A Comparison of Saliency Methods for Deep Learning Explainability

Salamata Konate[†,‡], Léo Lebrat[⋆,‡], Rodrigo Santa Cruz[⋆,‡], Elliot Smith[*],
Andrew Bradley[‡], Clinton Fookes[‡], and Olivier Salvado[†,‡]

[⋆] CSIRO Health and Biosecurity, The Australian eHealth Research Centre, [†] CSIRO Data61,
[‡] Image and Video Laboratory QUT, [*] Maxwell Plus, Australia.

2021 Digital Image Computing: Techniques and Applications (DICTA) | 978-1-6654-1709-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/DICTA52665.2021.9647419

***Abstract*—Saliency methods are widely used to visually explain "black-box" deep learning model outputs to humans. These methods produce meaningful maps which aim to identify the salient part of an image responsible for, and so best explain, a Convolutional Neural Network (CNN) decision. In this paper, we consider the case of a classifier and the role of the two main categories of saliency methods: backpropagation and attribution. The first method is based on the gradient of the output with respect to the network parameters, while the second tests how local image perturbations affect the output. In this paper, we compare the Gradient method, Grad-CAM, Extremal perturbation, and DEEPCOVER, and highlight the complexity in determining which method provides the best explanation of a CNN's decision.**

***Index Terms*—saliency, interpretability, CNN**

I. Introduction

Neural networks for machine learning have drawn significant attention since the evolution of artificial intelligence (AI). Within the field of computer vision, scientists have developed and applied models for a range of tasks including semantic segmentation [1], image classification [2], object recognition [3], and human motion tracking [4]. These models are getting more powerful and accurate, which increases the complexity of neural networks making them deep models.

One of the significant issues with deep learning models is their behaviour as a black-box. As such, it is difficult to understand their decisions and to discover insights into what information the model's predictions have been based on. Indeed, when applying DNN models to real life, such as medical imaging, one should aim to prove a robust, trusted and effective model. In 2018, the European Union released new regulations that stipulate that automated processing should be able to provide an explanation to end users [5, 6].

Recent interest has focused on the development of techniques to interpret imaging DNN models to provide a better understanding of what the network is looking at. In this paper, we focus on saliency techniques to explain deep neural networks. Saliency methods aim to create a map that contains the most important pixels/regions of an image responsible for the network's decision. These methods can be divided into two categories, backpropagation techniques and perturbation techniques [7]. Backpropagation techniques consist of computing the gradient of the output and backpropagating it into the image domain to obtain the salient regions which influence the prediction. These techniques include the gradient-based method [8], Deconvnet [9], Guided BackPropagation [10], SmoothGrad [11], CAM [12] and Grad-CAM [13]. All of these gradient-based methods [8] modify either the gradient computation algorithm or the network architecture. The second category of methods is perturbation methods which consist of finding the minimum perturbation of the pixels in the input image, which maximize the prediction output. Occlusion [9], RISE [14], Shapley value method [15–17], LIME [18], minimal [19] and extremal perturbation [20], and DEEPCOVER [21], occlude the images with either black or grey patches and calculate the impact on the image prediction. However, the occlusion algorithms differ from one method to another.

Even though saliency maps are gaining success in the domain of explainability for deep learning, some issues remain. In 2018, Adebayo et. al [22] demonstrated that many backpropagation-based algorithms have explainability limitations. The first issue is their sensitivity to both the model and the training dataset. Indeed, many of them fail to highlight the salient region of outlier images and are not robust enough to debug neural network models. The second issue [22] is the similarities and differences of the generated maps of these algorithms. In addition, multiple gradient-based methods yield scattered maps, which make the maps hard to interpret. Similarly, Kindermans et. al [7] show the lack of reliability of attribution methods as their maps are sensitive to input variation. These issues raise the need to further examine, understand, and compare the inner workings of state-of-the-art saliency methods for DNNs.

In this paper, we propose to compare state-of-the-art saliency models and discuss their characteristics. We will show that although all methods can highlight salient regions of an image responsible for the decision, it is difficult to qualitatively compare the maps as they do not always show the same area. Indeed, saliency maps are supposed to show the regions the network used to form its decision, however, these regions are often completely different for the same network architecture depending on the choice of the saliency method employed.

The methods that we investigate are Gradient method [8], Grad-CAM [9], Extremal perturbation method [20], and DEEPCOVER [21]. We compare the maps of the different

methods by first showing typical examples which yield to the same highlighted area. We then present examples where the four methods provide contradictory maps. Finally, we discuss important aspects and limitations of the different methods and report their processing time.

II. Methods

A. Deep Neural Networks DNN

Given a deep neural network (DNN) F with N layers, $F(x) = y$ predicts the class of a given colour image $x : \Omega \rightarrow \mathbb{R}^3$ where $\Omega = \{0, ..., H-1\} \times \{0, ..., W-1\}$ is a discontinuous domain. In this paper we investigate multi class classification. In this particular study, we investigate DNNs for Imagenet [23] such that $C_{classes} = 1000$ and $y \in [0, 1]^{1000}$. The aim of the saliency methods is to find the most representative subset of the image x which is responsible for the decision y. There are several techniques to identify this 'best' subset which yields different saliency methods. In the next section we describe some predominant approaches.

B. Saliency methods

Backpropagation-based methods: Many saliency methods are builds upon a backpropagation algorithm. These methods highlight the region of conspicuity of the image, which contributes toward the classification by computing the gradient of the network output with respect to each image pixel. Simonyan et al. [8] proposed to backpropagate the gradient of the network output to the input image to visualise the region responsible for the prediction. It only required a single backward pass from the output prediction to the input image to generate the mask. Similarly, Deconvnet [9], Guided BackPropagation [10] and SmoothGrad [11] computed the gradient but proposed to improve the quality of the saliency maps by reducing its bias and noise by either modifying the backpropagation computation or by average perturbing the saliency maps. To localise salient regions better, CAM [12] proposed to modify the network and Grad-CAM [13] to combine the gradient weight activation.

In this study, we consider the Gradient-based [8] and the Grad-CAM methods [13] that we explain below.

- **Gradient-based method** was proposed by Simonyan et. al in 2013 [8]. They introduced the concept of saliency maps in deep learning. Given an image x, a class c and a learned DNN model with a class score $F_c(x)$, saliency maps were computed based on the gradient w, defined as the derivative of F_c with respect to the image x at the point x_0. The vector w backpropagated the gradient through the DNN to find the pixels which had the most influence on the prediction, as per Eqn. (1).

$$w = \frac{\partial F_c}{\partial x}|_{x_0}. \tag{1}$$

  Finally, the saliency map was computed by reorganizing the elements of the vector w to extract the maximum magnitude M_{ij}: $M_{ij} = |w_{h(i,j)}|$, $h(i, j)$ being the index of the element w at the ith line and jth column. The gradient w considered the pixels to be the most salient as the least perturbed pixels that change the output prediction the most.

- **Grad-CAM** [13] produced an activation map that identified the discriminative region of an image in a single forward-pass. It is a generalisation of the earlier CAM method [12] without requiring the modification of the network architecture. To find the salient map, Grad-CAM first computed the gradient of the image class c with respect to the activation of the k features maps of the last convolutional layer, i.e. $\frac{\partial F_c}{\partial A^k_{ij}}$. Second, the neuron importance weights α^c_k was computed by global-average-pooling the gradient as,

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial F_c}{\partial A^k_{ij}}, \tag{2}$$

  with Z being a normalization factor. Finally, they performed the weighted average of the feature maps with the coefficient α^c_k computed in (2). The localization map Grad-CAM $M^c_{Grad-CAM}$ was then obtained by applying a ReLU on this average,

$$M^c_{Grad-CAM} = ReLU(\sum_k \alpha^c_k A^k). \tag{3}$$

  The ReLu activation unit removed the negative influence for a given class of interest.

Attribution based methods: These methods are also known as perturbation methods as they perturb the image and observe its impact on the output prediction. These aim to find the influence of each input pixel on the output value. Occlusion [9] and RISE [14] identified different sections of the input image by occluding random patches with the value zero. The prediction performance of the occluded image was then compared with the non-occluded image prediction. Shapley value method [15–17] instead, computed the difference of the average of different feature value combinations. LIME method [18] differed from other methods as it performed the linear approximation of the model to find the saliency map of an image. Recently, Fong et al. proposed minimal [19] and extremal [20] perturbation methods to visualise the important regions of the image responsible for the DNN decision. Sun et al. [24] suggested to use statistical fault localization (SFL) techniques [21] to compute a DEEPCOVER saliency map. In this paper, we considered the extremal perturbation [20] and DEEPCOVER methods [24].

- **Extremal perturbation method** was developed by Fong et al. in 2019 [20]. It looked for the set of pixels within the image which maximally affect the output prediction to understand deep networks by masking the pixels with a smooth mask. This study proposed to maximize the perturbation of an image so that the effect on the prediction was minimal. To this end, a gaussian blur mask m was applied to the input for a given size area of the mask

$a|\Omega|$. Eqn (4) computes the mask m_a that maximizes the model's prediction for a chosen size a and a fixed class c,

$$m_a = \underset{m:||m||_1=a|\Omega|, m\in M}{\operatorname{argmax}} F_c(m \otimes x). \quad (4)$$

The extremal perturbation method minimally perturbed an image with a smooth mask to alter the prediction the most.

- **DEEPCOVER method** is a ranking pixel method developed by Sun at al [24]. They defined an explanation as a minimal and sufficient subset of the input image to make the DNN predict the correct class. DEEPCOVER was a black-box method that used Statistical Fault Localization measures (SFL) [21] to produce an explanation map. SFL is a element-ranking algorithm used to detect default localisation in a program. DEEPCOVER assigned a score to each randomly masked pixel p_i of an input image x using SFL measurement. The score was seen as a vector $\langle a^i_{ep}, a^i_{ef}, a^i_{np}, a^i_{nf}\rangle$ of passing p or failing f tests when the pixel was either executed e (not masked), or not executed n (masked). Four types of SFL measures: $Zoltar$ [25], $Ochiai$ [26], $Tarantula$ [27] and $Wong-II$ [28] were investigated for the ranking procedure,

$$Ochiai: \frac{a^s_{ef}}{\sqrt{(a^s_{ef}+a^s_{nf})(a^s_{ef}+a^s_{ep})}}, \quad (5a)$$

$$Tarantula: \frac{\frac{a^s_{ef}}{a^s_{ef}+a^s_{nf}}}{\frac{a^s_{ef}}{a^s_{ef}+a^s_{nf}}+\frac{a^s_{ef}}{a^s_{ep}+a^s_{np}}}, \quad (5b)$$

$$Zoltar: \frac{a^s_{ef}}{a^s_{ef}+a^s_{nf}+a^s_{ep}+\frac{1000a^s_{nf}a^s_{ep}}{a^s_{ef}}}, \quad (5c)$$

$$Wong-II: a^s_{ef}-a^s_{ep}. \quad (5d)$$

After scoring all the pixels, they were sorted in descending order (higher values first). Ordered pixels were then added one by one until the network could predict the correct image class with the given subset of pixels. The obtained group of pixels thereby constituted the SFL explanation map.

III. Experiments

We compared the four methods described above: Gradient saliency [8], Grad-CAM [13], extremal perturbation [20] and DEEPCOVER method [24]. We used the pre-trained VGG16 network of Pytorch developed by [8]. As described in Figure 1, VGG16 consists of 16 weighted convolutional and dense layers, 5 max-pooling layers and a Rectified Linear Unit (ReLU) activation on each layer. We randomly sampled 1000 images of 224x224 pixels from the Imagenet dataset [23] as inputs images.

We first compared saliency maps between the different methods for typical examples, showing the corresponding binary masks by masking the non-salient part of the image. Second, we reported the processing time and parameters needed to compute the saliency maps for each method. Finally, we showcased examples where the methods generated different maps highlighting obvious discrepancies and associated challenges for interpreting the network decision.

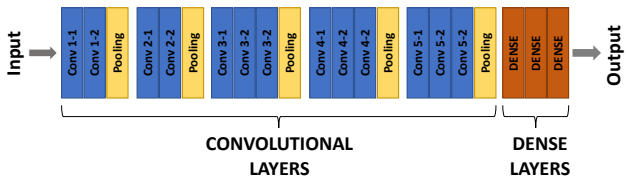

Fig. 1. VGG16 network architecture.

A. Comparison of saliency maps

We illustrate in Figure 2 typical examples of maps generated by the four methods using the VGG16 network to classify images from ImageNet: Gradient, Grad-CAM, Extremal Perturbation [1], and DEEPCOVER [2].

We computed and compared all the methods with the same stopping criterion. We aim to produce the minimal set of salient pixels to predict the input image class. The DEEPCOVER map was as described in the original publication [24]. We modified the three other methods to allow for a fair comparison. We used [20]'s TORCHRAY module for producing a raw saliency map showing a heat map for every pixel. Then, we ordered the pixels in descending order (higher value to lower value). Finally, we added those pixels in their rank order one by one and computed the network prediction until the prediction for the set of salient pixels was the same as the input image prediction output using the full image.

B. Processing time and parameters

Table I shows the parameters needed as input for each method as well as the processing time to generate the saliency map and the processing time while applying our stopping criterion. We thus measured the processing time to obtain an area that produced the correct class of each image. We show the average and standard deviation for 1000 images randomly selected from ImageNet [23]. For the extremal perturbation method, we tested a range of image areas ranging from 0 to 100%. We chose the smallest area that produced the correct class.

C. Manually selected examples

Figure 3 showcases examples where the 4 methods provide 4 different maps.

[1]Gradient, Grad-CAM, and Extremal perturbation are from https://github.com/facebookresearch/TorchRay.

[2]DEEPCOVER is from https://github.com/theyoucheng/deepcover

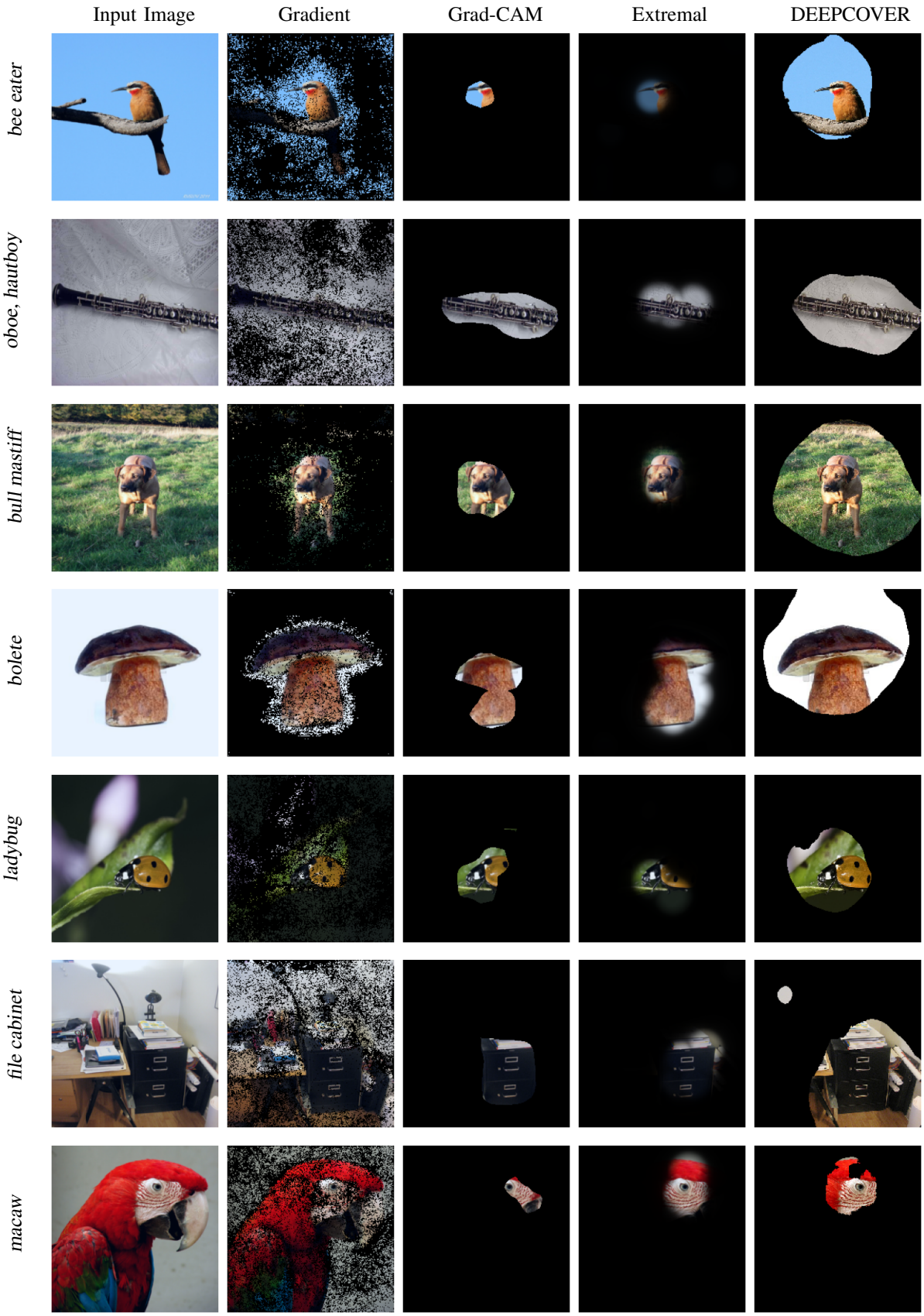

Fig. 2. **Comparison of saliency map methods.** We compare state-of-the-art saliency methods with the minimal essential pixels of the map to predict the input image class. From left to right: input image, gradient-based saliency, Grad-CAM, Extremal perturbation, and DEEPCOVER.

TABLE I
RUN-TIME ANALYSIS AND PARAMETERS OF DIFFERENT METHODS.

Methods	**Processing time (min) without stopping criterion**	**Processing time (min) with stopping criterion**	**Parameters**
Gradient	1.5e-4 $\pm$1.04e-5	2.9 $\pm$1.1	None
Grad-CAM	1.5e-4 $\pm$3.18e-5	1.2 $\pm$1.1	layer ($n \in N$)
Extremal	1.9e-1 $\pm$4.9e-4	2.2 $\pm$2.4	area ($a \in [0, 1]$)
DEEPCOVER	41.3 $\pm$11.4	41.3 $\pm$11.4	measure ($zoltar, tarantula...$) test suite size (s) fraction of masked pixel (σ)

IV. DISCUSSION

We compared four widely used saliency map methods using an experimental method allowing a fair assessment by computing the minimal set of pixels that provide the same output as the original image. We showed typical examples in Figure 2 along with the processing time and main parameters in Table I. We also selected six examples of images where the methods provided different maps illustrating some conflicting results that are now discussed.

The stopping criterion for each method differs, challenging the aim for a fair comparison. The DEEPCOVER study [24] defined a quality explanation map as a minimum set of input image pixels that predict the class. Nevertheless, as shown in Table I, one of the parameters needed for the extremal perturbation method is the size of the saliency area. Some authors have argued [20] that the size of the saliency maps might not be essential to determine whether a saliency map is useful. However, when comparing the methods, randomly choosing the size of the extremal saliency map will create a bias with the other maps. One option can be to set the size of the saliency map to be identical for all the methods. Setting the size of the maps can help to visually compare the different techniques by deducing which one generates the most accurate map to predict the class. However, choosing an arbitrary size can be subjective without any prior information. Intuitively, we would like to have the smallest region of interest that can explain the model output. Thus, we decided to apply the DEEPCOVER stopping criteria to all the techniques although, doing so might limit the other methods. The DEEPCOVER criterion applied to other methods may generate non-optimal maps.

Processing time varies greatly between methods. Table I shows the parameters and the processing time for each method. DEEPCOVER requires setting more parameters and is the slowest method (40 min on average per image), about 15 times slower than the other methods when applying our stopping criterion. However, the processing time of DEEPCOVER may vary depending on the chosen size of the test set and the chosen parameters. In addition, the SFL measurement influences the quality of the saliency maps. The authors of DEEPCOVER did not specify which measure they used in their publication but rather mentioned that there is no best measure as it depends on the input image. We investigated the proposed SFL and decided on the Tarantula measure: Equation 5b as it was providing better and more consistent results. In terms of size, the DEEPCOVER salient region is larger than the others as seen in Figure 2, while Grad-CAM and extremal perturbation maps look very similar in size.

Some methods require user inputs. Grad-CAM is the fastest method as shown in Table I. However, it requires the user to identify the layer of the computed maks. Generating a map in a layer other than the last layer might result in the saliency map being difficult to interpret as noted by others [29]. On the other hand, the gradient-based method is the technique that necessitates fewer parameters as it only needs the input image and the class label to generate the map. When comparing the methods in Figure 2 and 3, it is clear that the gradient method often outputs noisy maps while the other methods produce clearly defined regions of interest.

Several discrepancies between the different methods are illustrated in Figure 3, revealing 5 broad issues that we discuss below.

1) **Multiple objects of the same class in an image.** When multiple objects are present for the same class (kites, guinea pigs, and dandie dinmont images), the four methods provide different answers. For example, in the case of the guinea pig where three of them are present in the image, Grad-CAM highlighted a part of all guinea pigs, while for Extremal perturbation and DeepCover, only one animal is highlighted.

2) **Different salient regions.** When only one object of the image represents the class, the area of interest differs between maps. For the impala example, the gradient-based saliency highlights all the impala and disregards the background, while the Extremal map shows only a part of the animal's face. Grad-CAM and DEEPCOVER, however, focus the attention on the impala's horn. All four methods include part of the two ears.

3) **Background bias to predict the class.** Beyond the object of interest, some techniques select the background to be a salient region. DEEPCOVER, for example, predicts the kite image with both a part of the sky and one of the kites and uses the background images of the kakatoe image to base its prediction without considering the animal's face.

4) **Multiple salient regions to predict an output image.** The number of salient regions for an image may vary

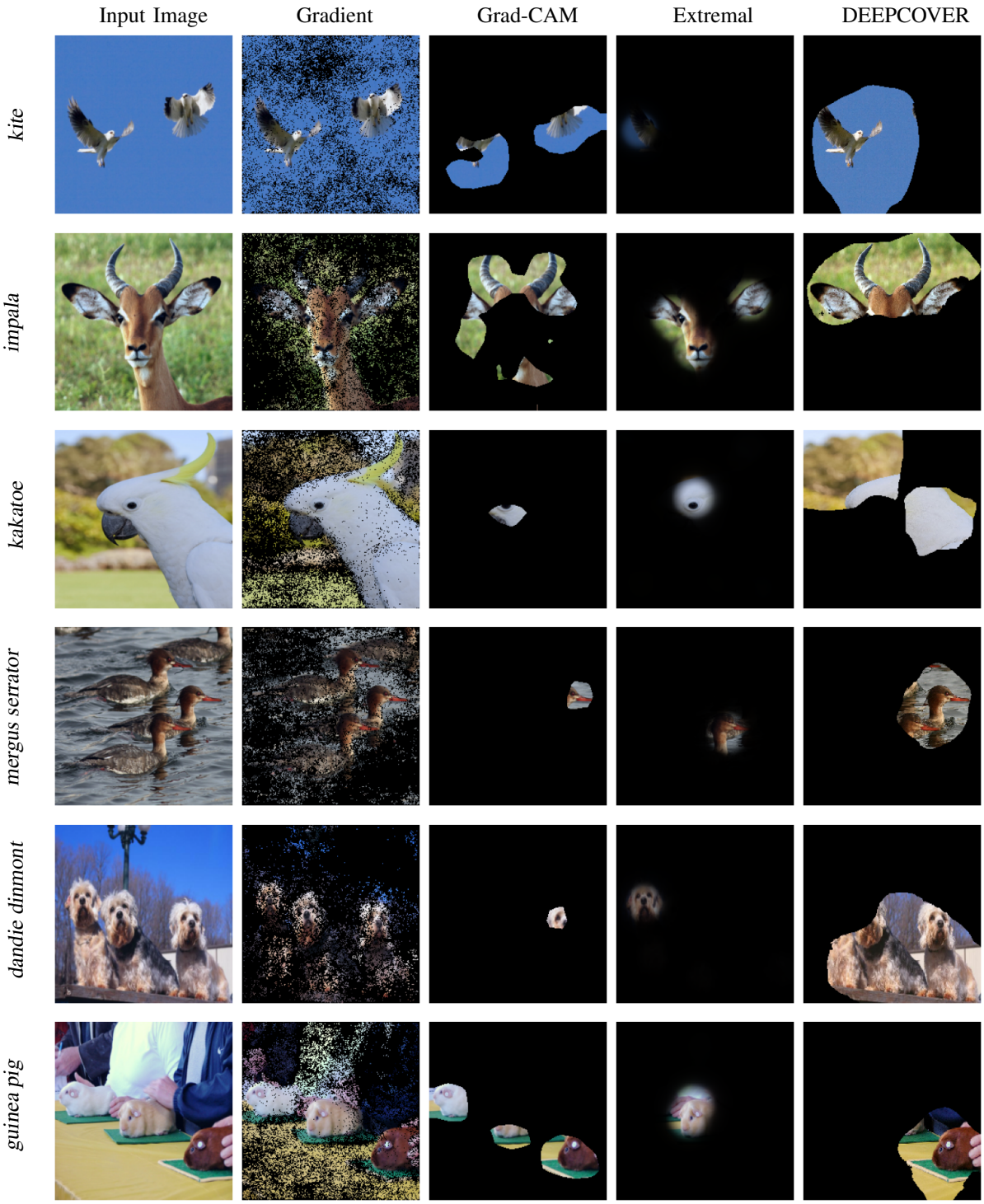

Fig. 3. **Difference between saliency maps.** We compare state-of-the-art saliency methods by requiring that the saliency maps only included the minimum needed set of pixels to predict the input image class. From left to right: input image, gradient-based saliency, Grad-CAM, Extremal perturbation, and DEEPCOVER.

depending on the technique and the input image. Grad-CAM found three areas of interest for the guinea pig, while only two are found for the Extremal method for the impalas, and one area is highlighted by DEEPCOVER for the kite prediction. Having multiple blobs or one big batch to represent the class can be a problem as one region might be enough to predict the image output in some cases, while in other situations, we might want to detect all the parts in an image that explains the DNN output prediction.

5) **No salient region overlaps between the saliency maps.** One of the most complex problems found while comparing the methods was when there were no overlaps between the maps. To predict the kakatoe, the saliency between Grad-CAM, Extremal and DEEPCOVER barely overlaps. However, all those regions are sufficient to predict the output of the image class. It raises the question of which areas the network really used to make its decision, and what is the first used region to classify an image.

One limitation of our experiments is the computation of the running time. We estimated timing when finding a saliency map that predicted the input class. It required running the prediction for the Extremal method many times, which is more time consuming than using the first generated map.

V. Conclusion

In this paper, we compared four state-of-the-art saliency methods (Gradient, Grad-CAM, Extremal Perturbation and DEEPCOVER). We show that although all methods provide insightful saliency maps, some key differences remain. Indeed, two techniques can highlight a different area of what seems to be the region of interest for an image. We demonstrate that judging which method is the most accurate is not trivial as the generated saliency maps are supposed to reflect the network decision and not the user interpretation.

Acknowledgment

This work was funded in part through an Australian Department of Industry, Energy and Resources CRC-P project between CSIRO, Maxwell Plus and I-Med Radiology Network.

References

[1] I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona, and H. Sebastian Seung, "Trainable weka segmentation: a machine learning tool for microscopy pixel classification," *Bioinformatics*, vol. 33, no. 15, pp. 2424–2426, 2017.

[2] S. B. Kotsiantis, I. Zaharakis, P. Pintelas *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[3] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[4] P. Ghosh, J. Song, E. Aksan, and O. Hilliges, "Learning human motion models for long-term predictions," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 458–466.

[5] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation"," *AI magazine*, vol. 38, no. 3, pp. 50–57, 2017.

[6] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[7] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (un) reliability of saliency methods," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019, pp. 267–280.

[8] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[9] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[10] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[11] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[12] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[13] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[14] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," *arXiv preprint arXiv:1806.07421*, 2018.

[15] A. B. Owen and C. Prieur, "On shapley value for measuring importance of dependent inputs," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 5, no. 1, pp. 986–1002, 2017.

[16] D. Janzing, L. Minorics, and P. Blöbaum, "Feature relevance quantification in explainable ai: A causal problem," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2907–2916.

[17] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9269–9278.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[19] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3429–3437.

[20] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth

masks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2950–2958.

[21] L. Naish, H. J. Lee, and K. Ramamohanarao, "A model for spectra-based software diagnosis," *ACM Transactions on software engineering and methodology (TOSEM)*, vol. 20, no. 3, pp. 1–32, 2011.

[22] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Advances in Neural Information Processing Systems*, 2018, pp. 9505–9515.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[24] Y. Sun, H. Chockler, X. Huang, and D. Kroening, "Explaining image classifiers using statistical fault localization," in *European Conference on Computer Vision*. Springer, 2020, pp. 391–406.

[25] R. Abreu, A. González, P. Zoeteweij, and A. J. van Gemund, "Automatic software fault localization using generic program invariants," in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, pp. 712–717.

[26] A. Ochiai, "Zoogeographic studies on the soleoid fishes found in japan and its neighbouring regions," *Bulletin of Japanese Society of Scientific Fisheries*, vol. 22, pp. 526–530, 1957.

[27] J. A. Jones and M. J. Harrold, "Empirical evaluation of the tarantula automatic fault-localization technique," in *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*, 2005, pp. 273–282.

[28] W. E. Wong, Y. Qi, L. Zhao, and K.-Y. Cai, "Effective fault localization using code coverage," in *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, vol. 1. IEEE, 2007, pp. 449–456.

[29] S.-A. Rebuffi, R. Fong, X. Ji, and A. Vedaldi, "There and back again: Revisiting backpropagation saliency methods," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8839–8848.