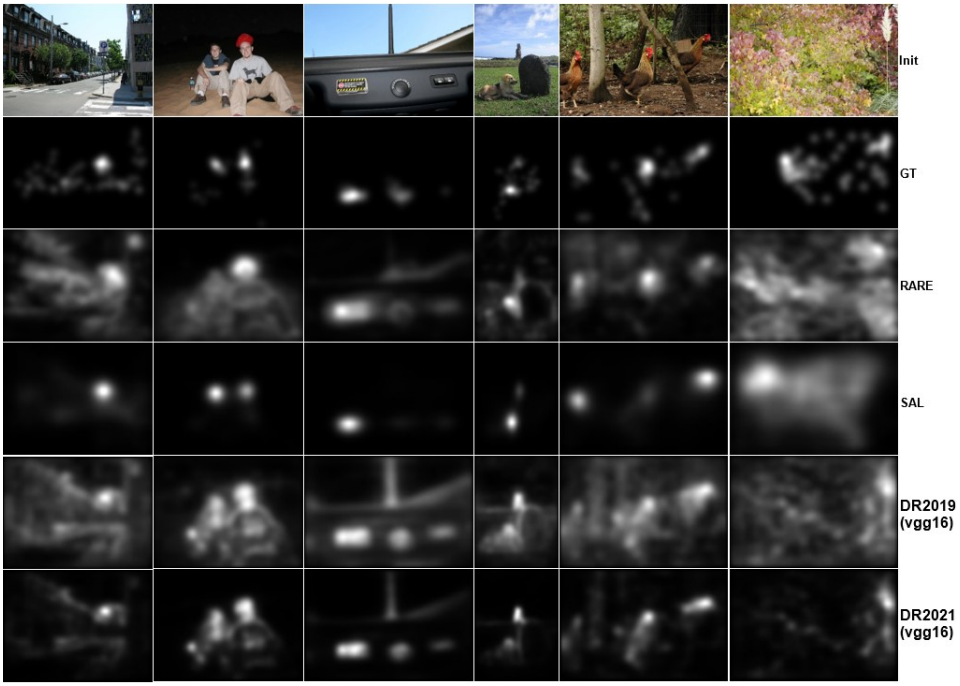


Figure 4.18. Selected samples MIT1003 dataset. From top to down : initial image, ground truth, RARE, SALICON, DR19, DR21.

4.3.5.2 Quantitative Validation on the Different Datasets

We compare our model with others on four datasets. First on MIT1003 and OSIE datasets which show general-purpose images where learning objects is very important. This dataset is basically one which should provide advantage to DNN-based models which focus on objects instead of salient information (faces, text, etc.). We previously showed in [33] that the DNN-based models mainly learn which objects are most of the time attended which leads to good results on images implying a high amount of top-down information while they are very bad in purely bottom-up models which is.

On the other side, we use P^3 dataset from [36] which shows synthetic psycho-physical images with pop-out bottom-up objects which should work better for classical saliency models and even more with **DR19** and **DR21** models.

Finally, we use O^3 dataset from [36] which also provides real life images but with odd-out-one regions. The dataset is somewhere in the middle between P^3 on one side and MIT1003 and OSIE dataset on the other side. The O^3 dataset should provide similar difficulty to classical and DNN-based saliency models.

1). MIT1003 dataset

We summarize in Table 4.9 the results of **DR19** and **DR21** and also results coming from [36] for ML-Net and SALICON models where ML-Net was trained with SALICON, P^3 and O^3 dataset and SALICON was trained with OSIE, P^3 and O^3 dataset. The idea is to avoid

Figure 4.19. Selected samples OSIE dataset. From top to down : initial image, ground truth, GBVS, SAM-ResNet, FAPTTX [33], DR19, DR21.

trainings of ML-NET or SALICON models on the MIT1003 dataset where it is evaluated. For other models (DeepFeat, eDN, GBVS, RARE2012, BMS, AWS), the figures come from [40].

For **DeepRare** the following variants are used : DR19-V16-WF (**DR19** with a VGG16 backbone and without using the faces layer), DR19-V16 (**DR19** with a VGG16 backbone and by using the faces layer), DR21-MN2 (**DR21** with a MobileNetV2 backbone and without using faces information), DR21-V16-WF (**DR21** with a VGG16 backbone and without using the faces layer), DR21-V16 (**DR21** with a VGG16 backbone and by using the faces layer), DR21-V19 (**DR21** with a VGG19 backbone and without using faces information). We also added to **DR21** using VGG16 the same top-down information (TD) than the one which was added to RARE2012 in [33] and called this model DR21-V16+TD.

The best model is definitely **DR21-V19** on all the metrics which is better than classical models but also than deep-features models (DeepFeat) and also all the DNN-based models in the Table 4.9. However, SALICON and ML-Net were trained on datasets which are different from the MIT1003 training set which makes their performances lower than if they were trained on images from MIT1003.

2). OSIE dataset

We summarize in Table 4.10 the results on the OSIE dataset. Here we added SAM-ResNet and FAPTTX models with the results reported in [33]. SAM-Resnet is used with its de-

Models	AUCJ ↑	AUCB ↑	CC ↑	KL ↓	NSS ↑	SIM ↑
DR21-V19	**0.86**	**0.85**	**0.56**	**0.88**	**1.93**	**0.50**
DR21-V16+TD	0.86	0.81	0.59	1.02	2.11	0.48
DR21-V16	0.84	0.83	0.50	1.19	1.81	0.43
DR21-V16-WF	0.84	0.83	0.49	1.16	1.75	0.42
DR21-MN2	0.84	0.83	0.50	1.14	1.71	0.42
DR19-V16	0.86	0.85	0.48	1.25	1.58	0.36
DR19-V16-WF	0.84	0.83	0.46	1.32	1.54	0.34
SALICON [25]	0.83	-	0.51	1.12	1.84	0.41
ML-Net [14]	0.82	-	0.46	1.36	1.64	0.35
DeepFeat [40]	0.86	0.83	0.44	1.41	-	-
eDN [68]	0.86	0.84	0.41	1.54	-	-
GBVS [22]	0.83	0.81	0.42	1.3	-	-
RARE2012 [55]	0.75	0.77	0.38	1.41	-	-
BMS [72]	0.75	0.77	0.36	1.45	-	-
AWS [19]	0.71	0.74	0.32	1.54	-	-

Table 4.9. Comparing results on the MIT1003 dataset. DR21 (VGG19 with faces), DR21-V16+TD, DR21 (VGG16 with faces), DR21 (VGG16 without faces), DR21 (MobileNet-V2), DR19 (VGG16 with faces), DR19 (VGG16 without faces), DeeFeat, eDN, GBVS, RARE2012, BMS, AWS results come from [40] and SALICON and ML-Net come from [36]. We added DR21 with VGG16 and top-down from [33] called DR21-V16+TD.

fault training parameters showing that when trained on general images without introducing datasets such as P^3 which can disturb the learning, modern DNN-based models are better than **DR21** in any version. FAPTTX also exhibits slightly better results showing the importance of top-down features in general images datasets. Our hypothesis here is that **DeepRare** models achieve better bottom-up scores than RARE2012 (verified on all datasets) but that the top-down information added to RARE2012 in FAPTTX makes it better. To verify this, we also added to **DR21** using VGG16 the same top-down information (TD) than the one which was added to RARE2012 in [33] and called this model DR21-V16+TD. This model is indeed better than FAPTTX proving that the top-down information is still missing from the **DeepRare** models.

The same idea is once again illustrated by the fact that **DR19-V16** is better (on both the MIT1003 and OSIE) than **DR19-V16-WF** even if the faces layer in VGG16 is much less efficient than a face detector as those used for the FAPTTX. This again shows that **DeepRare** models do not capture top-down information which let room for future improvements.

Another interesting point is that VGG16 backbone is slightly better for the OSIE dataset while VGG19 was better for MIT1003 showing that in MIT1003 maybe higher-level features are more important than in OSIE. A second point is about the fact that the FAPPTX model shows good results on this kind of images. The FAPPTX is built upon RARE2012 with additional

top-down features showing that adding top-down features to **DR21** would probably lead to results close to SAM-ResNet as **DR21** is better than RARE2012 in all configurations.

Models	AUCJ ↑	AUCB ↑	CC ↑	KL ↓	NSS ↑	SIM ↑
SAM-ResNet [13]	**0.90**	**-**	**0.77**	**1.37**	**3.1**	**0.65**
DR21-V16+TD	0.88	0.83	0.66	0.83	2.32	0.56
FAPTTX [33]	0.87	-	0.62	0.81	2.08	0.51
DR21-V16	0.87	0.86	0.59	0.91	2.06	0.52
DR21-V16-WF	0.87	0.86	0.58	0.84	2.01	0.51
DR21-V19	0.86	0.85	0.56	0.88	1.93	0.50
DR19-V16	0.87	0.86	0.55	0.98	1.75	0.44
DR19-V16-WF	0.86	0.86	0.53	1.01	1.66	0.43
DR21-MN2	0.85	0.84	0.51	1.06	1.55	0.42
DR21-V19	0.83	0.82	0.45	1.32	1.54	0.34
RARE2012 [55]	0.83	-	0.46	1.05	1.53	0.43
AWS [19]	0.82	-	0.45	1.11	2.02	0.42
GBVS [22]	0.81	-	0.43	1.08	1.34	0.42
AIM [4]	0.77	-	0.32	1.52	1.07	0.34

Table 4.10. comparing results on the OSIE dataset. DR21 (VGG16), DR21 (VGG16 without faces), DeepRare 2019 (VGG16), DR19 (VGG16 without faces), DR21 (MobileNet-V2), DR21 (VGG19), and SAM-ResNet, FAPTTX, RARE2012, AWS, GBVS, and AIM come from [36]. We added DR21 with VGG16 and top-down from [33] called DR21-V16+TD.

3). O^3 dataset

The O^3 dataset uses the MSR metric defined in [36]. When the MSR_t is higher, it is better as the target is well highlighted compared to the distractors. When MSR_b is lower, it is better, it means that the maximum of the saliency of the target is higher than the one of the background. The first measure will ensure that the target is visible compared to the distractors and the second that it is visible compared to the background.

Table 4.11 shows the MSR from [36] where we added the results from the **DeepRare** family (**DR19** and **DR21** in the versions using VGG16, VGG19, and MobileNetV2 architectures) splitting the dataset between the images where color is a good discriminator (Color) and the others (Non-color). All models work better for targets where color is an important feature and less well for non-color.

For MSR_t (higher is better) for Color **DR19** is less good especially compared to DNN-based models. However we can see that for Non-color images where the models fail much more **DR19** has a remarkable stability being second and very close the the best one (SAM-ResNet). **DR21** especially using the VGG19 and VGG16 architectures are definitely the best models

Model	Color		Non-color		All targets	
	$MSR_t \uparrow$	$MSR_b \downarrow$	$MSR_t \uparrow$	$MSR_b \downarrow$	$MSR_t \uparrow$	$MSR_b \downarrow$
DR21-V16	**1.66**	**0.74**	**1.31**	**1.31**	**1.45**	**1.01**
DR21-V19	1.63	0.78	1.29	1.39	1.43	1.13
DR21-MN2	1.19	1.02	1.06	1.54	1.12	1.32
DR19	1.14	0.75	1.00	1.00	1.06	0.89
SAM-ResNet [13]	1.47	1.46	1.04	1.84	1.40	1.52
CVS [18]	1.43	2.43	0.91	4.26	1.34	2.72
DGII [38]	1.32	1.55	0.94	1.95	1.26	1.62
FES [52]	1.34	2.53	0.81	5.93	1.26	3.08
ICF [39]	1.30	2.00	0.84	2.03	1.23	2.01
BMS [72]	1.29	0.97	0.87	1.59	1.22	1.07

Table 4.11. Comparing result between several models and DR family (DR19 and DR21 in the version VGG16, VGG19 and MobileNetV2). For MSR_t higher is better, For MSR_b lower is better.

being much better even than efficient DNN-based modes as SAM-ResNet on all the kinds of images.

If we take into account the MSR_b (lower is better), **DR19** clearly outperforms all the others providing the best discrimination between the target and the background. **DR19** is the only model with a MSR_b smaller than 1 which means that in average the maximum of the target saliency is higher than the maximum of the background saliency. **DR21** with VGG16 architecture is still better than all classical and DNN-based models and even better than **DR19** for Color images.

In conclusion, for MSR_t and MSR_b metrics, the models from the DeepRare family and especially **DR21** with VGG16 architecture outperform all the other models including efficient DNN-based models on both Color or Non-color images on O^3 dataset.

Table 4.12 shows the results of the **DeepRare** family compared to two other DNN-based models tested on the whole O^3 dataset (both Color and Non-color images). Our models outperform both SALICON and MLNet models on both MSR_t (all the **DeepRare** models are better) and MSR_b (**DR19** is better) metrics. According to [36], the results we show here for SALICON are the ones where it was trained on the OSIE by adding with P^3 and O^3 datasets. The ML-Net was trained on SALICON by adding with P^3 and O^3 datasets.

4). P^3 dataset

The P^3 dataset is the one which exhibits the less top-down information and it even does not have any centered bias. Naturally, for this dataset, the DNN-based models perform the worst. We will check here how **DeepRare** family models deal with the data.

Models	$MSR_t \uparrow$	$MSR_b \downarrow$
DR21-V16	**1.45**	**1.01**
DR21-V19	1.43	1.13
DR21-MN2	1.12	1.32
DR19	1.06	0.89
ML-Net [14]	0.96	0.91
SALICON [25]	0.90	1.26

Table 4.12. SALICON, ML-Net, DR19, and DR21 with MobileNetV2, VGG19, and VGG16 architectures) results on O^3 dataset.

First we use the average number of fixations (Avg. # fix.) and found percentage (% found) metrics. The average # of fixations is better if lower as it means that the target is found more rapidly and the found percentage metric is better if higher as it means that a higher percentage of the target is found after 100 fixations. Table 4.13 shows first the results on P^3 for **DeepRare** family models compared with SALICON and ML-Net models. For the SALICON and ML-Net models, they were trained the same way as in previous section (O^3 dataset). Our models all definitely outperform the two DNN-based models and need much less fixations to discover more of the targets showing here very good results.

Model	Avg. # fix. ↓	% found ↑
DR21-V16	**13.53**	**89**
DR21-V19	13.86	89
DR21-MN2	33.82	72
DR19	16.34	87
ML-Net [14]	42.00	44
SALICON [25]	49.37	65

Table 4.13. Comparing result on P^3 dataset: consider on average number of fixation and found percentage.

Table 4.14 provides more details about the found percentage metric after different numbers of fixations (15, 25, 50 and 100) and for specific images where the target is due to color, orientation or size features with 100 fixations. The results here are compared with classical models which are better in this dataset than DNN-based models. On this table, **DR21** model is the best again and especially **DR21** with the VGG16 architecture. While BMS can exhibit 100% for color or orientation target percentage found, it is more efficient in terms of detection to find the target (even if not its entire surface) very quickly (15 fixations) than to find all of the target surface but after 100 fixations. So if we look at the results after 15 fixations only the **DR21** method is all much better than the others.

Model	%fd15	%fd25	%fd50	%fd100	%fd-C	%fd-O	%fd-S
DR21-V16	**84.82**	**86.71**	**88.60**	**89.76**	**92.20**	**92.93**	**83.92**
DR21-V19	84.27	86.32	88.10	89.14	92.65	92.36	82.14
DR21-MN2	61.37	64.81	69.37	72.46	77.17	71.75	68.21
DR19	80.61	83.27	86.63	87.87	91.29	89.58	82.50
RARE2012 [55]	59.87	63.52	79.75	93.48	99.54	90.26	88.53
BMS [72]	58.94	66.37	83.56	95.14	**100**	**100**	82.76
ICF [39]	32.63	41.38	68.47	70.18	69.41	**100**	42.45
oSALICON [65]	30.25	39.75	55.45	78.53	76.35	81.58	70.42

Table 4.14. Comparing result on P^3 dataset. Details on the percentage found after the number of fixation of 15, 25, 50, and 100. Percentage found of the color (-C), orientation (-O), and size (-S) features taken separately.

Figure 4.20 shows the **DR21** model compared to the best classical model (IMSIG) and the best DNN-based model (SALICON). If we look at the percentage of targets found after only 15 fixations, than **DR21** with the VGG16 and VGG19 architectures are the best followed by **DR19**, IMSIG and SALICON which is definitely worse. According to [36], for the OpenSALICON (oSALICON), the saliency maps are obtained using the pre-trained OpenSALICON weights on the SALICON dataset. In that way oSALICON is not trained on the P^3 dataset again to remain fair.

Finally, the Global Saliency Index (GSI) metric is computed on this dataset. This score is better when higher as it measures how target average saliency is distinguished from the distractors. For GSI, table 4.15 shows the average figures for the whole dataset (GSI-Avg) and for each of the dataset classes : images where the feature of the target is based on color (GSI-Color), on the orientation (GSI-Orientation) and on size (GSI-Size). The average scores for the GSI metric are much higher for the DeepRare family and especially for **DR21** with the VGG16 architecture. While the results of classical models such as BMS or RARE2012 can be comparable on GSI-Color, for GSI-Orientation or GSI-Size they are much less good than those of the DeepRare family. If we take into account the DNN-based models, than the GSI scores begin to be even negative, showing that distractors are in average more visible than the salient areas.

Figure 4.21, Fig. 4.22, and Fig. 4.23 let us compare the dynamics of the GSI scores on the three classes of models (GSI-Color, GSI-Orientation and GSI-Size). For each figure, we show the three best classical models with the three best DNN-based models (name in bold) on the left and DeepRare family results with the best classical and the best DNN-based model on the right.

For color targets (see Fig. 4.21, right graph) we see that the maximum of GSI score for **DR21** with a VGG16 architecture where GSI is at more than 0.9. If RARE2012 model is better on small target/distractor color difference, **DR21** is better for larger differences. The ICF model is less good than all the other models from the DeepRare family on any

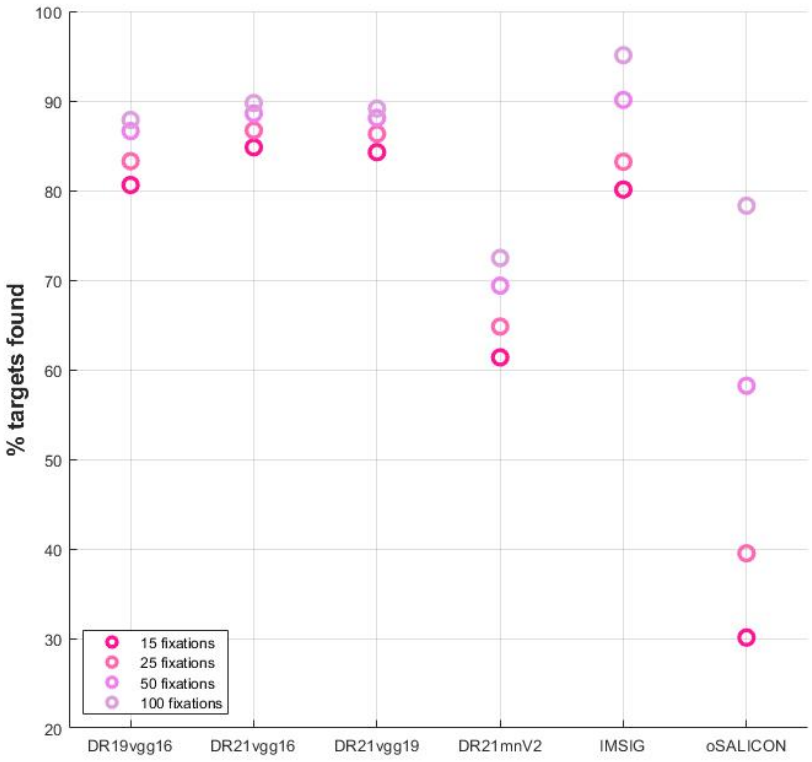


Figure 4.20. Number of fixations (horizontal axis) vs. % of targets detected (vertical axis). It is chosen on 15, 25, 50, and 100 fixations.

Model	GSI-Color	GSI-Orientation	GSI-Size	GSI-Avg.
DR21-V16	**0.77**	**0.50**	**0.49**	**0.59**
DR21-V19	0.75	0.49	0.51	0.58
DR21-MN2	0.66	0.42	0.51	0.53
DR19	0.42	0.17	0.15	0.25
RARE2012 [55]	0.74	0.01	0.18	0.31
BMS [72]	0.72	0.01	-0.02	0.24
ICF [39]	0.18	-0.02	-0.51	-0.12
oSALICON [65]	-0.01	0.04	-0.11	-0.03

Table 4.15. Comparing result on P^3 dataset. Global Saliency Index score on color, orientation, and size features, and average score from these 3 features.

target/distractor color difference. We also see that **DR21** model is better than **DR19** model for all used architectures.

In addition, the shape of the GSI curve exhibited by the DeepRare family of models is coherent from a biological point of view: if the difference between the target color and the distractor color is small, then the model detects less well the target (left-side of the curve) than when the color of the target and background is very different (right-side of the curve). The models from the DeepRare family are the only ones to provide a biologically plausible GSI curve.

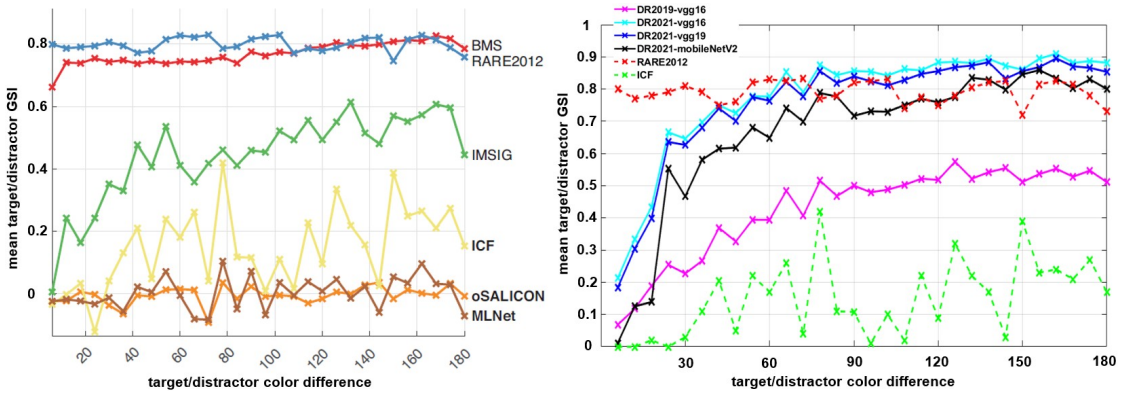


Figure 4.21. The GSI score for color target/distractor difference. Left plot: generated by [36]. Right plot: several classical and deep learning models including DR21 model.

For orientation targets (see Fig. 4.22, right graph) we see that the maximum of GSI score for **DR21** with a VGG16 architecture is at more than 0.6 (right graph). This score is drastically higher than the best DNN-based model and the best classical model on all target/distractor orientation difference. We also remark again that **DR21** model is better than **DR19** for all used architectures.

Also, the shape of the GSI curve exhibited by DeepRare family models is again coherent from a biological point of view: if the difference between the target orientation and the distractor orientation is small (left-side of the curve), then the model detects the target less well than when target orientation is very different from the distractors (right-side of the curve). Here also only the DeepRare family models have a dynamic which is close to the one expected from a human.

For size targets (see Fig. 4.23, right graph) we see that the maximum of GSI score for the best model (**DR21** with a MobileNetV2 architecture) is about 0.7 which makes it close to RARE2012 in terms of maximum GSI. The best classical model (SSR) is less good when the target/distractor size ratio is smaller or bigger (left-side or right-side of the curve) but better when this ratio is close to 1 where there is a small difference between the target and the distractors (center of the graph). The best (here eDN) is much worse than the DeepRare family models on any target/distractor ratio. **DR21** with any architecture is again much better here than **DR19**.

The shape of the GSI curve exhibited by our model is finally again coherent from a biological point of view: if the difference between the target size and the distractor size is small (center of the curve), then the model detects the target less well than when its size is very different (left and right sides of the curve). We can also see an asymmetry in the curve showing that it is easier for **DR19** to detect target twice bigger than distractors than targets twice smaller

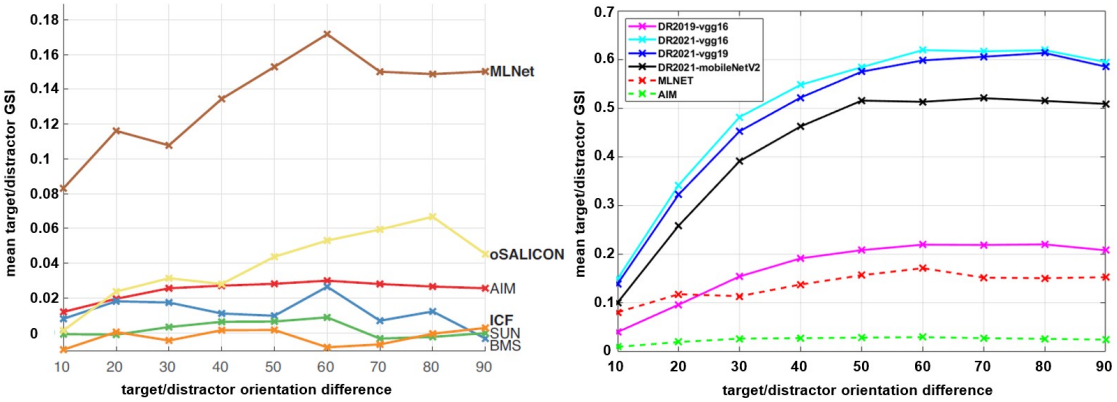


Figure 4.22. The GSI score for orientation target/distractor difference. Left plot: generated by [36]. Right plot: several classical and deep learning models including DR21 model.

than the distractors which is again biologically coherent. This is also true for **DR21** even if for very big target size (2 times bigger than the distractors) we can see a decrease in the performance.

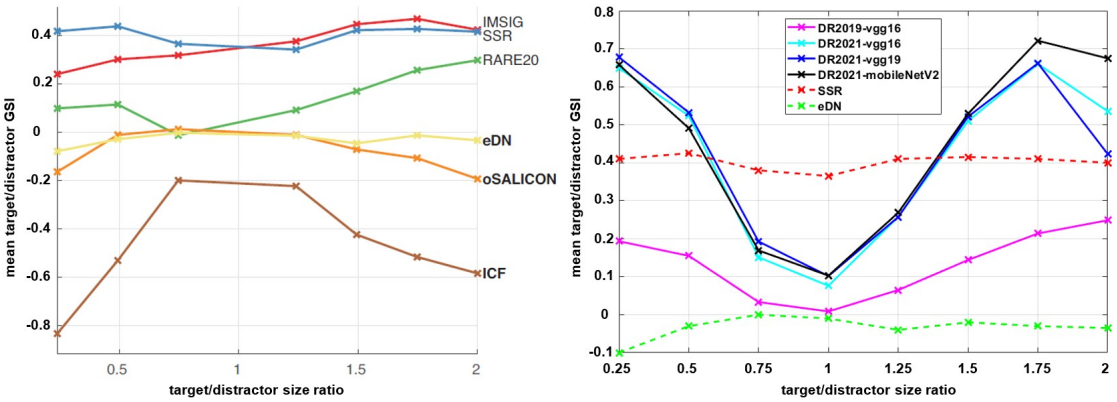


Figure 4.23. The GSI score for size target/distractor ratio. Left plot: generated by [36]. Right plot: several classical and deep learning models including DR21 model.

4.3.6 Discussion and Conclusion

We proposed a novel saliency model called DeepRare using the rarity idea of [55] applied on the deep features extracted by a deep neural network pretrained on ImageNet dataset. This exhibits several interesting features:

- It needs no training and the default ImageNet training is enough.
- The model is computationally efficient and is easy to run on CPU.
- Our approach is very modular, and it is very easy to adapt to any neural network architecture such as VGG16, VGG19, or even more complex architectures such as MobileNetV2 for adaptation on mobile devices as smart phones or for edge computing.

- It is possible to check each layer contribution and thus better understand the result. It is also possible to check several thresholds to see which areas of the images are considered as the most rare compared to the others and at which levels. This opportunity is a key feature of **DeepRare2021** contrary to black-box DNN-based models.
- The **DeepRare2021** model is very generic and stable through all kinds of different datasets where other models are sometimes better but only for one dataset and/or a specific metric but much worse for the others. The **DeepRare2021** version is specifically better than **DeepRare2019** on all datasets when compared with the same VGG16 architecture.

We show that this framework, especially **DeepRare2021**, is the most stable and generic when testing it on 4 very different datasets. It was first tested on MIT1003 and OSIE where it outperforms all the classical models and most of the DNN-based models. However some DNN-based models, especially the latest ones, still provide better results.

We then tested **DeepRare2021** model on the O^3 dataset, where the **DeepRare2021** outperforms all the models on target/background discrimination and on target/distractor discrimination. Finally, on P^3 dataset, our model is first for the target discrimination based on the number of fixations. When computing the average GSI metric our model is also the best for all the features (color, orientation, size) and the only one to exhibit a GSI plot which is biologically plausible.

While one cannot expect from an unsupervised model such as **DeepRare2021** model to be better on MIT1003 or OSIE dataset than DNN-based models which are trained and tuned on similar data, those DNN-based models are bad or even completely lost on O^3 and P^3 datasets.

Our tests show that **DeepRare**DeepRare family models and especially **DeepRare2021** models are optimized models overcoming any classical model and being only beaten by recent DNN-based models on MIT1003 or OSIE datasets. They are generic, unsupervised and stable in results on all kind of datasets. Even if they take into account low- and high-level features, they still remain bottom-up approaches as FAPTTX results show [33]. Indeed by adding top-down information to RARE2012, the results of FAPTTX are still comparable or a little better than for the **DeepRare2021** model. However if we add the same top-down information as in [33] to **DeepRare2021** instead of RARE2012, **DeepRare2021** with top-down outperforms RARE2012 with top-down information. The fact that top-down information is important can also be seen with the fact that DR21-V16 is most of the time better than DR21-V16-WF because it uses information about faces.

This remark leads to future works for future implementations of the **DeepRare2021** model. Adding top-down information on top of **DeepRare2021** would probably drastically improve its performance on the MIT1003 and OSIE dataset while keeping similar results on O^3 and on P^3 datasets.

4.4 In Brief

The summaries of this chapter are:

- **DeepRare2019** model: We proposed a novel saliency model called **DeepRare2019** using the rarity idea of [55] applied on the deep features extracted by a VGG16 network pretrained on ImageNet dataset. This model needs no training and is computationally efficient. We show that this model is definitely the most stable and generic when testing it on 3 very different datasets such as the MIT1003, the O^3, and the P^3. When computing the average GSI metric our model is the only one to be in the top-three for all the features (color, orientation, size) and the only one to exhibit a GSI plot which is biologically plausible. Our approach is very modular, and it still has room for improvement using VGG16 but also other networks such as ResNET50 for example. It is also possible to check each layer contribution and thus better understand the result contrary to black-box DNN-based models. All those advantages show that feature-engineered models might make their come back in visual attention field.
- **DeepRare2021** model: We upgrade our saliency model from the **DeepRare2019** and is called **DeepRare2021**. We conduct more experiments on other two neural networks (beside the VGG16): VGG19 and MobileNetV2 and test on one more dataset: the OSIE. While one cannot expect from an unsupervised model such as **DeepRare2021** model to be better on MIT1003 or OSIE dataset than DNN-based models which are trained and tuned on similar data, those DNN-based models are bad or even completely lost on O^3 and P^3 datasets. The most significant point is that the **DeepRare2021** with VGG16 architecture outperforms all the other models including efficient DNN-based models on both Color or Non-color images on O^3 dataset. The **DeepRare** family framework shows that deep-features-engineered models might become a good choice in visual attention field especially when the 1) images they are applied on are special and specific and 2) eye-tracking datasets are not available on this kind of images or when 3) explaining the result is of high importance for example the case of industrial standardization.

Chapter 5

Conclusions

5.1 Contributions

This thesis presented several advances in the field of visual saliency. Solutions proposed in this work were shown to provide some improvements in computational visual attention modeling based-on top-down approach and validation compared to existing state-of-the-art techniques both traditional and deep learning methods. The original contributions of this thesis are the following:

- **Bottom-up attention maps with text detection:** We have proposed a model which consists of three combination between bottom-up and top-down information. First, we just add the bottom-up information (using the RARE model) with the top-down information (using the traditional text detection model). This step provides us with minor significant result because it just improve the saliency map a bit. Second, we multiply the bottom-up information (saliency map) with text detection. This result is better than the previous one, however, it shows the top-down information more than bottom-up information. It means that it makes some important bottom-up information disappear. Finally, we use the weight factor for the bottom-up and top-down information. As the result, the weight factor provides us with good combination result compared to those two previous steps. Due to the facts that this was just the first step our research, many other activities and experiments are required to reach our research goals. Thus, we must conduct more experiments using both saliency map and text detection features by considering on the weight factor.
- **Bottom-up attention maps with fact and text detection:** Several simple algorithms are made which can easily add to bottom-up saliency maps. However, detection result using a general purposed object detector may include too many objects which are less likely to attract visual attention. For generating both face and text features, we used the state-of-the-art traditional face and text detector models. The size of the top-down object is very important. This was more the case with text where the difference in terms of eye fixations between big text (titles) and small text (description) is very important. The result was polluted by small text which really decrease a lot the overall text importance. In addition to that, traditional models can have a behavior which can be explained while DNNs provide results without letting any chance to the programmers to explain why exactly their model works well or not. Moreover, we used the OSIE dataset to test our model. After conducting the experiment, we show how to simply add top-down information to any bottom-up saliency models in a generic way.

- **Bottom-up attention maps with fact, text, and object detection:** The purpose of this contribution is to understand the differences in visual attention computation between traditional bottom-up saliency models and DNN-based saliency models. It also focuses on the relative importance of bottom-up and top-down information. In addition, our results show that the influence of the main objects in images is the following: 1) face detection is the most important, 2) text detection is about half of the importance of face detection, 3) animal detection is about half less important than text detection. Furthermore, the influence of person and transportation detection is marginal or even negative because the viewer gazes probably focus on small parts of persons or cars but not everywhere on their bounding boxes. This means that the bottom-up information still remains important and should not be neglected in visual attention, especially in the complex and crowded images where it is hard to identify faces. Finally, we showed that mixing a bottom-up model with our naive top-down information framework leads on the MIT300 saliency benchmark to the best results among all bottom-up models and overtakes number of DNN-based models especially on KLD which measures the probability distribution resemblance with eye-tracking. The important points to evaluate our result are as following: 1) we generate results by using our own model and the MIT300 dataset, 2) then we send our results to the MIT benchmark team because they don't provide us with the fixation maps on the MIT300 dataset, 3) finally, we compare our results to others on MIT website.
- **Building a new Retail dataset:** A new bottom-up dataset is made by taking photos from supermarkets. This new dataset can improve the saliency maps while there is no top-down information. Firstly, we retrain the SAM-ResNet network to maintain the existing weight from this network by loading the original weight. Then we add new weight from our new dataset (200 images) as bottom-up information. We used the original weights of SALICON 2015 and 2017. After that, we also train this network on the CAT2000 dataset. Finally, we compare our results to those original weighs, and we evaluate our result by using only three metrics such as CC, KLD, and NSS. Moreover, we did experiment on 14 people from eye-tracker to get the groundtruth fixation map for each image. This experiment shows that deep learning models are good for generating saliency maps, but it neglects bottom-up information. It also shows that by adding bottom-up information to deep learning models, it can improve saliency maps compared to the fixation maps.
- **DeepRare family models (DR19 and DR21):** We have proposed a novel saliency model called **DeepRare** (**DR**) using the rarity idea of [55] applied on the deep features extracted by a VGG16 network [58] pretrained on ImageNet dataset. The **DR** mixes the deep feature extraction advantages and the feature-engineered advantages giving a generic and explainable model. It shows good results whatever the dataset is and considers both low-level and high-level features. DR does not need any training and only uses the default ImageNET training, and it is computationally efficient. When trained on a general dataset such as ImageNET, the network will extract a complete set of features that one finds in images at several scales (from very low-level in the first layers to very high level in the last ones). We decide here to use a VGG16 with its default training on ImageNET dataset as a feature extractor. In our implementation, we use the Keras framework to extract any layer and feature map within this layer. We do not use the pooling layers (as they are redundant with the previous convolutional layer) and

the final fully connected classification layers. In a VGG16, the convolutional layers are gathered within 5 groups separated by the pooling layers : 1) the first low-level features in layers 1 and 2, then 2) second set of low-level features from layers 4 and 5, after that 3) the first middle-level layers 7, 8 and 9 and 4) the second middle-level layers 11, 12 and 13 and finally 5) the high-level features from layers 15, 16 and 17. On each feature map within the layers we compute the data rarity. A very simple rarity function based on the histogram of each feature map sampled on 11 bins. Once the rarity histogram is computed, the resulting rarity image is reconstructed by backprojection. This image will highlight pixels in the feature map which are rare compared to the other pixels in the feature map. Once the rarity of all feature maps is computed, the results need to be fused together. In a second stage, the same fusion method is applied for each of the 5-layer groups arriving to 5 deep groups conspicuity maps (DGCM). Finally, the 5 DGCM are summed up and a top-down face map is added. This face map is the feature map 105 from layer 15 which is known to detect faces. Moreover, DR is very modular and still having room for improvement (also RestNet50). When we tested on 4 very different datasets (MIT1003, OSIE, P^3, O^3), they provide us with the best genericity and stability in all circumstances compared to other models.

5.2 Perspectives

We would like to do more experiments by using other datasets such as MIT300, CAT2000, and so on and applying some other deep learning frameworks such as InceptionV3, ResNet50, and so on. Our main point of view is to make our model in generic way. Since the deep learning models can provide us with good results of visual attention by mainly focusing on top-down information. In addition, for further research on this model it can apply to evaluate the attention region of visual designs such as advertisement paper or poster, web page images, supermarket items, and so on. It means that our model can analyze and give feedback to those visual designs whether they can provide people with enough attention or not. For better results, we can do more experiments by building some datasets as the ground truth for those designs or images using eye-tracking equipment. Then we can compare those datasets to the results which are produced by our model. Another interesting point, our model can make an application on quality assessment such as fruit, clothes, shoes, and so on. For short term project, it just create an application on smart phones to detect an abnormal or surprising information. For long term, it can apply to the camera surveillance for automatic detected that information, and it can be used in the factories or the manufacturing places.

Bibliography

[1] Tamar Avraham and Michael Lindenbaum. Esaliency (extended saliency): Meaningful attention using stochastic image modeling. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):693–708, 2010.

[2] Ali Borji. Boosting bottom-up and top-down visual features for saliency estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 438–445. IEEE, 2012.

[3] Donald E. Broadbent. Perception and communication. *Pergamon Press*, 1958.

[4] Neil Bruce and John Tsotsos. Attention based on information maximization. *Journal of Vision*, 7(9):950–950, 2007.

[5] Neil DB Bruce and John K Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of vision*, 9(3):5–5, 2009.

[6] N. J. Butko and J. R. Movellan. Optimal scanning for faster object detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758, June 2009.

[7] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark, 2015.

[8] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[9] Moran Cerf, Jonathan Harel, Wolfgang Einhäuser, and Christof Koch. Predicting human gaze using low-level saliency combined with face detection. In *Advances in neural information processing systems*, pages 241–248, 2008.

[10] Kai-Yueh Chang, Tyng-Luh Liu, Hwann-Tzong Chen, and Shang-Hong Lai. Fusing generic objectness and visual saliency for salient object detection. In *Proceedings of the 2011 International Conference on Computer Vision*, pages 914–921. IEEE Computer Society, 2011.

[11] Chuanbo Chen, He Tang, Zehua Lyu, Hu Liang, Jun Shang, and Mudar Sarem. Saliency modeling via outlier detection. *Journal of Electronic Imaging*, 23, 09 2014.

[12] Francois Chollet et al. Keras, 2015.

[13] 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):5142–5154, Oct 2018.

[14] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*, 2016.

[15] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] Krista A. Ehinger, Barbara Hidalgo-sotelo, Antonio Torralba, and Aude Oliva. Modeling search for people in 900 scenes: A combined source model of eye guidance. *Visual Cognition*, 2009.

[18] Erkut Erdem and Aykut Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, 13(4):11–11, 03 2013.

[19] Anton Garcia-Diaz, Victor Leboran, Xose R Fdez-Vidal, and Xose M Pardo. On the relationship between optical variability, visual saliency, and eye fixations: A computational approach. *Journal of vision*, 12(6):17–17, 2012.

[20] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Transactions on Image Processing*, 19(1):185–198, 2010.

[21] Junwei Han, Dingwen Zhang, Xintao Hu, Kaiming Li, Jinchang Ren, and Feng Wu. Background prior based salient object detection via deep reconstruction residual. *IEEE Transactions on Circuits and Systems for Video Technology*, 25:1–1, 01 2014.

[22] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2007.

[23] Xiaodi Hou, Jonathan Harel, and Christof Koch. Image signature: Highlighting sparse salient regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(1):194–201, 2012.

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

[25] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 262–270, 2015.

[26] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004.

[27] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 40(10-12):1489–1506, 2000.

[28] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[29] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. In *MIT Technical Report*, pages 1–22, 2012.

[30] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th international conference on*, pages 2106–2113. IEEE, 2009.

[31] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[32] Wolf Kienzle, Felix A Wichmann, Matthias O Franz, and Bernhard Schölkopf. A non-parametric approach to bottom-up visual saliency. In *Advances in neural information processing systems*, pages 689–696, 2007.

[33] P. Kong, M. Mancas, N. Thuon, S. Kheang, and B. Gosselin. Do deep-learning saliency models really model saliency? In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2331–2335, Oct 2018.

[34] Phutphalla Kong, Matei Mancas, Seng Kheang, and Bernard Gosselin. Saliency and object detection. In *2018 International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI)*, pages 523–528. CENPARMI, 2018.

[35] Gert Kootstra, Arco Nederveen, and Bart De Boer. Paying attention to symmetry. In *British Machine Vision Conference (BMVC2008)*, pages 1115–1125. The British Machine Vision Association and Society for Pattern Recognition, 2008.

[36] I. Kotseruba, C. Wloka, A. Rasouli, and J. Tsotsos. Do saliency models detect odd-one-out targets? new datasets and evaluations. *arXiv preprint arXiv:2005.06583*, 2020.

[37] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint arXiv:1411.1045*, 2014.

[38] Matthias Kümmerer, Thomas S. A. Wallis, and Matthias Bethge. Deepgaze II: reading fixations from deep features trained on object recognition. *CoRR*, abs/1610.01563, 2016.

[39] Matthias Kümmerer, Thomas SA Wallis, Leon A Gatys, and Matthias Bethge. Understanding low-and high-level contributions to fixation prediction. In *2017 IEEE International Conference on Computer Vision*, pages 4799–4808, 2017.

[40] Ali Mahdi and Jun Qin. Deepfeat: A bottom up and top down saliency model based on deep features of convolutional neural nets. *IEEE Transactions on Cognitive and Developmental Systems*, PP, 09 2017.

[41] Atsuto Maki, Peter Nordlund, and J.-O Eklundh. Attentional scene segmentation: Integrating depth and motion. *Computer Vision and Image Understanding*, 78:351–373, 06 2000.

[42] Matei Mancas. Relative influence of bottom-up and top-down attention. In *International Workshop on Attention in Cognitive Systems*, pages 212–226. Springer, 2008.

[43] Matei Mancas, Vincent P Ferrera, Nicolas Riche, and John G Taylor. *From human attention to computational attention*, volume 2. Springer, 2016.

[44] Matei Mancas, Phutphalla Kong, and Bernard Gosselin. Visual attention: Deep rare features. In *2020 Joint 9th International Conference on Informatics, Electronics Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, pages 1–6, 2020.

[45] Matei Mancas and Olivier Le Meur. Applications of saliency models. In *From Human Attention to Computational Attention*, pages 331–377. Springer, 2016.

[46] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2232–2239, 2009.

[47] AJAY K. MISHRA and YIANNIS ALOIMONOS. Active segmentation. *International Journal of Humanoid Robotics*, 06(03):361–386, 2009.

[48] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3538–3545. IEEE, 2012.

[49] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[50] Jun Qin and Lin Xu. Data acquisition and digital instrumentation engineering modelling for intelligent learning and recognition. *Biosensors Journal*, 4:1–4, 03 2015.

[51] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*,

2017.

[52] Hamed Rezazadegan Tavakoli, Esa Rahtu, and Janne Heikkil. Fast and efficient saliency detection using sparse sampling and kernel density estimation. volume 6688, pages 666–675, 05 2011.

[53] Nicolas Riche. Study of parameters affecting visual saliency assessment. In *From Human Attention to Computational Attention*, pages 227–243. Springer, 2016.

[54] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and human fixations: state-of-the-art and study of comparison metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 1153–1160, 2013.

[55] Nicolas Riche, Matei Mancas, Matthieu Duvinage, Makiese Mibulumukini, Bernard Gosselin, and Thierry Dutoit. Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. *Signal Processing: Image Communication*, 28(6):642–658, 2013.

[56] Ruth Rosenholtz. A simple saliency model predicts a number of motion popout phenomena. *Vision research*, 39(19):3157–3163, 1999.

[57] C. Wloka S.A. Yoo R. Sengupta and J. Tsotsos. Psychophysical evaluation of saliency algorithms. *Journal of Vision*, 16(12):1291–1291, 2016.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[59] Bongwon Suh, Haibin Ling, Benjamin B. Bederson, and David W. Jacobs. Automatic thumbnail cropping and its effectiveness. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, pages 95–104, 2003.

[60] Pengfei Sun and Jun Qin. Enhanced factored three-way restricted boltzmann machines for speech detection. *arXiv preprint arXiv:1611.00326*, 2016.

[61] Pengfei Sun and Jun Qin. Neural networks based eeg-speech models, 2017.

[62] X. Sun. Semantic and contrast-aware saliency. *arXiv preprint arXiv:1811.03736*, 2018.

[63] Dlib team. Dlib c++ library using hog and cnn. http://dlib.net. Accessed: 2017-06-10.

[64] ICDAR team. Icdar 2013. http://www.icdar2013.org. Accessed: 2016-12-05.

[65] Christopher Lee Thomas. Opensalicon: An open source implementation of the salicon saliency model. Technical Report TR-2016-02, University of Pittsburgh, 2016.

[66] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image

with connectionist text proposal network. In *European conference on computer vision*, pages 56–72. Springer, 2016.

[67] Eleonora Vig, Michael Dorr, and David 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*, pages 2798–2805, 2014.

[68] Eleonora Vig, Michael Dorr, and David Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. pages 2798–2805, 06 2014.

[69] Juan Xu, Ming Jiang, Shuo Wang, Mohan S Kankanhalli, and Qi Zhao. Predicting human gaze beyond pixels. *Journal of vision*, 14(1):28–28, 2014.

[70] Juan Xu, Ming Jiang, Shuo Wang, Mohan S. Kankanhalli, and Qi Zhao. Predicting human gaze beyond pixels. *Journal of Vision*, 14(1):28–28, 01 2014.

[71] Jimei Yang and Ming-Hsuan Yang. Top-down visual saliency via joint crf and dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):576–588, 2017.

[72] J. Zhang and S. Sclaroff. Saliency detection: A boolean map approach. In *2013 IEEE International Conference on Computer Vision*, pages 153–160, Dec 2013.

[73] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1274, 2015.