



*electronics*



Article

---

# DeepRare: Generic Unsupervised Visual Attention Models

---

Phutphalla Kong, Matei Mancas, Bernard Gosselin and Kimtho Po

Special Issue

Important Features Selection in Deep Neural Networks

Edited by




Dr. Matei Mancas, Dr. Jean-Benoit Delbrouck and Dr. Sidi Ahmed Mahmoudi



<https://doi.org/10.3390/electronics11111696>

## Article

# DeepRare: Generic Unsupervised Visual Attention Models

Phutphalla Kong <sup>1,2,\*</sup> , Matei Mancas <sup>2,\*</sup> , Bernard Gosselin <sup>2</sup>  and Kimtho Po <sup>1</sup>

<sup>1</sup> Institute of Technology of Cambodia (ITC), Russian Conf. Blvd., Phnom Penh P.O. Box 86, Cambodia; kimtho@itc.edu.kh

<sup>2</sup> Numediart Institute, University of Mons (UMONS), 31, Bd. Dolez, 7000 Mons, Belgium; bernard.gosselin@umons.ac.be

\* Correspondence: phutphalla@itc.edu.kh (P.K.); matei.mancas@umons.ac.be (M.M.)

† These authors contributed equally to this work.

**Abstract:** Visual attention selects data considered as “interesting” by humans, and it is modeled in the field of engineering by feature-engineered methods finding contrasted/surprising/unusual image data. Deep learning drastically improved the models efficiency on the main benchmark datasets. However, Deep Neural Networks-based (DNN-based) models are counterintuitive: surprising or unusual data are by definition difficult to learn because of their low occurrence probability. In reality, DNN-based 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 article, we propose a new family of visual attention models called DeepRare and especially 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 this common framework. **DR21** (1) does not need any additional training other than the default ImageNet training, (2) is fast even on CPU, (3) is tested on four very different eye-tracking datasets showing that **DR21** is generic and is always 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.

**Keywords:** eye tracking; deep features; odd one out; rarity; saliency; visual attention prediction; visibility



**Citation:** Kong, P.; Mancas, M.; Gosselin, B.; Po, K. DeepRare: Generic Unsupervised Visual Attention Models. *Electronics* **2022**, *11*, 1696. <https://doi.org/10.3390/electronics11111696>

Academic Editor: George A. Papakostas

Received: 21 March 2022

Accepted: 5 May 2022

Published: 26 May 2022

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Visual Attention: Deep Learning Trouble

The human visual system (HVS) [1] handles a huge quantity of incoming visual information, and it cannot carry out multiple complex tasks at the same time in the whole visual field. This bottleneck [2] implies that it has an exceptional ability of sampling the surrounding world and paying attention to objects of interest. In computer vision, visual attention is mainly modeled through the so-called saliency maps. The modeling of visual attention has numerous applications such as object detection, image segmentation, image/video compression, robotics, image re-targeting, visual marketing, and so on [3]. Visual attention is considered to be a mix of bottom-up and top-down information. Bottom-up information is based on low-level features such as luminance, chrominance, or texture. Top-down information is more related to knowledge people already have about their tasks or objects they see such as faces, text, persons, or animals. In this paper, we do not focus on the use of the saliency maps, which might help in deep neural network architectures to enhance their task results but on saliency maps which intend to mimic human attention with no specific task. It is about human general attention and a visual efficiency for humans: is the information shown in an optimal way to humans?

Since the early 2000s [4], numerous models of visual attention based on image features were provided. In this paper, they will be referred to as **classical models**. While they can be very different in their implementation, 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 the seminal work of [5] or [6], but also more recent work based on information processing such as AIM [7]. Finally, some models became a reference for classical models such as GBVS [8], RARE [9], BMS [10], or AWS [11].

With the arrival of the deep learning wave, most researchers have focused on Deep Neural Networks saliency, which will be referred to as **DNN-based** in this paper. DNN-based models triggered a revolution in terms of results on the main benchmark datasets such as the MIT benchmark [12], where DNN-based saliency models definitely outperformed classical models. The DNN-based models have already been used in several applications such as image and video processing, medical signal processing, or big data analysis [13–17]. Some of the DNN-based models became new references such as SALICON [18], MLNet [19], or SAM-ResNet [20]. Since then, other novel models made additional progress such as TranSalNet [21], MSI-NET [22], SalFBNNet [23], or DeepGazeII [24]. New models also begin to use a common framework for images or video like UNISAL [25].

However, recently DNN-based models have been criticized for some drawbacks. First, they underestimate the importance of bottom-up attention [26], which indicates that they were mostly trained to detect the attractive top-down objects rather than detect saliency itself. In [27], 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 and thus, by definition, difficult to learn. This shows that saliency cannot be learnt but instead objects [28] which are often attended by human gaze (such as faces, text, bodies, etc.) are learned, and, furthermore, they are enough to provide good results on the main benchmarks.

A second drawback of the DNN-based models is that, in addition to not taking into account low-level features' surprise levels, DNN-based models are not generic enough to adapt to new images, which are different enough from the training dataset. Indeed, recently, Ref. [29] 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 the MIT dataset on natural images, their results drastically drop on  $P^3$  and  $O^3$ .

A third drawback of the DNN-based models is linked to DNNs themselves, which are black boxes. When a model fails to predict saliency, there is no way to understand why this prediction failed.

Parallel to DNN-based models, DeepFeat [30] or SCAFI [31] deal with models where pre-trained deep features are used. Those models will be called **deep-features models** in this article. However, they are not yet comparable to DNN-based models for general images datasets such as the MIT benchmark. Based on the new datasets in [29], DeepRare2019 [32] provides a new deep-feature saliency model by mixing deep features and the philosophy of an existing classical model [9]. This model is efficient on all the datasets, with no need for any additional training and efficient in terms of computation even on CPU.

In this article, we build on DeepRare2019 to improve it in several ways: (1) different DNN architectures are used and compared (VGG16, VGG19, and MobileNetV2) on more datasets, (2) a threshold on the feature rarity is introduced, which lets us understand which parts of the image are the most surprising at different levels providing transparency to the model, and (3) the best combination of thresholds and an improved post-processing, which lead to results that are much better than for DeepRare2019. This new model is called DeepRare2021 (**DR21**) and shows that the DeepRare framework is modular and can easily evolve.

In Section 2, **DR21** is described and the threshold on feature rarity is used to show how the DNN features' rarity can become explainable. In Section 3, this model is tested

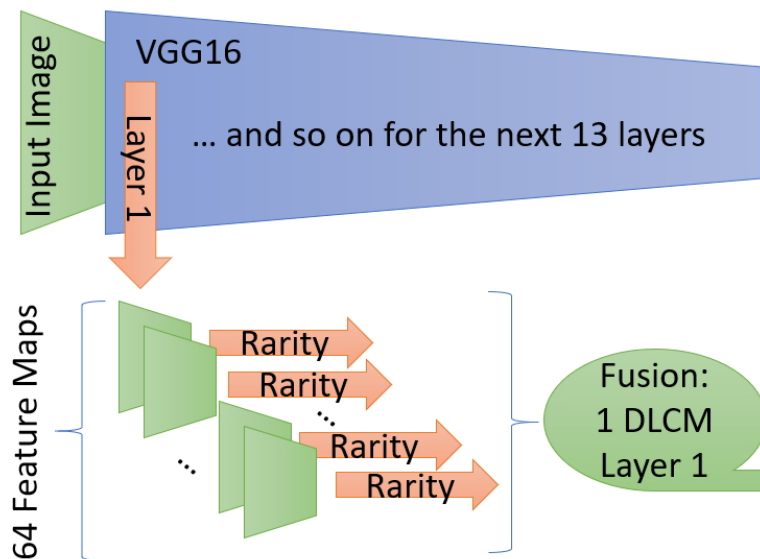
on the datasets proposed in [29] but also on an additional dataset. We finally discuss and conclude on the pertinence of the resurgence of the feature engineering models.

## 2. DeepRare2021 Model: Digging into Rare Deep Features

In this article, we extend the approach in [32] where a framework called DeepRare is proposed which mixes the simplicity of the idea of rarity computation to find the most salient features with the advantages of deep features extraction. Indeed, rare features attract human attention as they are surprising compared to the other features within the image. This combination has the advantage of being fast (less than 1 s per image on CPU with a VGG16 feature extractor) and easily able to adapt to any default DNN architectures (VGG19, ResNet, etc.). Here, we extend **DR19** adding the possibility to have thresholds on the rarity maps and also the possibility to use several DNN architectures. This additional work leads to the DeepRare2021 (**DR21**) representation of the image where the features are selected based on their rarity before combining them. In the following sections, we describe **DR21** and its feature visualization.

### 2.1. DeepRare Framework

Figure 1 summarizes the DeepRare architecture. From an input image, features are extracted based on a CNN encoder (such as a VGG16). This network will extract features needed to solve its training task. Here, we use the image classification task on the ImageNet dataset [33], which is a dataset made of very diverse images and more than 1000 classes of objects, thus a general purpose dataset. These weights are available by default in Keras [34] or other development frameworks. Once features from selected layers are extracted, their rarity is computed. The next step lets us select the most rare features and to represent them easily. In the end, the selected features are fused and post-processed in a final saliency map. All of these steps are described in the following sections.



**Figure 1.** Processing for Layer 1. This processing is iterated for all interesting layers from a CNN encoder network. In this example, 13 layers are in total chosen for a VGG16 encoder.

### 2.2. CNN Architectures and Layers Taken into Account

While in **DR19**, the algorithm is applied only to a VGG16 architecture, **DR21** can be applied to various convolutional architectures. In this paper, 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 of being light in terms of weight and computation, which makes 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. In a VGG16, the convolutional layers are gathered within five 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. For VGG19, the same approach was taken into account. We take layers 1 and 2 for the first low-level features; layers 4 and 5 for the second low-level features; layers 7, 8, 9, and 10 for the first middle-level features; layers 12, 13, 14, and 15 for the second middle-level features; and layers 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 layers 16 and 18 for the first low-level features; layers 24 and 32 for the second low-level features; layers 41, 50, 59, and 67 for the first middle-level features; layers 76, 85, 94, and 102 for the second middle-level features; and layers 111, 120, 137, and 146 for the high-level features.

In general, it is important to minimize the number of layers that are taken into account. All the fully-connected layers and embeddings are excluded as the purpose is to reconstruct an image. In addition, layers with redundant information (as pooling or others) are excluded. The idea is to group together in five different groups (from low level 1 to high level) the convolutional layers and only those will be used to extract the rare features. The code provided on GitHub provides the possibility to switch between a VGG16, VGG19, and MobileNetV2 architectures.

### 2.3. Rarity of Deep Features and Top-Down Information

Once the layers taken into account in the algorithm are selected for the given CNN architecture, it is necessary to compute the feature maps rarity within those layers. Figure 1 shows that, on each feature map/activation map within a selected layer, we compute the data rarity. For that, as in DR19, we use the main idea from [9] without the multi-resolution part, which is naturally achieved by the convolutional network 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 (1):

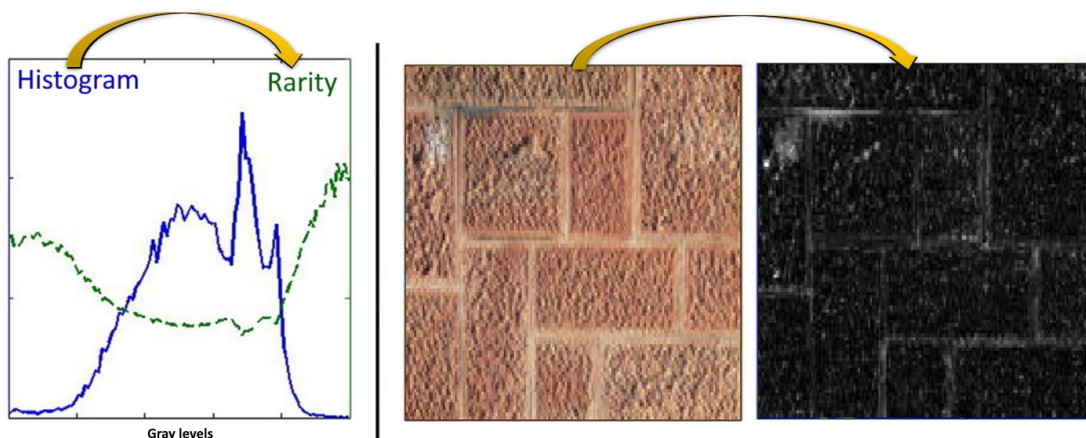
$$R(i) = -\log(p(i)) \quad (1)$$

where  $p(i)$  is the occurrence probability for the pixels of bin  $i$ . The process is explained in Figure 2. A histogram  $p(i)$  is computed (see Figure 2, blue graph in the middle) from a feature map (see Figure 2, left image). The rarity  $R$  is then computed from the histogram using Equation (1) (see Figure 2, green graph in the middle) and a rarity image is reconstructed by backprojection (see Figure 2, right image). This operation projects on each pixel of the output image the rarity value corresponding to the input pixel. This image will highlight pixels in the feature map/activation map, which are rare compared to the other pixels in the feature map. Based on [9], rare pixels are the ones which might attract human attention.

The advantage of this approach is that it is very fast to compute, and this is important as it needs to be applied to numerous feature maps.

### 2.4. Digging into Rare Deep Features

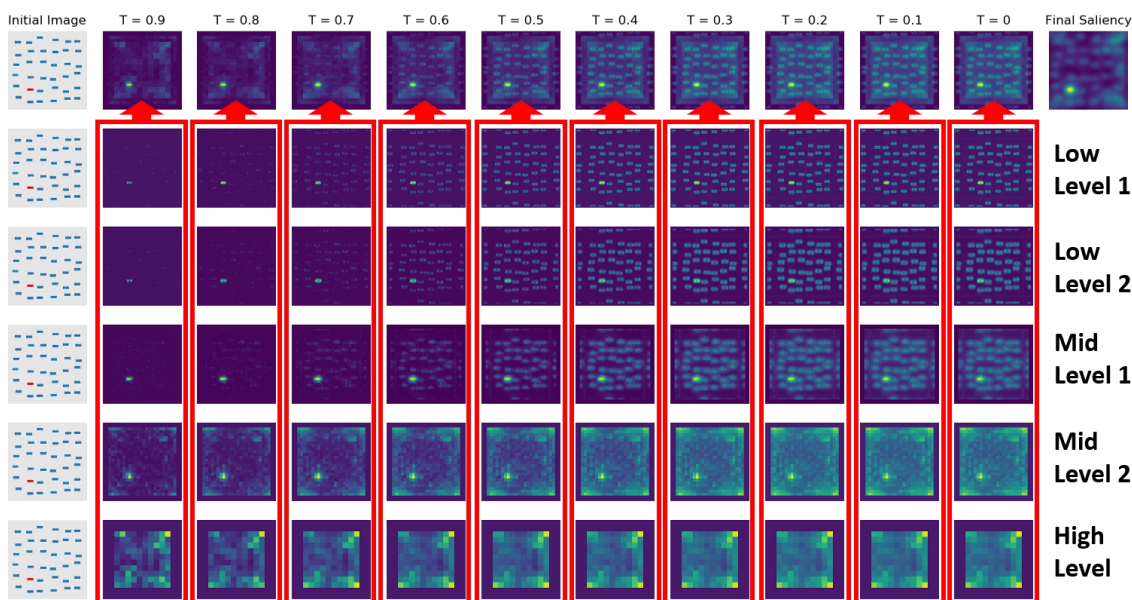
Once we decide the layers which will be taken into account into the model and we compute their rarity, we can go further and select the most rare features in the feature maps. With that aim, we decided to apply a threshold on the computed rarity maps. This threshold is applied directly on the rarity 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.



**Figure 2.** Rarity image reconstruction adapted from [9]. Rarity function (green curve in the left graph) is computed from a histogram (blue curve) of a feature/activation map (middle image). The output is a reconstruction of the map where high values are given for the most “rare” areas (right image).

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, Figures 3–5 are computed with a VGG16 architecture and the five groups discussed in Section 2.2.

In Figure 3, 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 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.

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 (five levels when using a VGG16 architecture) which are already a fusion of the selected layers rarity maps (for the fusion, see the next section). The final fusion of the five levels can be found on the first line. Each map of the first line is the sum of the five DGCM maps obtained for the different

levels which are shown on each column (see Section 2.5 for more details on the fusion). The post-processed final saliency map (see Section 2.6) can be found in the 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 the three first levels from Low level 1 to Mid level 1), 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 Figure 4, 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 like the one in Figure 3.

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 last 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 the person and the large letters in the top-right enough.

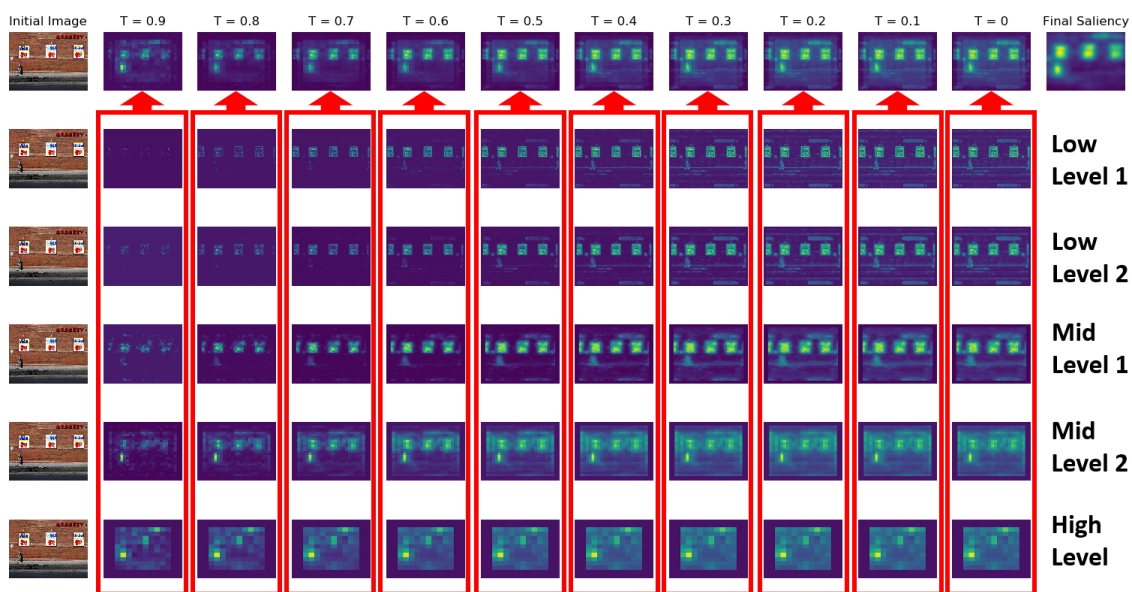
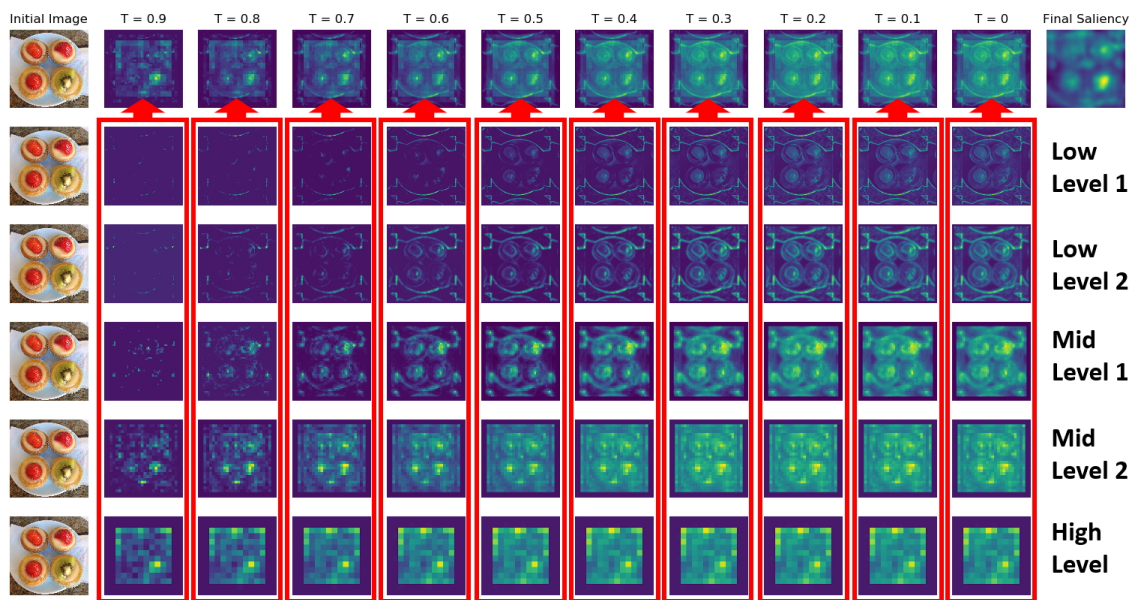


Figure 4. Detailed maps of different levels (from Low Level 1 to High Level) and different thresholds on feature rarity (from 0.9 to no threshold) within the VGG16 architecture.

In Figure 5, one can see the result 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 levels 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 detects the abnormal cake, whatever the threshold is.

Overall, in Figures 3–5, mid level 2 and high level always provide better results with a high threshold such as  $T = 0.9$ , while the lower level feature works better on this high threshold only in specific kinds of images with obvious abnormal patterns due to low-level features. We already understand from here that several thresholds need to be combined to provide better final results.

In [27], 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. However, those rarity-based features were only low-level features. In the current paper, 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 5.** 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.

### 2.5. Data Fusion

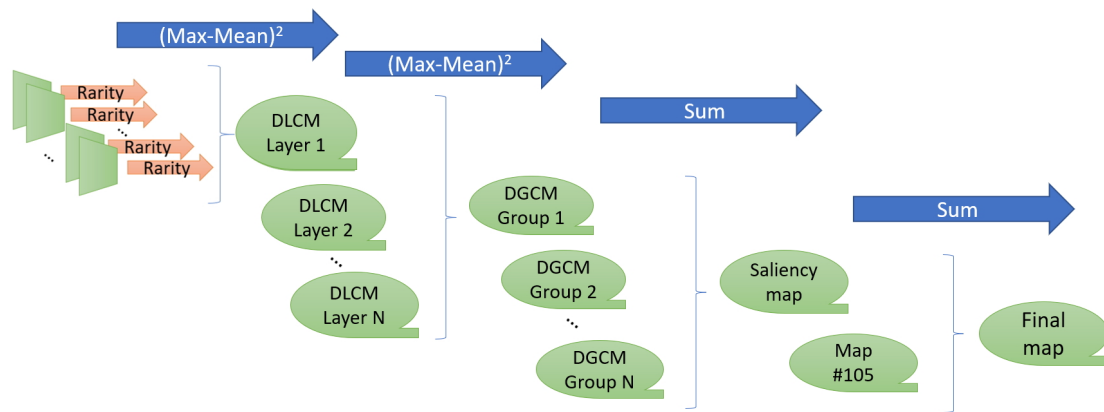
Once the rarity of all feature maps is computed, the results need to be fused together. We use a classical map fusion from [35] where the fusion weights depend on the squared difference between the max and the mean of each map. This is applied to all feature maps within each layer leading to the deep layer conspicuity maps (DLCM), one for each convolutional layer (see Figure 1 for first layer). This fusion approach is efficient and simple, which is important, as it is applied a lot of times on all the feature maps within each layer.

In a second stage, the same fusion method is applied for each of the layer groups arriving to five deep groups' conspicuity maps (DPCM). This fusion is made in a way to give more importance to higher level layers.

Finally, the five DPCM are summed up. In the case that a VGG16 architecture is used, a top-down face map can be added based on feature map #105 from layer 15, which is known to detect faces that are large enough [31].

The entire process is summarized in Figure 6. The final step concerning the #105 map is optional and only works on the VGG16 architecture. The #105 feature map of the 5th convolutional layer of a VGG16 often highlights human faces, especially if they are big enough [31]. However, this layer is very specific to VGG16 and is thus not generic, and the face detection accuracy is much smaller than the one of a real face detector, especially for small faces. It can thus help in improving results on some datasets, but it also introduces false positives. It is much better to use a classical face detector in addition to the model than adding this map in practice.





**Figure 6.** Details on the fusion techniques. The last step using map #105 is optional and only makes sense for a VGG16 architecture.

We show here different configurations of thresholds on the layers and check the results for the VGG16 architecture (Tables 1 and 2). The accuracy is computed here by using the correlation metric (CC) between the final saliency map (image 1) and the real people gaze obtained by using eye-tracking (image 2). This metric is the Pearson correlation between two images described in Equation (2), where  $x_i$  is the intensity of the  $i$ th pixel in image 1,  $y_i$  is the intensity of the  $i$ th pixel in image 2,  $x_m$  is the mean intensity of image 1, and  $y_m$  is the mean intensity of image 2:

$$CC = \frac{\sum_i(x_i - x_m)\sum_i(y_i - y_m)}{\sqrt{\sum_i(x_i - x_m)^2}\sqrt{\sum_i(y_i - y_m)^2}} \tag{2}$$

**Table 1.** The OSIE dataset. Tests with different rarity thresholds, both face and without face features on VGG16.

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 2.** The MIT1003 dataset. Tests with different rarity thresholds, both face and without face features on VGG16.

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

We observe that, on two different validation datasets with natural images (OSIE and MIT1003 which are better described in Section 3), the use of the face in case of the use of VGG16 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 thresholded rarity layers on OSIE (Table 1) and less better results on MIT1003 (Table 2). However, the combination of the thresholds 0 and 0.9 is better in both cases, while the combination between 0 and 0.4 is a little worse 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 and only lets the most rare regions pass. At the end, we have the best mix, which is to take into account all the rare data (threshold 0) and reinforce the areas with very rare data (threshold 0.9).

### 2.6. Saliency Map Post Processing

Once maps were fused, it is well known [36] that a post-processing of the saliency maps can improve the final results depending on the validation metrics. Indeed, the eye-tracking data, which were provided by the datasets that will be used for validation, lead to rather fuzzy eye-tracking saliency maps. The post-processing intends to provide, for each algorithm, the optimal results in terms of correlation with the fuzzy predicted saliency maps to avoid bias due to different post-processing for the tested models. Here, we used a Gaussian low-pass smoothing filtering approach to optimize the final saliency map with the same parameters as in [27].

In addition to smoothing the data, we tested the fact of squaring the data after the smoothing. Tables 3 and 4 show the results for the chosen configuration in Section 2.5, which is the mix of thresholds 0 and 0.9 (1) not filtered, (2) using the filtering technique from [32], 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 cube the image or even more, results are worse, so we decided to keep as the final post processing scheme the filtering from [32] followed by the squared map.

**Table 3.** The OSIE dataset. Tests on threshold 0 and 0.9 by considering the saliency map without filtering, with filtering, and with filtering and squared.

VGG16	With Face	Without Face
$(0 + 0.9)/2$	CC	CC
Not filtered	0.54	0.53
Filtered	0.57	0.56
Filtered + squared	0.59	0.58

**Table 4.** The MIT1003 dataset. Tests on threshold 0 and 0.9 by considering the saliency map without filtering, with filtering, and with filtering and squared.

VGG16	With Face	Without Face
$(0 + 0.9)/2$	CC	CC
Not filtered	0.43	0.42
Filtered	0.48	0.47
Filtered + squared	0.51	0.50

To summarize the difference between the DR19 and DR21 methods, Table 5 shows the main differences. While the basic principle of rarity of the deep features and post processing are the same, several new features are available in DR21.

**Table 5.** DR19 versus DR21.

Features	DR19	DR21
Deep features rarity	yes	yes
Several architectures	no	yes
Rarity thresholds	no	yes
Use of several rarity thresholds	no	yes
Post processing	yes	yes

### 3. Experiments and Results

We use four datasets, namely OSIE [37], MIT1003 [38],  $P^3$ , and  $O^3$  datasets [29] to validate our results. The OSIE dataset contains information at three levels: pixel-level image attributes, object-level attributes, and semantic-level attributes. The MIT1003 dataset contains general-purpose real-life images but has no specific categories or attributes. The  $P^3$  dataset evaluates the ability of saliency algorithms to find singleton targets that focus on color, orientation, and size (without center bias). The  $O^3$  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 four very different datasets to check how saliency models behave when facing images in different contexts.

Concerning metrics, we use measures from [29]. The *number of fixations* (# fix.) is 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) measures how well the target mean saliency is distinguished from the distractors. The *maximum saliency ratio* (MSR) focuses on maximum saliency of the target versus the distractors [39] and the same for the background versus target ( $MSR_b$  and  $MSR_t$ ). We also use standard eye-tracking evaluation metrics from the MIT benchmark [12] such as *Correlation Coefficient* (CC), *Kullback–Leibler divergence* (KL), *Area Under the ROC Curve from Judd* (AUCJ), *Area Under the ROC Curve from Borji* (AUCB), *Normalized Scan-path Saliency* (NSS), and *Similarity* (SIM).

#### 3.1. Qualitative Validation on the Different Datasets

We compare our model to other models on  $P^3$  and  $O^3$  datasets. According to [29], 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 7 shows six samples from the  $P^3$  dataset, which exhibits color, orientation, and size differences of the target. While distractors are still visible on the **DR19** saliency map, the targets are always correctly highlighted compared to RARE2012, which works well mainly for colors and two DNN-based models (MLNet 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).

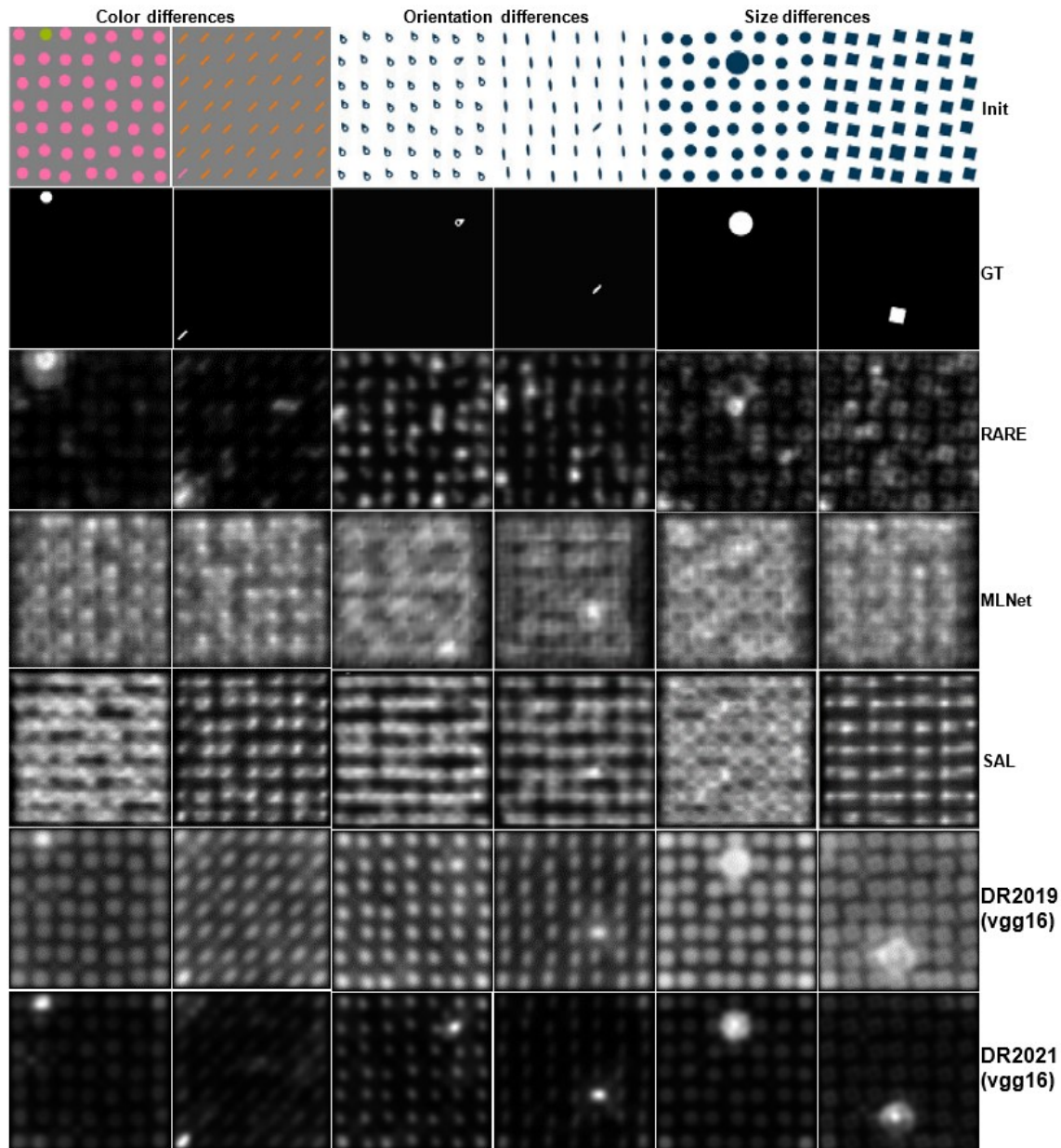
Figure 8 shows images from the  $O^3$  dataset for different target categories (easy or difficult). Again, **DR19** highlights the target better than the DNN-based models. **DR19** seems equivalent on average with RARE. **DR21** shows again a much more precise detection, eliminating distractors and background information. From a qualitative point of view, on the image in Figure 8, **DR21** is the closest to the second line images (ground truth).

Figure 9 shows images from the 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 that might decrease its scores on MIT1003. **DR21** is more precise but still keeps 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 might affect its scores, even if those scores should be higher than **DR19**.

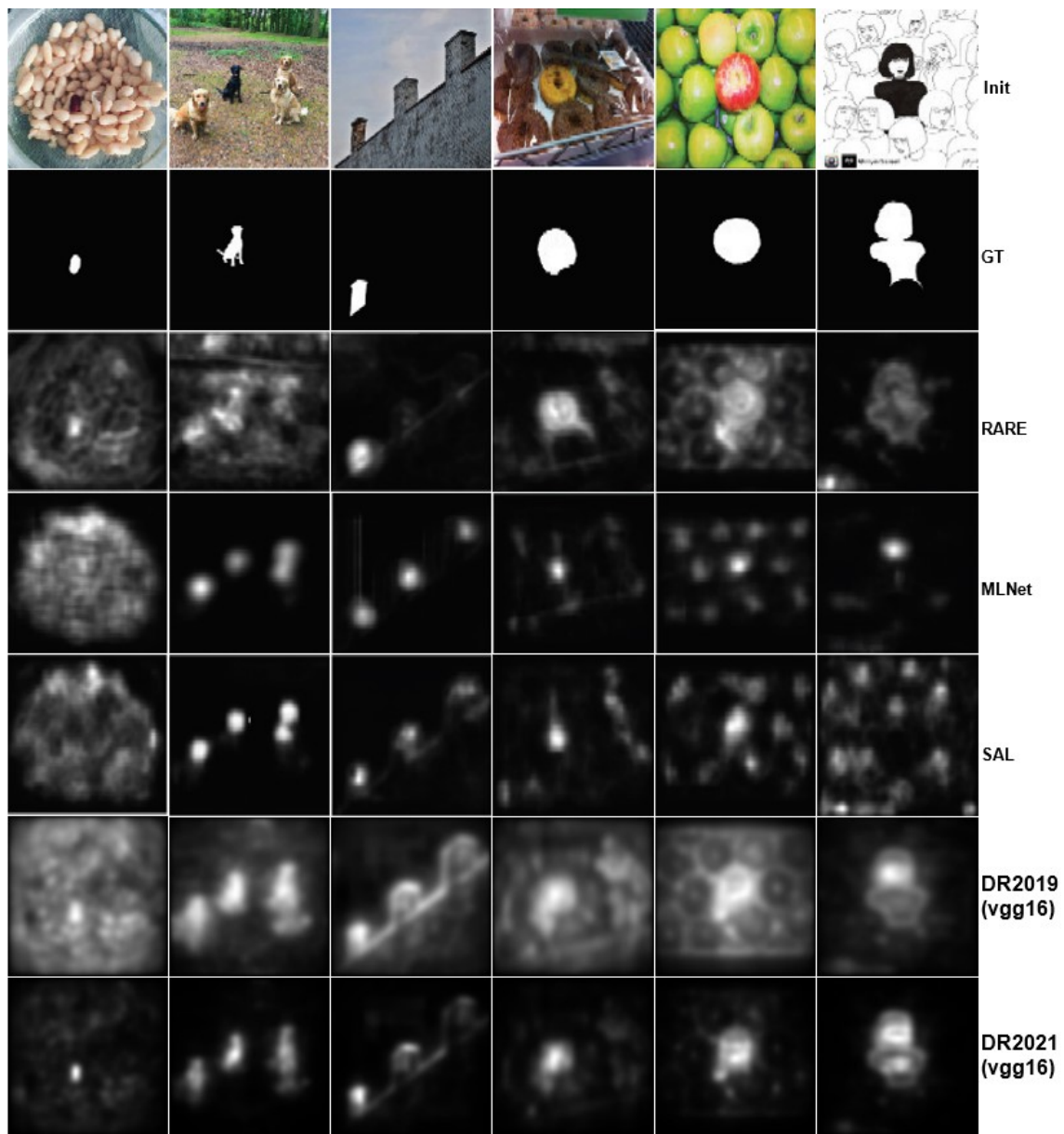
Figure 10 shows the images from the OSIE dataset. **DR19** again spots the main correct salient regions but exhibits a lot of noise or distractors around them with a saliency map less focused as, for example, the one of the ground truth (line two). This issue is partially solved by **DR21**, which is much more selective but still less than some DNN-based models.

Overall, the qualitative study reveals that **DR21** spots the most important regions in all datasets most of the time. On MIT1003 and OSIE datasets, the results of **DR21** are in most cases correct. If some DNN-based models are probably better on MIT1003 or OSIE datasets, one reason is that they are more focused on the top-down areas only as the ground-truth is. Indeed, DNN-based models are trained on images with content close to the ones of MIT1003 and OSIE (general natural images) and different from the other datasets. On  $O^3$

and P<sup>3</sup> datasets, **DR21** clearly shows their superiority on DNN-based models, which are sometimes completely lost with very bad results. **DR19** and especially **DR21** exhibit the most stable behavior, performing well on all datasets while other models might be good on some images but much less good on others.



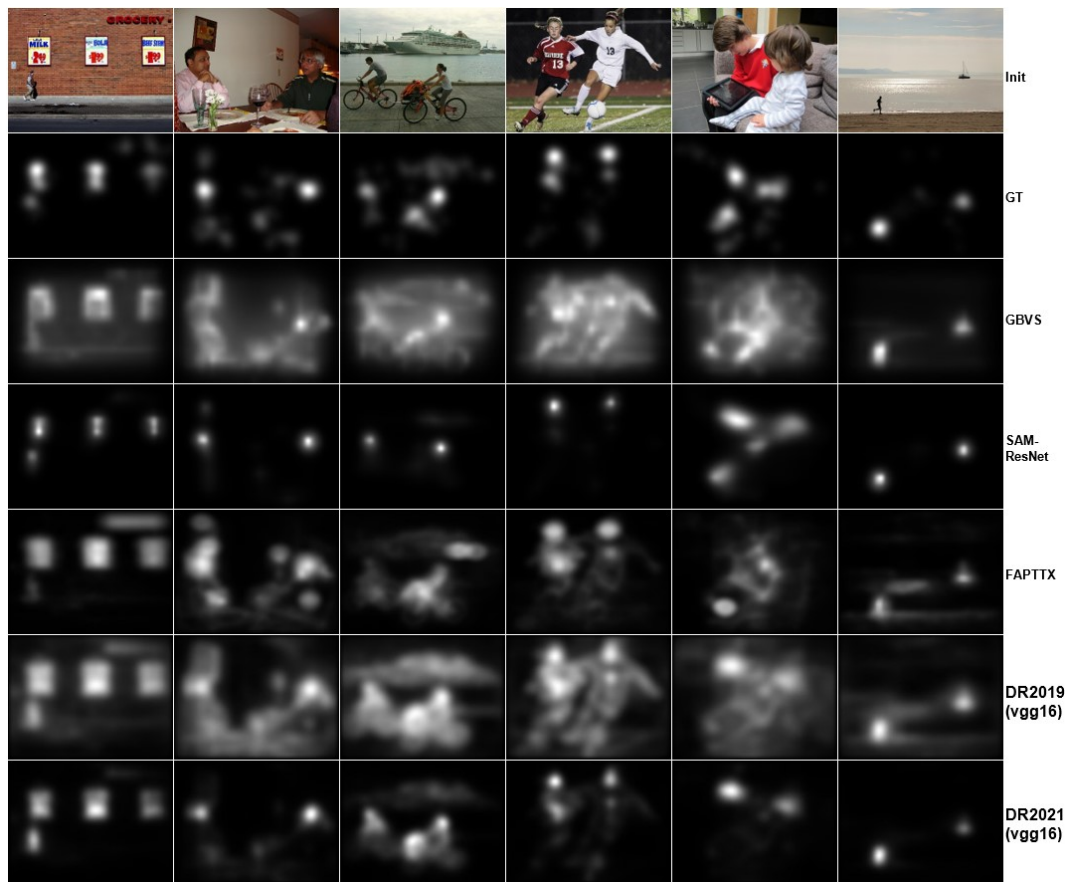
**Figure 7.** Selected samples P<sup>3</sup> dataset. From left to right: target difference in color, orientation, and size. From top to down: initial, ground truth, RARE2012, MLNET, SALICON, DR2019, DR2021.



**Figure 8.** Selected samples O<sup>3</sup> dataset. From top to down: initial, ground truth, RARE2012, MLNET, SALICON, DR2019, DR2021.



**Figure 9.** Selected samples MIT1003 dataset. From top to down: initial, ground truth, RARE2012, SALICON, DR2019, DR2021.



**Figure 10.** Selected samples OSIE dataset. From top to down: initial, ground truth, GBVS, SAM-ResNet, FAPTTX [27], DR2019, DR2021.

### 3.2. Quantitative Validation on the Different Datasets

We make a quantitative validation of different models based on the DeepRare framework on the four datasets shown in the previous section—first, on MIT1003 and OSIE datasets, which show general-purpose images where learning objects is very important. Those datasets should definitely provide an advantage to DNN-based models, which focus on top-down information such as objects (faces, text, etc.) instead of bottom-up salient information. We previously showed in [27] that the DNN-based models mainly learn which objects are attended most of the time, which leads to good results on images implying a high amount of top-down information, while they are very bad in purely bottom-up information.

On the other side, we use the P<sup>3</sup> dataset from [29], which shows that synthetic psychophysical images with pop-out bottom-up objects should work better for classical saliency models and even more with **DR19** and **DR21** models.

Finally, we use the O<sup>3</sup> dataset from [29], which also provides real life images but with odd-out-one regions. The dataset is somewhere in the middle between P<sup>3</sup> on one side and MIT1003 and OSIE datasets on the other side. The O<sup>3</sup> dataset should provide similar difficulty to classical and DNN-based saliency models.

#### 3.2.1. MIT1003 Dataset Evaluation

We summarize in Table 6 the results of **DR19** and **DR21** and also results coming from [29] for MLNet and SALICON models where MLNet was trained with SALICON, P<sup>3</sup> and O<sup>3</sup> datasets, and SALICON was trained with OSIE, P<sup>3</sup>, and O<sup>3</sup> datasets. The idea is to avoid training of MLNet or SALICON models on the MIT1003 dataset where it is evaluated to be fair towards unsupervised models, and, of course, this gives lower results than the same models trained with the MIT1003 dataset. For other models (DeepFeat, eDN, GBVS, RARE2012, BMS, AWS), the figures come from [30].

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), and **DR21-V19** (**DR21** with a VGG19 backbone and without using faces information).

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 Table 6. However, SALICON and MLNet were trained on datasets that are different from the MIT1003 training set, which makes their performances lower than if they were trained on images from MIT1003.

**Table 6.** MIT1003 dataset. DeepRare2021 (VGG19: DR21-V19, VGG16 without faces: DR21-V16-WF, VGG16 with faces: DR21-V16, MobileNetV2: DR21-MN2), DeepRare2019 (VGG16: DR19-V16), DeepRare2019 (VGG16 without faces: DR19-V16-WF, VGG16 with faces: DR19-V16), DFeat, eDN, GBVS, RARE2012, BMS, AWS results come from [30] and SALICON and MLNet come from [29].

Models	AUCJ ↑	AUCB ↑	CC ↑	KL ↓	NSS ↑	SIM ↑
<b>DR21-V19</b>	<b>0.86</b>	<b>0.85</b>	<b>0.56</b>	<b>0.88</b>	<b>1.93</b>	<b>0.50</b>
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	0.83	-	0.51	1.12	1.84	0.41
MLNet	0.82	-	0.46	1.36	1.64	0.35
DFeat	0.86	0.83	0.44	1.41	-	-
eDN	0.86	0.84	0.41	1.54	-	-
GBVS	0.83	0.81	0.42	1.3	-	-
RARE2012	0.75	0.77	0.38	1.41	-	-
BMS	0.75	0.77	0.36	1.45	-	-
AWS	0.71	0.74	0.32	1.54	-	-

### 3.2.2. OSIE Dataset Evaluation

We summarize in Table 7 the results on the OSIE dataset. Here, we added SAM-ResNet and FAPTTX models with the results reported in [27]. SALICON and MLNet models are trained as in Section 3.2.1. SAM-Resnet is used with its default training parameters showing that, when trained on general images without introducing datasets such as P<sup>3</sup>, which can disturb the learning in general images cases, modern DNN-based models are better than DeepRare models 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 DeepRare2021 using VGG16, the same top-down information (TD) as the one that was added to RARE2012 in [27] 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 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 FAPTTX. This again shows that DeepRare models do not capture top-down information, which allows room for future improvements.

Another interesting point is that the 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 FAPPTX shows good results on these kinds of images. 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.

**Table 7.** OSIE dataset. DeepRare2021 (VGG19: DR21-V19, VGG16 without faces: DR21-V16-WF, VGG16 with faces: DR21-V16, MobileNetV2: DR21-MN2), DeepRare2019 (VGG16: DR19-V16), DeepRare2019 (VGG16 without faces: DR19-V16-WF, VGG16 with faces: DR19-V16), and SAM-ResNet, FAPTTX, RARE2012, AWS, GBVS, and AIM come from [27]. We added DeepRare2021 with VGG16 and top-down from [27] called DR21-V16+TD.

Models	AUCJ ↑	AUCB ↑	CC ↑	KL ↓	NSS ↑	SIM ↑
<b>SAM-ResNet</b>	<b>0.90</b>	-	<b>0.77</b>	1.37	<b>3.1</b>	<b>0.65</b>
DR21-V16+TD	0.88	0.83	0.66	0.83	2.32	0.56
FAPTTX	0.87	-	0.62	<b>0.81</b>	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
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	0.83	-	0.46	1.05	1.53	0.43
AWS	0.82	-	0.45	1.11	2.02	0.42
GBVS	0.81	-	0.43	1.08	1.34	0.42
AIM	0.77	-	0.32	1.52	1.07	0.34

### 3.2.3. O<sup>3</sup> Dataset Evaluation

The O<sup>3</sup> dataset uses the MSR metric defined in [29]. When the MSR<sub>t</sub> is higher, it is better as the target is well highlighted compared to the distractors. When MSR<sub>b</sub> is lower, it is better; this 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 8 shows the MSR measures from [29], where we added the results from the DeepRare models (**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.



**Table 8.** Comparing results between several models (SAM-Resnet, CVS, DeepGaze II, FES, ICF, and BMS) 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.

Models	Color		Non-Color		All Targets	
	$MSR_t \uparrow$	$MSR_b \downarrow$	$MSR_t \uparrow$	$MSR_b \downarrow$	$MSR_t \uparrow$	$MSR_b \downarrow$
<b>DR21-V16</b>	<b>1.66</b>	<b>0.74</b>	<b>1.31</b>	1.31	<b>1.45</b>	<b>1.01</b>
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	<b>1.00</b>	1.06	0.89
SAM-ResNet	1.47	1.46	1.04	1.84	1.40	1.52
CVS	1.43	2.43	0.91	4.26	1.34	2.72
DGII	1.32	1.55	0.94	1.95	1.26	1.62
FES	1.34	2.53	0.81	5.93	1.26	3.08
ICF	1.30	2.00	0.84	2.03	1.23	2.01
BMS	1.29	0.97	0.87	1.59	1.22	1.07

For  $MSR_t$  (higher is better) for Color, **DR19** is worse, 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 to the best one (SAM-ResNet). **DR21**, especially using the VGG19 and VGG16 architectures, are definitely the best models, being much better even than efficient DNN-based models such as SAM-ResNet on all the various kinds of images.

If we take into account the  $MSR_b$  (lower is better), the DeepRare models clearly outperform all the others providing the best discrimination between the target and the background. DeepRare models are the only ones with an  $MSR_b$  smaller than 1, which means that on 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 the  $O^3$  dataset.

Table 9 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 [29], 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 MLNet was trained on SALICON by adding with  $P^3$  and  $O^3$  datasets.

**Table 9.** SALICON, MLNet, and DeepRare family (**DR19** and **DR21** with MobileNetV2, VGG19, and VGG16 architectures) results on the  $O^3$  dataset.

Models	$MSR_t \uparrow$	$MSR_b \downarrow$
<b>DR21-V16</b>	<b>1.45</b>	1.01
DR21-V19	1.43	1.13
DR21-MN2	1.12	1.32
DR19	1.06	<b>0.89</b>
MLNet	0.96	0.91
SALICON	0.90	1.26

### 3.2.4. $P^3$ Dataset Evaluation

The  $P^3$  dataset is the one that 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 the DeepRare models deal with the data.

First, we use the average # of fixations and found percentage metrics. The average # of fixations is better if it is lower, as it means that the target is found more rapidly and the percentage metric found is better if it is higher, as it means that a higher percentage of the target is found after 100 fixations. Table 10 shows first the results on  $P^3$  for DeepRare models compared with SALICON and MLNet models. For the SALICON and MLNet

models, they were trained the same way than in Section 3.2.3. 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.

**Table 10.** Comparing results on P<sup>3</sup> dataset.

Models	Avg. # Fix. ↓	% Found ↑
<b>DR21-V16</b>	<b>13.53</b>	<b>89</b>
DR21-V19	13.86	89
DR21-MN2	33.82	72
DR19	16.34	87
MLNet	42.00	44
SALICON	49.37	65

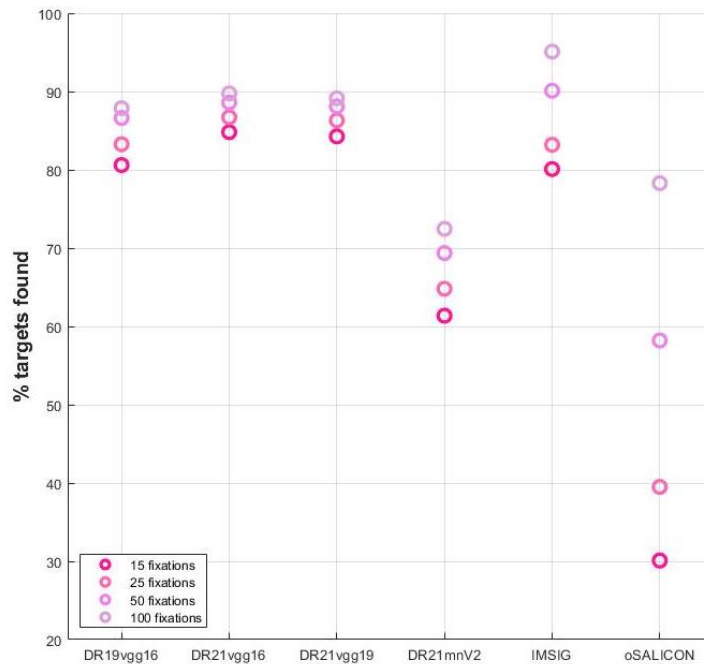
Table 11 provides more details about the found percentage metric after different numbers of fixations (15, 25, 50, and 100) and for specific images where the target is due to color, orientation, or size features with 100 fixations. The results here are compared with classical models which are better in this dataset than DNN-based models. In this table, DeepRare models are 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. Thus, if we look at the results after 15 fixations, only the DeepRare methods are all much better than the others.

**Table 11.** Comparing results on the P<sup>3</sup> dataset. Details on the percentage found after the number of fixation of 15 (%fd15), 25 (%fd25), 50 (%fd50), and 100 (%fd100). Percentage found of the color (%fd-C), orientation (%fd-O), and size (%fd-S) features taken separately.

Models	%fd15	%fd25	%fd50	%fd100	%fd-C	%fd-O	%fd-S
<b>DR21-V16</b>	<b>84.82</b>	<b>86.71</b>	<b>88.60</b>	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	59.87	63.52	79.75	93.48	99.54	90.26	<b>88.53</b>
BMS	58.94	66.37	83.56	<b>95.14</b>	<b>100</b>	<b>100</b>	82.76
ICF	32.63	41.38	68.47	70.18	69.41	<b>100</b>	42.45
oSALICON	30.25	39.75	55.45	78.53	76.35	81.58	70.42

Figure 11 shows the DeepRare family models compared to the best classical model (IMSIG) and the best DNN-based model (oSALICON). If we look at the percentage of targets found after only 15 fixations, then **DR21** with the VGG16 and VGG19 architectures are the best followed by **DR19**, IMSIG, and oSALICON. which are definitely worse. oSALICON (OpenSALICON) refers to [29], and 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<sup>3</sup> dataset again to remain fair.

Finally, the GSI metric (Global Saliency Index) is computed on this dataset. This score is better when it is higher as it measures how target average saliency is distinguished from the distractors. For GSI, Table 12 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 models 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 worse than those of the DeepRare models. If we take into account the DNN-based models, then the GSI scores begin to be even negative, showing that distractors are on average more visible than the salient areas, indicating that DNN-based models do not work at all here.



**Figure 11.** Number of fixations (horizontal axis) vs. % of targets detected (vertical axis). It is chosen on 15, 25, 50, and 100 fixations.

**Table 12.** Comparing results on the P<sup>3</sup> dataset. Global Saliency Index score on color, orientation, and size features, and average score from these three features.

Models	GSI-Color	GSI-Orien.	GSI-Size	GSI-Avg.
<b>DR21-V16</b>	<b>0.77</b>	<b>0.50</b>	0.49	<b>0.59</b>
DR21-V19	0.75	0.49	<b>0.51</b>	0.58
DR21-MN2	0.66	0.42	<b>0.51</b>	0.53
DR19	0.42	0.17	0.15	0.25
RARE2012	0.74	0.01	0.18	0.31
BMS	0.72	0.01	−0.02	0.24
ICF	0.18	−0.02	−0.51	−0.12
oSALICON	−0.01	0.04	−0.11	−0.03

Figures 12–14 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 DeepRare models results with the best classical and the best DNN-based models (in dotted lines).

For color targets (Figure 12), we see a maximum of the GSI score for **DR21** with a VGG16 architecture where GSI is at more than 0.9. If the RARE2012 model (best classical model in dotted red line) is better on small target/distractor color difference, **DR21** is better for larger differences. The ICF model (dotted green line) is worse than all the other models from the DeepRare family on any target/distractor color difference. We also see that **DR21** model is better than **DR19** 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. This is not the case for all tested classical or deep-based models [29] which have a more constant behavior and do not take into account the color difference.

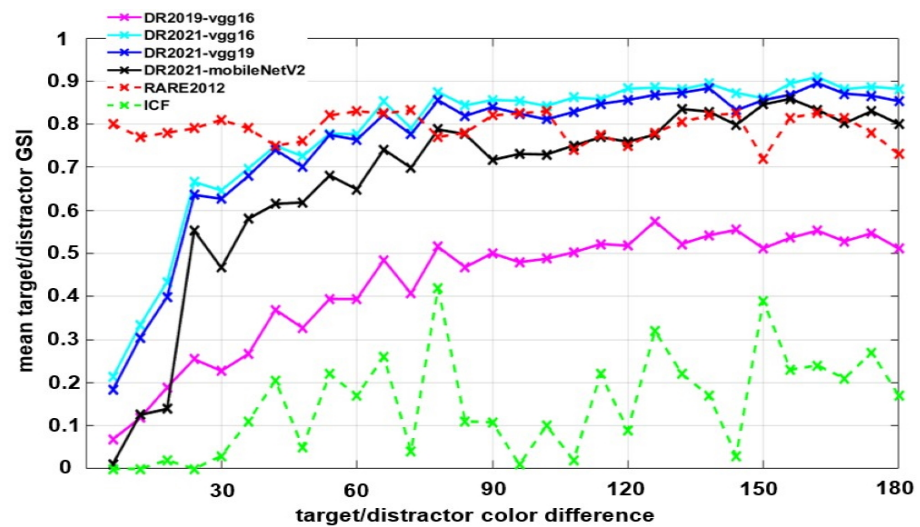


Figure 12. The GSI score for color target/distractor difference. Best classical (dotted red line) and best deep-based (dotted green line) along with the DeepRare family models.

For orientation targets (Figure 13), 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 (red dotted line) and the best classical model (green dotted line) on all target/distractor orientation differences values. We also remark again that the DR21 model is better than DR19 for all used architectures.

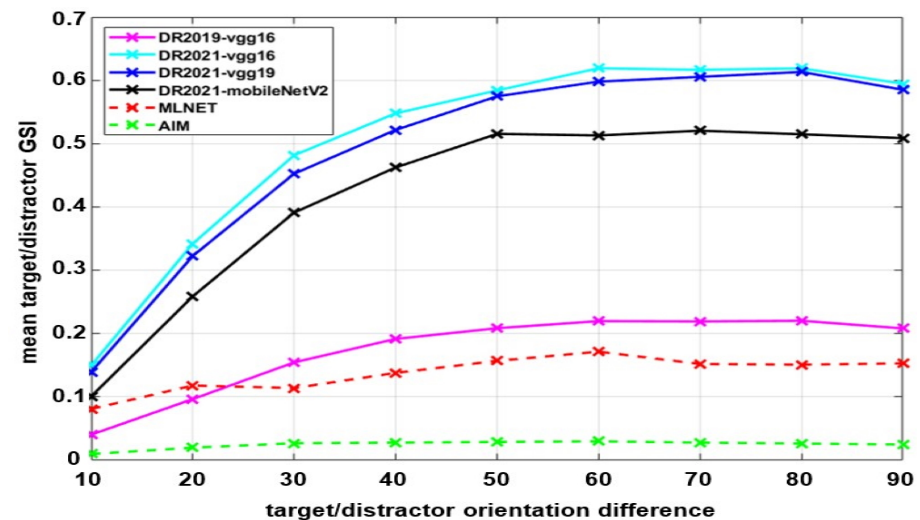


Figure 13. The GSI score for the orientation target/distractor difference. Best classical (green dotted line) and deep learning (red dotted line) along with the DeepRare family models.

In addition, 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, in addition, only the DeepRare family models have a dynamic that is close to the one expected from a human. Indeed, the dotted lines are again more constant and this is the case for all the tested models [29].

For size targets (Figure 14), we see that the maximum of GSI score for the best model (DR21 with a MobileNetV2 architecture) is about 0.7. The best classical model (SSR, red dotted line) is worse 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 DNN-based

(here eDN, green dotted line) 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 targets twice as big as distractors than targets two times smaller than the distractors, which is again biologically coherent. This is also true for DR21, even if, for a very big target size (two times bigger than the distractors), we can see a decrease in the performance. Classical and DNN-based models are again more constant on all target/distractor size ratios which does not make sense from a biological point of view.

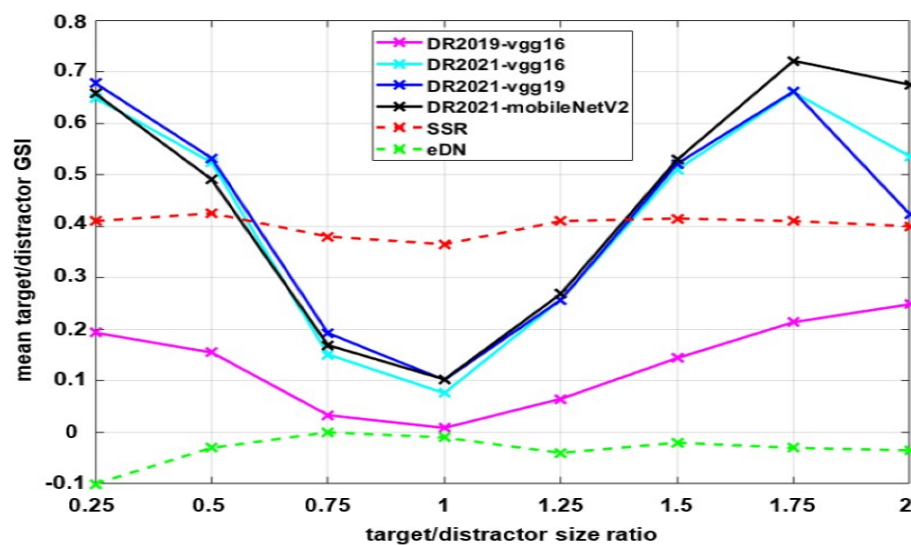


Figure 14. The GSI score for size target/distractor size ratio. Best classical (red dotted line) and deep learning models (green dotted line) along with the DeepRare family models.

#### 4. Discussion and Conclusions

We propose a novel saliency framework called DeepRare using the simplified rarity idea of [9] applied on the deep features extracted by a deep neural network pre-trained on the ImageNet dataset. After a first instantiation of this framework called DeepRare2019, we propose here DeepRare2021, which exhibits several interesting features:

- It needs no additional training, and the default ImageNet training is enough. ImageNet is a generic image dataset that lets DNN encoders extract most of the useful image-related features needed to understand images;
- DeepRare2021 introduces several novelties compared to DeepRare2019, among which is the use of thresholded rarity maps, which drastically improves the results in terms of performance compared to DeepRare2019. The use of the thresholded maps makes DeepRare2021 much less sensitive to distractors, allowing it to focus more on the main surprising areas;
- The model is computationally efficient and is easy to run even on CPU only where the model takes about 1 s per image (Google Colab);
- In comparison with DeepRare2019, where only a VGG16 architecture could fit to the model, the DeepRare2021 approach is very modular, and it is 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 smartphones or for edge computing;
- It is possible to check each layer contribution against the final saliency map 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 DeepRare2019 and even more contrary to black-box DNN-based models. Indeed, if DeepRare2021 does not work well, it is easy to segment the different layer maps and understand from which layer the issue comes from, or if the issue comes from the fusion step. In a specific case, for example, the model was not able to find a surprising object which was very big. While looking to the decomposition of the different layers such the ones that can be seen in Figure 5, only one level (the higher) detected the surprising object, but the final saliency map was not highlighting it because all the other levels were not detecting this object, so the fusion step was the issue in this specific case;

- DeepRare models are 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. DeepRare2021 is thus the most generic model and, when applying it to a new and unknown dataset, it will surely provide results that make sense, while, with DNN-based models, there is no certitude that on a new image dataset it will provide meaningful results (especially if the dataset is not close to the ones used for training). If this is not a crucial issue on natural images which are more or less close to the training datasets, specific datasets such as images with defects for industrial quality control DNN-based models will perform very poorly. In addition, if the defects do not have a specific shape, even by re-training the DNN-based models, they will not be able to learn defects with various shapes as rare features are very hard to learn by definition. If defects attract human gaze, it is specifically because they are unknown and surprising and humans are not able to learn them.

We show that this framework, especially DeepRare2021, is the most stable and generic when testing it on four 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 DeepRare models on the  $O^3$  dataset, where DeepRare2021 outperforms all the models on target/background discrimination and on target/distractor discrimination. Finally, on the  $P^3$  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 that is biologically plausible.

While one cannot expect an unsupervised model such as DeepRare models to be better on the 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 and on any dataset containing surprising areas which have various shapes and thus cannot be learnt by DNN architectures.

Our tests show that DeepRare 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 kinds of datasets. Even if they take into account low- and high-level features, they still remain bottom-up approaches, as FAPTTX results show [27]. Indeed by adding top-down information to RARE2012, the results of FAPTTX are still comparable or a little better than for the DeepRare models. However, if we add the same top-down information as in [27] to DeepRare21 instead of Rare2012, DeepRare 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 DeepRare models. Adding top-down information on top of DeepRare2021 would probably drastically improve

its performance on MIT1003 and OSIE datasets while keeping similar results on  $O^3$  and on  $P^3$  datasets.

The DeepRare family framework shows that deep-features-engineered models might become a good choice in the 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 these kinds of images or when (3) explaining the result is of high importance, for example in the case of industrial standardization. The code of DeepRare2021 can be found at <https://github.com/numediart/VisualAttention-RareFamily>, accessed on 30 November 2019.

**Author Contributions:** Conceptualization, P.K. and M.M.; methodology, P.K., M.M.; software, P.K., M.M.; validation, P.K., investigation, P.K., M.M.; writing—original draft preparation, P.K.; writing—review and editing, M.M., B.G., K.P.; supervision, M.M., B.G., K.P.; project administration, B.G., K.P.; funding acquisition, M.M., B.G., K.P., P.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by ARES-CCD (program AI 2014–2019) by Belgian university cooperation.

**Acknowledgments:** We thank for the support of Walloon region, Belgium to the TRAIL institute for AI and ARIAC project (<https://trail.ac>, accessed on 30 November 2019).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hubel, D.; Wiesel, T. *Brain and Visual Perception: The Story of a 25-Year Collaboration*; Oxford University Press: Oxford, UK, 2004.
2. Broadbent, D.E. *Perception and Communication*; Pergamon Press: Oxford, UK, 1958.
3. Mancas, M.; Le Meur, O. Applications of saliency models. In *From Human Attention to Computational Attention*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 331–377.
4. Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259. [[CrossRef](#)]
5. Itti, L.; Koch, C. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vis. Res.* **2000**, *40*, 1489–1506. [[CrossRef](#)]
6. Rosenholtz, R. A simple saliency model predicts a number of motion popout phenomena. *Vis. Res.* **1999**, *39*, 3157–3163. [[CrossRef](#)]
7. Bruce, N.; Tsotsos, J. Attention based on information maximization. *J. Vis.* **2010**, *7*, 950. [[CrossRef](#)]
8. Harel, J.; Koch, C.; Perona, P. Graph-based visual saliency. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, Vancouver, BC, Canada, 4–7 December 2006; pp. 545–552.
9. Riche, N.; Mancas, M.; Duvinage, M.; Mibulumukini, M.; Gosselin, B.; Dutoit, T. RARE2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. *Signal Process. Image Commun.* **2013**, *28*, 642–658. [[CrossRef](#)]
10. Zhang, J.; Sclaroff, S. Saliency detection: A boolean map approach. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, Sydney, Australia, 1–8 December 2013; pp. 153–160.
11. Garcia-Diaz, A.; Leboran, V.; Fdez-Vidal, X.; Pardo, X. On the relationship between optical variability, visual saliency, and eye fixations: A computational approach. *J. Vis.* **2012**, *12*, 17. [[CrossRef](#)]
12. MIT Saliency Benchmark. Available online: [http://saliency.mit.edu/results\\_mit300.html](http://saliency.mit.edu/results_mit300.html) (accessed on 30 November 2019).
13. Sun, P.; Qin, J. Neural networks based eeg-speech models. *arXiv* **2012**, arXiv:1612.05369.
14. Zhao, R.; Ouyang, W.; Li, H.; Wang, X. Saliency detection by multi-context deep learning. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015; pp. 1265–1274.
15. Qin, J.; Xu, L. Data Acquisition and digital Instrumentation Engineering Modelling for Intelligent Learning and Recognition. *Biosens. J.* **2015**, *4*, 1–4.
16. Han, J.; Zhang, D.; Hu, X.; Li, K.; Ren, J.; Wu, F. Background Prior Based Salient Object Detection via Deep Reconstruction Residual. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *25*, 1.
17. Sun, P.; Qin, J. Enhanced factored three-way restricted boltzmann machines for speech detection. *arXiv* **2016**, arXiv:1611.00326.
18. Jiang, M.; Huang, S.; Duan, J.; Zhao, Q. SALICON: Saliency in Context. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015; pp. 1072–1080.
19. Cornia, M.; Baraldi, L.; Serra, G.; Cucchiara, R. A Deep Multi-Level Network for Saliency Prediction. *arXiv* **2017**, arXiv:1609.01064.
20. Cornia, M.; Baraldi, L.; Serra, G.; Cucchiara, R. Predicting Human Eye Fixations via an LSTM-based Saliency Attentive Model. *IEEE Trans. Image Process.* **2018**, *27*, 5142–5154. [[CrossRef](#)]
21. Lou, J.; Lin, H.; Marshall, D.; Sauppe, D.; Liu, H. TranSalNet: Visual saliency prediction using transformers. *arXiv* **2021**, arXiv:2110.03593.

22. Kroner, A.; Senden, M.; Driessens, K.; Goebel, R. Contextual encoder–decoder network for visual saliency prediction. *Neural Netw.* **2020**, *129*, 261–270. [[CrossRef](#)] [[PubMed](#)]
23. Ding, G.; İmamoğlu, N.; Caglayan, A.; Murakawa, M.; Nakamura, R. SalFBNet: Learning pseudo-saliency distribution via feedback convolutional networks. *Image Vis. Comput.* **2022**, *120*, 104395. [[CrossRef](#)]
24. Linardos, A.; Kümmerer, M.; Press, O.; Bethge, M. DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 12919–12928.
25. Droste, R.; Jiao, J.; Noble, J. Unified image and video saliency modeling. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 419–435.
26. Kümmerer, M.; Wallis, T.S.; Gatys, L.A.; Bethge, M. Understanding Low- and High-Level Contributions to Fixation Prediction. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4799–4808.
27. Kong, P.; Mancas, M.; Thuon, N.; Kheang, S.; Gosselin, B. Do Deep-Learning Saliency Models Really Model Saliency? In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 2331–2335.
28. Kong, P.; Mancas, M.; Kheang, S.; Gosselin, B. Saliency and Object Detection. In Proceedings of the 2018 11th International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI), Montréal, QC, Canada, 14–17 May 2018; pp. 523–528.
29. Kotseruba, I.; Wloka, C.; Rasouli, A.; Tsotsos, J. Do Saliency Models Detect Odd-One-Out Targets? New Datasets and Evaluations. *arXiv* **2020**, arXiv:2005.06583.
30. Mahdi, A.; Qin, J. DeepFeat: A Bottom Up and Top Down Saliency Model Based on Deep Features of Convolutional Neural Nets. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *12*, 54–63. [[CrossRef](#)]
31. Sun, X. Semantic and contrast-aware saliency. *arXiv* **2018**, arXiv:1811.03736.
32. Mancas, M.; Kong, P.; Gosselin, B. Visual Attention: Deep Rare Features. In Proceedings of the 2020 Joint 9th International Conference on Informatics, Electronics Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision Pattern Recognition (icIVPR), Kitakyushu, Japan, 26–29 August 2020; pp. 1–6.
33. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
34. Chollet, F.; Keras. Available online: <https://github.com/fchollet/keras> (accessed on 14 March 2015).
35. Itti, L. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Trans. Image Process.* **2004**, *13*, 1304–1318. [[CrossRef](#)]
36. Judd, T.; Durand, F.; Torralba, A. A Benchmark of Computational Models of Saliency to Predict Human Fixations. In *MIT Technical Report*; MIT: Cambridge, MA, USA, 2012; pp. 1–22.
37. Xu, J.; Jiang, M.; Wang, S.; Kankanhalli, M.S.; Zhao, Q. Predicting human gaze beyond pixels. *J. Vis.* **2014**, *14*, 28. [[CrossRef](#)]
38. Judd, T.; Ehinger, K.; Durand, F.; Torralba, A. Learning to predict where humans look. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2106–2113.
39. Wloka, C.; Yoo, S.A.; Sengupta, R.; Kunic, T.; Tsotsos, J. Psychophysical evaluation of saliency algorithms. *J. Vis.* **2016**, *16*, 1291. [[CrossRef](#)]