Interpreting Neural Net Responses

P. Chandrikasingh
puja.chandrikasingh@student.uva.nl
University of Amsterdam
11059842

R. Leushuis
radmir.leushuis@student.uva.nl
University of Amsterdam
10988270

P. Lintl
philipp.lintl@student.uva.nl
University of Amsterdam
12152498

A. Vicol
anca.vicol@student.uva.nl
Vrije Universiteit Amsterdam
12408913

ABSTRACT

This report expands the research of Srinivas and Fleuret [16] by studying the reproducibility of their proposed FullGrad algorithm. The obtained results are overall similar, but the evidence in favour of the better performance of FullGrad is less convincing. However, FullGrad is a promising algorithm as it can be applied to problems in different fields, such as fairness and accountability. It is, for example, extremely suitable for detecting bias, as the salient regions are tighter than for other saliency algorithms.

1 INTRODUCTION

In the last decade, with the rise of computational power, increasingly more advanced and accurate machine learning models for solving everyday and complex problems have emerged. This has also lead to the benefits of applying these models in everyday tasks to inflate. Hence, companies are increasingly more often using these models for making influential decisions, which can have a high impact on the lives of people [13]. This makes it crucial to understand the precise dynamics driving these decisions for the purpose of establishing trust in these decisions. The field that concerns itself with this exact topic is transparency analysis.

The research in this field comprises of understanding the predictions of complex models. This is done in either a global approach, in which the model as a whole is analyzed, or a local approach, in which the individual model predictions are analyzed [13]. Srinivas and Fleuret [16] focuses on the latter by proposing a full-gradients method, which decomposes the neural net response into input and per-neuron sensitivity components. Moreover, for convolutional nets they introduce a saliency map representation, *FullGrad*, that is based on aggregating the full-gradients.

This report expands the research of Srinivas and Fleuret [16] by studying the reproducibility of their proposed FullGrad algorithm. This includes redoing their experiments and testing the robustness of their results by extending the used evaluation metrics and model architectures. Moreover, this report connects the field of transparency with the field of fairness by detecting bias in models. The research in the latter field comprises of analyzing whether an algorithm is fair, i.e. whether it is equally likely to make a mistake about you as about others.

The remainder of this report is organized as follows. Firstly, the full-gradients algorithm is explained in detail. Secondly, the experimental settings for evaluating the algorithm are described. Thirdly, the results are presented. Subsequently, the findings are explained. Next, the findings are connected to other papers in the field. Lastly, the findings are summarized.

2 METHOD

In order to understand the precise dynamics of the proposed saliency algorithm, this section elaborates on the construction of FullGrad and saliency algorithms in general. An important characteristic of saliency algorithms is that they are unable to capture the two main properties of interpretability: *completeness* and *weak dependence*. Srinivas and Fleuret [16] solve this by proposing a method, which is able to capture both the notion of global and local importance. They do so by utilizing full-gradients, which are given by the following pair:

$$G = \left(\nabla_x f(\mathbf{x}), f^b(\mathbf{x})\right) \in \mathbb{R}^{D+F} \tag{1}$$

where $f : \mathbb{R}^D \rightarrow \mathbb{R}$ is a neural network function with inputs $\mathbf{x} \in \mathbb{R}^D$ and biases $\mathbf{b} \in \mathbb{R}^F$. Note however that this report uses the shorthanded notation $f(\mathbf{x})$ instead of the explicit notation $f(\mathbf{x}, \mathbf{b})$. Furthermore, $\nabla_x f(\mathbf{x})$ represents the input-gradients and $f^b(\mathbf{b}) = \nabla_b f(\mathbf{x}, \mathbf{b}) \odot \mathbf{b}$ the bias-gradients.

In order to interpreted the full-gradients, Srinivas and Fleuret [16] introduce a projection[1] of full-gradients for convolutional nets, *FullGrad*, for which this report analyzes the reproducibility.[2] The per-neuron salient maps can be obtained by visualizing a spatial map $\in \mathbb{R}^D$ for every convolutional filter. Subsequently, if a layer l has c_l channels, then one can obtain the per-layer maps by aggregating these maps for all channels c in layer l. Aggregating these maps over the layers l, in turn, will lead to an approximation of the network-wide saliency map. This leads to the following expression for the network-wide saliency map, where we have a total of L layers:

$$S_f(\mathbf{x}) = \psi\left(\nabla_{\mathbf{x}} f(\mathbf{x}) \odot \mathbf{x}\right) + \sum_{l \in L} \sum_{c \in c_l} \psi\left(f^b(\mathbf{x})_c\right) \tag{2}$$

Note that the gradients cannot be visualized directly, hence Srinivas and Fleuret [16] introduce a post-processing operator $\psi(\cdot)$ that performs standard post-processing steps that ensure good viewing contrast and the proper scaling with respect to the input image. This function can differ per task. Srinivas and Fleuret [16] analyze the performance of FullGrad, hence the next section of this report considers how the robustness of their results is assessed.

[1] A reduction form of full-gradients to FullGrad.
[2] The FullGrad algorithm is compared to the Random, gradCAM and Input-Gradient algorithm.

3 EXPERIMENTAL SETUP

The first part of this section describes the metrics Srinivas and Fleuret [16] used and the additional metrics to assess the robustness of the results. The second part elaborates on the experimental settings in which these metrics are used.

3.1 Metrics

Firstly, in order to evaluate the quality of various saliency algorithms, a number of metrics have to be defined. Similar to Srinivas and Fleuret [16], two of the metrics used in this report are given by the accuracy (ACC) and the absolute fractional output change (AFOC). The prior of these is defined as:

$$ACC = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(\hat{y}_i = y_i) \tag{3}$$

where n is the total number of images in the data set and $\mathbb{I}(\hat{y}_i = y_i)$ is an indicator function that is one if the predicted class of image i, $\hat{y}_i$, is equal to the correct class y_i and zero otherwise. The second metric, AFOC, is defined as:

$$AFOC = \frac{|FFN(x_k) - FFN(x)|}{FFN(x)} \tag{4}$$

where x is the original input, x_k is the transformed input[3] and $FFN(x)$ is the feed forward network which outputs the estimated class probabilities for an input x. A side note to the definition of this metric is that Srinivas and Fleuret [16] do not define the precise formula for the metric, which makes it possible that different results are obtained due to a different definition.

In addition to these metrics, this reports expands the metrics proposed by Srinivas and Fleuret [16] by using two additional metrics, given by the Kullback–Leibler (KL) divergence [9] and the absolute percentage change in the unnormalized output (PCUO) [7]. Out of these, the KL divergence measures the difference between two probability distribution and is defined as:

$$D_{\mathrm{KL}}(P_k \| Q) = \sum_{x \in \mathcal{X}} P_k(x_k) \log\left(\frac{P_k(x_k)}{Q(x)}\right) \tag{5}$$

where Q and P_k denote the discrete probability distributions of the output of the network that follow from the final softmax layer. The difference between the two, however, is that for Q the unaltered image, x, is used as input, while for P_k k% pixels are removed from the original image, which results in the transformed input x_k.

One can use the KL divergence in order to evaluate saliency algorithms in the following manner. If a saliency algorithm is effective in identifying important pixels, removing these will lead to P_k significantly differing from Q, which makes $D_{\mathrm{KL}}(P\|Q)$ large. By ordering the saliency algorithms on the magnitude of $D_{\mathrm{KL}}(P_k\|Q)$ for various levels of k one can evaluate the quality of the different algorithms. Naturally, the converse holds for removing the least salient pixels, as under an ideal saliency algorithm this leads to $P_k \approx Q$ and making $D_{\mathrm{KL}}(P_k\|Q) \approx 0$.

A drawback of the KL divergence, however, is that it is effected by the normalization of the probabilities (final softmax layer). I.e. removing the k% most salient pixels does not only reduce the probability of the predicted class, but also increases other probabilities.[4] Moreover, removing the k% least salient pixels should not effect the probability of the predicted class, but can lead to redistribution of the probabilities of the other classes. This in turn leads to a high KL divergence, even though the raw signal (activation before the softmax layer) of the predicted class does not change.

Hence, the last metric (PCUO) utilizes the unnormalized probabilities that are obtained as the output of the penultimate layer (before the softmax layer). This metric uses the percentage change in the unnormalized probability as the result of removing k% of the pixels. It can be interpreted as the change in the raw signal strength of a certain class for a given input. The metric, itself, is defined as follows:

$$PCUO = \frac{|FFN_i^{-1}(x_k) - FFN_i^{-1}(x)|}{FFN_i^{-1}(x)} \tag{6}$$

where $FFN_i^{-1}(x)$ is defined as the output of the second-last layer, hence the layer before the softmax, of the previously defined feed forward network for class i when image x is given as input. A notable detail to this metric is that it is not calculated for each of the classes at once, but concerns itself with only class i. Hence, removing the k% most (least) salient pixels, based on a well working saliency algorithm, leads to this metric being high (low) for the predicted class and preferably low (high) for the other classes, as specific pixels should be important to one specific class. This leads to correct identification of the important (unimportant) regions of the image.

3.2 Experiment

Subsequently, in order to evaluate the quality of saliency maps of the proposed framework, a number of quantitative experiments are performed in which the described metrics are used. For this purpose, this subsection provides a discussion on the experimental framework as used by Srinivas and Fleuret [16], after which extensions to their set up are proposed.

The first experimental framework, utilised by Srinivas and Fleuret [16], is based on a augmented pixel perturbation (PP) scheme and consists of measuring how well a method can identify the unimportant regions of an image. In order to do this, the augmented PP scheme analyzes the effect of removing the k% least salient pixels on the absolute AFOC as defined in equation (4). This is in contrast with the original PP scheme, as it concerns the effect of removing the k% most salient pixels. The main argument for this augmentation is the mitigation of high-frequency artifacts, which leads to a more robust evaluation framework [16]. Srinivas and Fleuret [16] argue that these artifacts, which are created by the change of the most salient pixels to black pixels, lead to the output varying not due to correct identification of salient pixels, but due to these artifacts creating noise in the image.

The second experimental framework utilized by Srinivas and Fleuret [16] is based on the RemOve And Retrain (ROAR) method. Unlike the PP scheme, it consists of calculating how well a method can identify important regions. The experimental framework, similar to the set up of Hooker et al. [5], consists of removing the k%

[3] Transformed by removing k% of the least/most salient pixels for the augmented/original PP scheme.

[4] As the output probabilities need to sum up to one.

of the most salient pixels, after which the classifier is retrained on the new data set. Subsequently, using the retrained model, the change in accuracy, as defined equation (3), is analyzed. The benefit of the retraining step in this case is the mitigation of the previously mentioned artifact artifacts.

In addition to these two experimental frameworks, this report also uses the original definition of the PP scheme, in which the most salient pixels are removed, as defined by Wu et al. [17]. This has as benefit that the results can be compared to existing literature in the field, which evaluates algorithms solely using the original PP scheme. Moreover, instead of blacking out pixels in the removal step for the metrics, this report proposes replacing the pixels by the mean value of the image in order to mitigate the creation of artifact artifacts.[5] This scheme is used in addition to the original blackening scheme. The comparison of the results of both schemes shows the precise effects that artifacts have on the results. Moreover, the added experimental frameworks analyze whether the results also hold for varying settings and assesses the robustness of the results.

Using the frameworks defined above, the FullGrad algorithm[6] is compared to three other algorithms: the Random algorithm[7], the Input-Gradient algorithm[8] and the gradCAM algorithm[9]. Moreover, the three metrics that rely on the removal of $k\%$ of the pixels ($AFOC$, D_{KL} and $PCUO$), are used to evaluate in the different PP schemes. The results in the ROAR framework are assessed by the ACC metric. Furthermore, similar to Srinivas and Fleuret [16] the PP schemes are applied to the ImageNet data set [3] and the ROAR framework is applied to the CIFAR100 data set [8] in order to obtain results that can be compared. However, as Srinivas and Fleuret [16] do not mention which part of ImageNet is used by them and this report only utilized the publicly available validation part of ImageNet, the results might differ.

Another source of differences in the results could be the difference in the used models. Even though both Srinivas and Fleuret [16] and this report use a VGG-16 model for the PP scheme, this report utilizes the retrained VGG-16 model from Pytorch [2] and it is unclear which version of the models Srinivas and Fleuret [16] use. Moreover, Srinivas and Fleuret [16] use a 9-layer VGG model for the ROAR experiments. However, as there is no pretrained version of this model, the pretrained VGG-11 from Pytorch [2] is used in this report.[10] Lastly, in order to determine to what extent the results depend on the utilized model, this report also expands the analysis by additionally using the ResNet model [1].

[5]This is the result of the transition of the remaining pixels to the removed pixel to be smoother in terms of absolute value of the channels [4].
[6]This report utilizes the code implementation of Srinivas and Fleuret [16] with the difference being that it has been adjusted to run on GPU.
[7]This scheme randomly removes $k\%$ of the pixels as a baseline method to estimate the effect of artifact edge creation, as is done by Srinivas and Fleuret [16].
[8]This algorithm is also utilized by Srinivas and Fleuret [16], who uses it as one of the algorithms to which the FullGrad algorithm is compared. This report implements this algorithm by taking the first gradient (with autograd) after forwarding an image.
[9]This algorithm is also utilized by Srinivas and Fleuret [16], who uses it as one of the algorithms to which the FullGrad algorithm is compared. This report uses a similar implementation to Selvaraju et al. [15] available at K. [6]
[10]The VGG-11 model is trained on Imagenet. Hence, using the concept of transfer learning [14], the last fully connected layers are retrained on CIFAR100.

4 RESULTS

In this section, the main results of the analysis are presented. As a consequence of nearly identical results for using the ResNet model and VGG-16 model, only the results for the VGG-16 model are presented, while the results for the ResNet model are shown in the appendix. The section starts with the results using the PP scheme, after which the results using the ROAR scheme are presented. Lastly, we relate the findings of the transparency analysis of the FullGrad algorithm to an implementation in bias detection.

Firstly, comparing Figure 8 and 9 (appendix) with Figure 1 and 2, one can clearly see that the PP scheme using the AFOC metric for evaluation indeed suffers from artifacts.[11] The first two figures are obtained by setting pixels to black, while the latter two are obtained by putting the pixels to the mean of the normalisation which results in the mitigation of artifacts. As all algorithms perform better[12] in the latter case, it is clear that it is important to account for artifacts.

Moreover, using the mean in the adjusted PP scheme in which the $k\%$ of the least salient pixels are changed, results in a clear outperformance of FullGrad with respect to the other algorithms. Hence, the conclusion that follow from Figures 1 and 8 are in line with the findings of Srinivas and Fleuret [16] in terms of performance of FullGrad with respect to other benchmark algorithms. However, we can see that the shape of the AFOCs curves in Figure 8 (Appendix) differ from Srinivas and Fleuret [16] in terms of value and shape. A possible explanation for this could be the difference in the data sets, as Srinivas and Fleuret [16] do not mentioned what part of ImageNet is used for obtaining their results and this report utilizes the validation set. Furthermore, Srinivas and Fleuret [16] do not specify exactly how they define AFOC, which might have lead to a different definition.

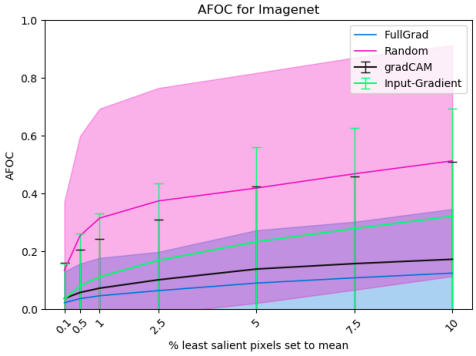

Figure 1: AFOC after changing least salient pixels in a PP scheme with the mean

Furthermore, comparing Figure 8 (and Figure 1) with the results of Srinivas and Fleuret [16] a notable difference is seen. Figure 1 shows that the difference in AFOCs between the gradCAM and FullGrad algorithms increases with k, while Srinivas and Fleuret [16] have found a clear decrease in the distance between FullGrad and gradCAM. However, the differences are small and can be due

[11]Note that in all the figures the lines are the means over the batches. Moreover, the distance covered by the error bars/filling represent one standard deviation from the mean.
[12]On first sight the Random algorithm performs better when the $k\%$ most salient pixels are replaced with black pixels then when they are replaced with the mean. Note, however, that the AFOC line in the latter case lies within one standard error of the line in the first case. Hence, there is not a significant difference between the performances.

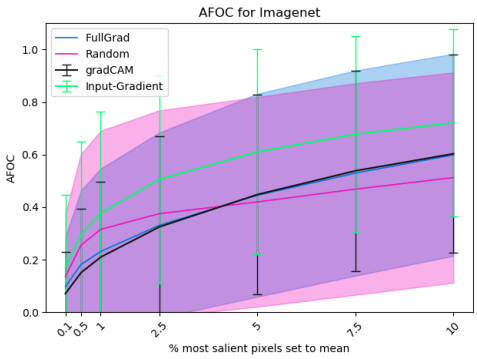

Figure 2: AFOC after replacing the most salient pixels in a PP scheme with the mean

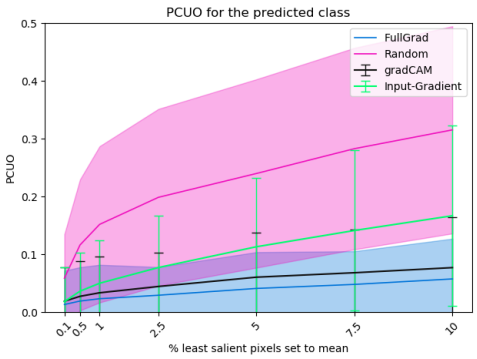

Figure 3: PCUO predicted class after replacing the least salient pixels in a PP scheme with the mean

to the relatively high standard deviation in both this report and the paper of Srinivas and Fleuret [16].

Moreover, Figure 2 shows that when the original PP scheme is used, FullGrad and gradCAM outperform the random algorithm from $k \approx 5\%$. The outperformance of the random algorithm compared to the two other algorithm for low k could be the result of the clustering of salient pixels, as removing a single pixel does not change the cluster of the salient pixels that much. This in turn might lead to the AFOC also changing only marginally. Furthermore, the Input-Gradient algorithm outperforms all other algorithms, which is unexpected considering the results of Srinivas and Fleuret [16], as this implies that the Input-Gradient algorithm is better at identifying important regions than FullGrad. However, the standard errors of Input-Gradient are relatively large and all the other algorithms fall within one standard error distance to it. Hence, there is no significant difference.

Besides evaluating AFOC on the PP scheme, this report also evaluates the Kullback-Leiber divergence, denoted as D_{KL}, and the absolute percentage change in the normalized output, denoted as $PCUO$. From both metrics, we can draw similar conclusions as with the previous metric. Firstly, one can conclude that all algorithms benefit from the mitigation of artifacts, as the performance of the algorithms increases when the pixels are set to the mean instead of black (see figures in Section A.1.1 and A.1.2). Secondly, the overall performance of FullGrad is better, although the standard errors remain relatively large. This implies that the performances are not significantly different.

However, there is a notable difference with the previous results when the PCUO values are considered after removing the least salient pixels. In this setting the performance of FullGrad is substantially better than the performance of the other algorithms, as FullGrad is more than one standard deviation away from Random and Input-Gradient for larger values of k. However, part of this out performance is probably due to artifacts as Random and Input-Gradient perform substantially better when the mean value is used instead of black (Figure 3 and 15).

However, FullGrad cannot identify the most important pixels for a specific class. This can be seen from the identical magnitudes of the PCUO value of the predicted class (Figure 14) and the magnitudes of the PCUO of the 10 most probable classes (besides the predicted class) (Figure 4). Note, however, that it could be the case that the up to 10% most salient pixels are in general important for several classes, which would also result in identical magnitudes. This result does not only hold for FullGrad, but also for all the other algorithms.

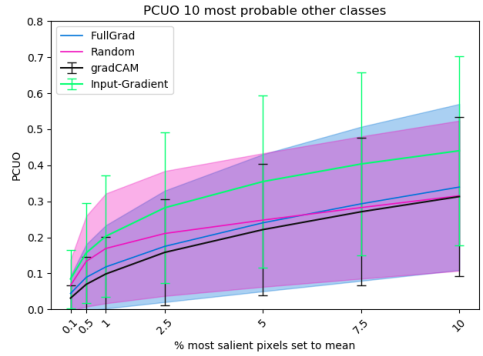

Figure 4: Average PCUO for 10 most probable classes after replacing the most salient pixels in a PP scheme with the mean

So far, the results obtained by the PP framework are in line with the results of Srinivas and Fleuret [16] as the performance of the algorithms do not significantly differ and the mean performance of FullGrad is in general slightly better. Subsequently, the ROAR framework is considered. Figure 5 shows that FullGrad performs better than Input-Gradient and Random algorithm as the decrease in accuracy is the steepest. This decrease is caused by removing the most salient pixels, as the model is less likely to classify an image correctly without these pixels. Hence, the algorithm that caused the steepest decay in accuracy, indeed identifies the most salient pixels.

This result is agreement with the findings of Srinivas and Fleuret [16]. However, Srinivas and Fleuret [16] find significant changes in accuracy, while the standard errors in Figure 5 show that the obtained results are not significant. Note that this can be due to the previously mentioned difference in used models.

Next, it is interesting to extend the transparency analysis with a fairness analysis. This is done by performing a qualitative study on the saliency maps of doctors and nurses in order to detect bias in the classification model. In Section A.3 a detailed description of the dataset and used model is presented. By reviewing the saliency maps, obtained by FullGrad, for both correct and incorrect classified

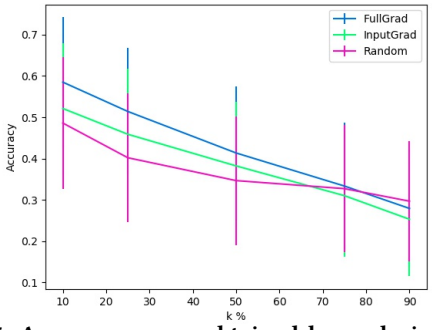

Figure 5: Accuracy scores obtained by replacing the most salient pixels in a ROAR scheme with black pixels

images, it can be seen that the classification model focuses more on ties than on stethoscope when determining if a person is a doctor (see images in A.4). This implies that the model is unfair, as it is more likely to make a mistake about a female doctor than a male doctor, as female doctors usually do not wear ties. An illustrative example is shown in Figure 6 and 7. These figures show that the model correctly classifies the female as doctor. However, despite the female doctor wearing a white lab coat and a stethoscope, the saliency map shows that the classification is based on the man in the back who is wearing a tie and a stethoscope.

Figure 6: Original image

Figure 7: Saliency map

One could argue that the performed bias analysis can be done with any saliency map algorithm and does not require FullGrad specifically. However, most of the other saliency map algorithms display either noisy object boundaries or general and wide important regions. FullGrad on the other hand, can tightly indicate which pixels are important. This can be seen in Section A.5 which include a few illustrative examples. These examples are comparable with the examples of Srinivas and Fleuret [16].

5 DISCUSSION

The findings found in this report are largely in line with the findings of Srinivas and Fleuret [16]. The results obtained by the PP framework are in line with the results of Srinivas and Fleuret [16] as the performance of the algorithms do not significantly differ and the mean performance of FullGrad is in general slightly better. A notable difference, however, is that the results obtained in the ROAR framework yield insignificant results as well. This could be due to fact that there is a difference in models.

A shortcoming of the used frameworks is that it suffers from the creation of artifacts. These frameworks are used to evaluate the quality of salient maps and focus on removing either the least or the most salient pixels. As this leads to artifacts appearing in the image, the changes in the accuracy of the model might be the result of these artifacts and not the removal of the pixels [16]. Even retraining the model, as is done in the ROAR scheme, will lead to sub-optimal evaluation, since it will lead to the model using sections of the altered input that it had formerly not used [16]. Naturally, this leads to poor performance in explaining the original model. Future work can focus on designing new evaluation frameworks, which are unbiased to the problems described above.

6 BROADER IMPLICATIONS

In the previous sections the performance of FullGrad is analyzed. This section briefly considers the relation of FullGrad with existing literature. One of the first proposed methods for importance attribution of features was proposed by Ribeiro et al. [13], who created a framework for approximating the network locally. This enabled one to understand the dynamics of the model in the neighbourhood of a certain input by using these approximations. The FullGrad algorithm expands on the latter by combining a local and global methodology in order to understand the analysis in transparency of the model prediction. Moreover, a notable benefit in using FullGrad for transperency is that it does not create tension between the accuracy and interpretability of the model [10].

Moreover, relating the findings of this paper on transparency analysis to bias detection and accountability, we have shown that one can use the FullGrad algorithm for undesirable output interpretation. This in turn can provide a valuable addition to the accountability framework as proposed by Raji and Buolamwini [12]. The FullGrad algorithm is especially useful in the audit of usage of gender and skin type in commercial facial analysis models [12]. With the recent rise of AI techniques in various applications, it is also useful to enrich the output predictions of black-box models [11].

7 CONCLUSION

In general the findings that are obtained in this report are in line with the results of Srinivas and Fleuret [16]. Moreover, the results are robust as we tested it for different models and with additional metrics in changing experimental frameworks. However, in most cases the performance of FullGrad is not significantly different from other algorithms, which is in the case of the ROAR framework in contrast with the findings of Srinivas and Fleuret [16]. Furthermore, Srinivas and Fleuret [16] do not provide a detailed description of the used models, the used (parts of the) data sets and the precise definitions of the used metrics. Hence, the paper of Srinivas and Fleuret [16] is not awarded with an ACM badge.

However, FullGrad is a promising algorithm as it can be applied to problems in different fields, such as fairness and accountability. Moreover, the saliency maps that are produced by FullGrad highlight a tighter region of important pixels than other algorithms. Hence, more research into FullGrad and into evaluating frameworks is desirable.

REFERENCES

[1] Torch Contributors. [n.d.]. Deep residual networks pre-trained on ImageNet. https://pytorch.org/hub/pytorch_vision_resnet/. Accessed: 2020-01-31.

[2] Torch Contributors. [n.d.]. SOURCE CODE FOR TORCHVISION.MODELS.VGG. https://pytorch.org/docs/stable/_modules/torchvision/models/vgg.html. Accessed: 2020-01-31.

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

[4] Charles Hessel and Jean-Michel Morel. 2018. Quantitative Evaluation of Base and Detail Decomposition Filters Based on their Artifacts. *arXiv preprint arXiv:1808.09411* (2018).

[5] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2018. Evaluating feature importance estimates. *arXiv preprint arXiv:1806.10758* (2018).

[6] Nakashima K. [n.d.]. grad-cam-pytorch, howpublished = https://github.com/kazuto1011/grad-cam-pytorch, note = Accessed: 2020-01-31.

[7] Connie Kou, Hwee Kuan Lee, Jorge Sanz, and Teck Khim Ng. 2018. Theoretical and Experimental Analysis on the Generalizability of Distribution Regression Network. *arXiv preprint arXiv:1811.01506* (2018).

[8] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[9] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *Ann. Math. Statist.* 22, 1 (1951), 79–86.

[10] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*. 4765–4774.

[11] José Mena, Oriol Pujol, and Jordi Vitrià. 2019. Dirichlet uncertainty wrappers for actionable algorithm accuracy accountability and auditability. *arXiv preprint arXiv:1912.12628* (2019).

[12] Inioluwa Deborah Raji and Joy Buolamwini. 2019. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 429–435.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. 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*. ACM, 1135–1144.

[14] Chilamkurthy S. [n.d.]. TRANSFER LEARNING FOR COMPUTER VISION TUTORIAL. https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html. Accessed: 2020-01-31.

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[16] Suraj Srinivas and François Fleuret. 2019. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems*. 4126–4135.

[17] Jinhai Wu, Bin B Zhu, Shipeng Li, and Fuzong Lin. 2004. A secure image authentication algorithm with pixel-level tamper localization. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, Vol. 3. IEEE, 1573–1576.

A APPENDIX

This part of the report consists of additional results. First the results for the PP scheme are given. Thereafter the results obtained by using ResNet instead of VGG-16 model are given. Subsequently, the details for the bias experiment are discussed. After which some results of this experiment are shown. Then, the saliency maps that are obtained by the different algorithms are compared. Lastly, the contribution of each team member is evaluated.

A.1 Additional results PP scheme

This section contains the results for several metrics that are applied in the PP framework on the Imagenet dataset and with an VGG-16 architecture.

Figure 8 shows the performance evaluated with the AFOC metric when the least salient pixels are set to black. The performance is similar to the performance when the pixels are set to the mean as in Figure 1.

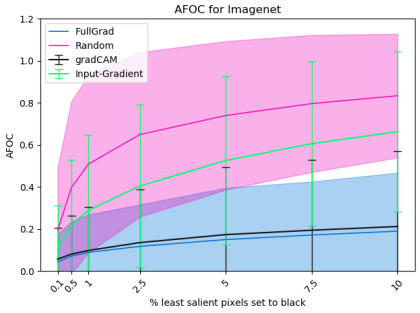

Figure 8: AFOC after removing least salient pixels in a PP scheme

Analysing Figure 9 for the performance of the algorithms in the original PP scheme, one sees that the random algorithm outperforms the others as its mean is the highest. This could imply that although the other algorithms were able to effectively capture unimportant regions of an image, capturing important regions of an image is still quite difficult. However, the under performance could also be due to the formation of artifact edges that distorts both algorithms as the PP scheme neglects the retraining step as a measure of correction. This is indeed confirmed by comparing these results with the results form Figure 2. Furthermore, note that the all the means fall within the confidence interval of the random algorithm, which indicates once again insignificant differences between the algorithms.

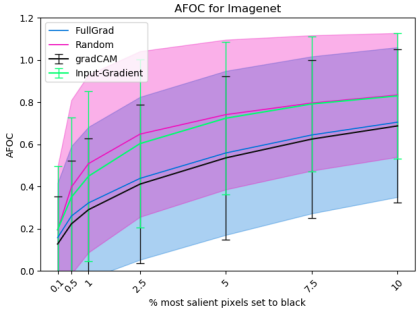

Figure 9: AFOC after removing most salient pixels in a PP scheme

A.1.1 KL divergence. Figure 10 until 13 show the D_{KL} results. The implications of these results are similar to ones obtained by AFOC.

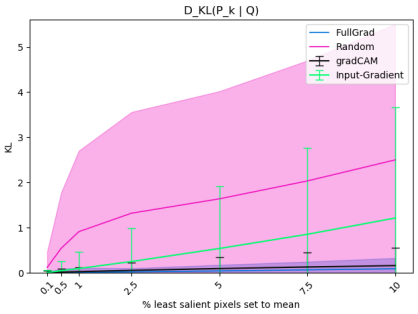

Figure 10: D_{KL} after changing least salient pixels in a PP scheme with the mean

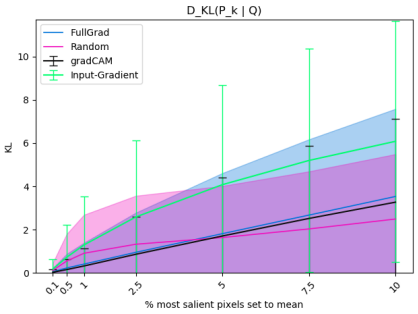

Figure 11: D_{KL} after replacing the most salient pixels in a PP scheme with the mean

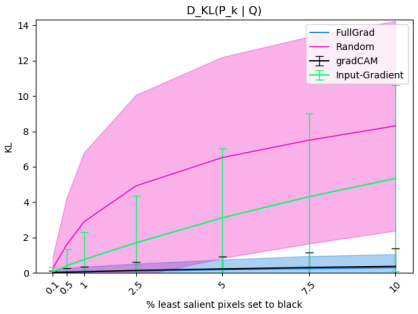

Figure 12: D_{KL} after removing least salient pixels in a PP scheme

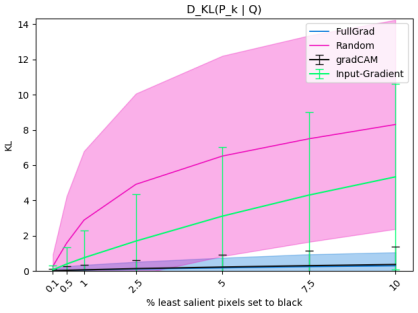

Figure 13: D_{KL} after removing most salient pixels in a PP scheme

A.1.2 PCUO. Figure 14 until 16 show the PCUO results for the predicted class. The implications of these results are similar to ones obtained by AFOC. Figure 17 shows the PCUO results for the 10 most probable classes besides the predicted one.

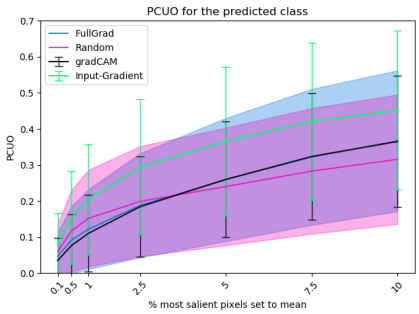

Figure 14: PCUO after replacing the most salient pixels in a PP scheme with the mean

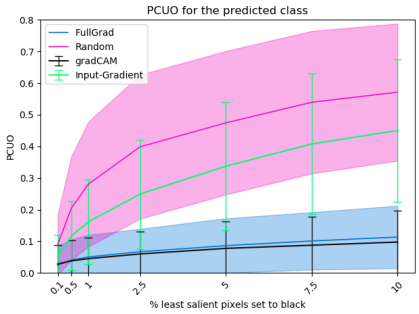

Figure 15: PCUO after removing least salient pixels in a PP scheme

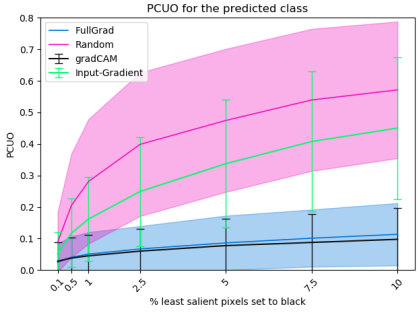

Figure 16: PCUO after removing most salient pixels in a PP scheme

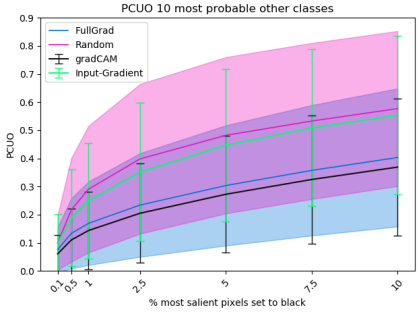

Figure 17: Average PCUO after replacing the most salient pixels in a PP scheme with black

A.2 Additional results ResNet

This section presents the additional results obtained by applying ResNet architecture on ImageNet. The results are similar to the results of the VGG-16 architecture, especially when one takes the confidence intervals into account. Figure 18 until 21 show the results for the different PP schemes in which the algorithms are evaluated with the AFOC metric.

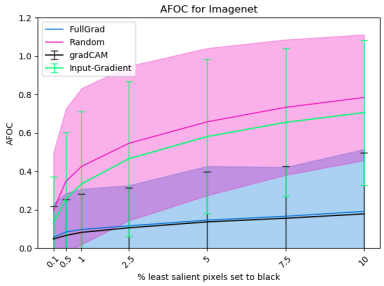

Figure 18: AFOC after setting the least salient pixels in a PP scheme to black

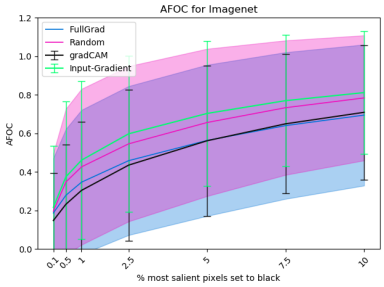

Figure 19: AFOC after setting most salient pixels in a PP scheme to black

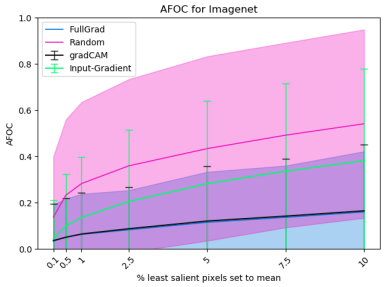

Figure 20: AFOC after setting the least salient pixels in a PP scheme to mean

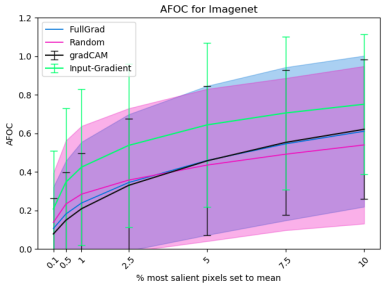

Figure 21: AFOC after setting most salient pixels in a PP scheme to mean

Figure 22 until 25 show the results for the different PP schemes in which the algorithms are evaluated with the D_{KL} metric.

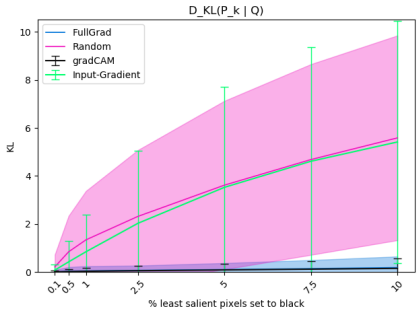

Figure 22: D_{KL} after setting the least salient pixels in a PP scheme to black

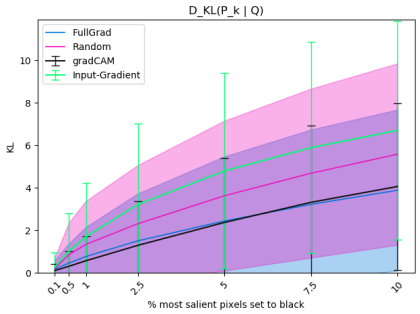

Figure 23: D_{KL} after setting most salient pixels in a PP scheme to black

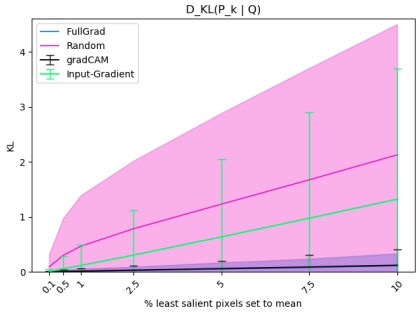

Figure 24: D_{KL} after setting the least salient pixels in a PP scheme to mean

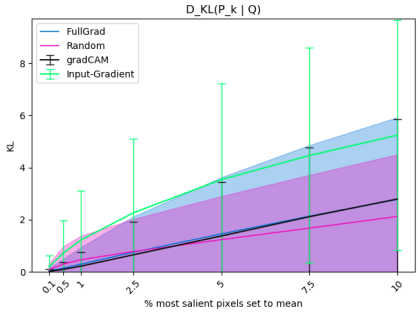

Figure 25: D_{KL} after setting most salient pixels in a PP scheme to mean

Figure 26 until 29 show the results for the different PP schemes in which the algorithms are evaluated with the PCUO metric for the predicted class.

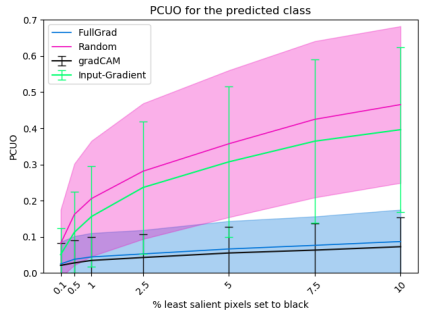

Figure 26: PCUO after setting the least salient pixels in a PP scheme to black

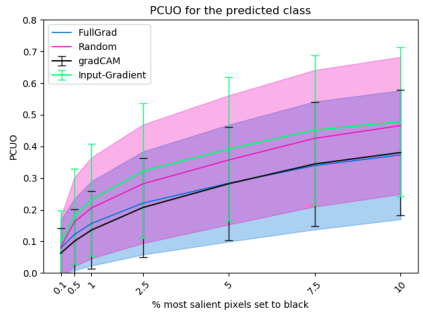

Figure 27: PCUO after setting most salient pixels in a PP scheme to black

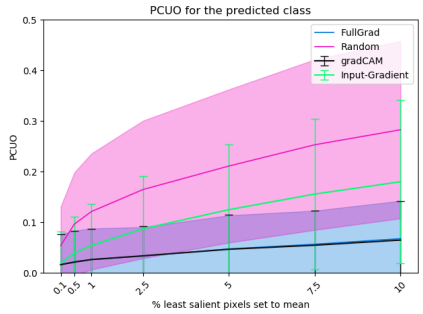

Figure 28: PCUO after setting the least salient pixels in a PP scheme to mean

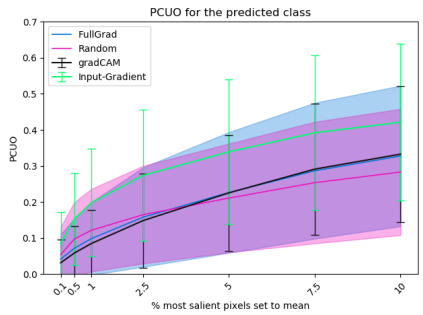

Figure 29: PCUO after setting most salient pixels in a PP scheme to mean

Figure 30 and 31 show the results for the different PP schemes in which the algorithms are evaluated with the with the PCUO metric for the 10 most probable classes (besides the predicted class).

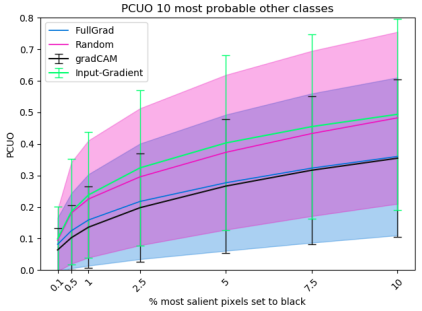

Figure 30: Average PCUO after setting most salient pixels in a PP scheme to black

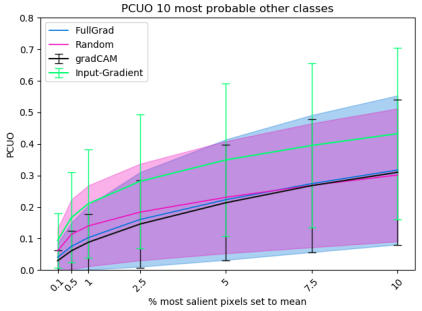

Figure 31: Average PCUO after setting most salient pixels in a PP scheme to mean

A.3 Bias dataset and model

This section contains the description of the bias experiment. The dataset is obtained by using the google-images-download to scrape google images for both male and female doctors, as well as for nurses. The preprocessing of data consists of the removal of not suitable images and cropping of backgrounds.

The training set is composed of 144 doctors of which 19 are female and 210 nurses of which 165 are female. The test set includes 36 doctors of which 19 are female and 33 nurses of which 18 are female.

Two models are used to obtain the results, which are retrained using the principles of transfer learning [14]. The first model is the ResNet 18 model for which the output of the last fully connected layer is changed to dimension 2. The second model is the VGG-16 model with batch normalization for which the classifier layer is also set to output dimension 2. Both models are trained once with frozen convolutions and once by training all parameters. We observe a higher validation accuracy for both model architectures, when the early convolutional layers are trained, as well.

A.4 Images from bias experiment

Figure 32 until 35 show that the model focuses more on ties than on stethoscopes. Figure 36 and 37 shows that this is a disadvantage for female doctors.

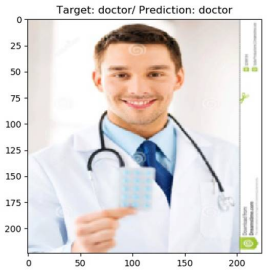

Figure 32: Original image

Figure 33: Saliency map

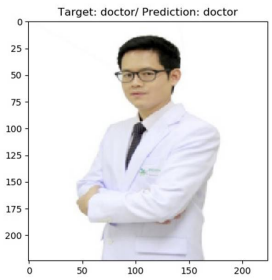

Figure 34: Original image

Figure 35: Saliency map

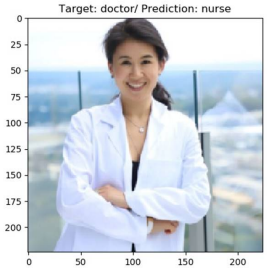

Figure 36: Original image

Figure 37: Saliency map

A.5 Saliency map comparison

This section contains a few illustrative examples to show that Full-Grad highlights a tighter region compared to the other algorithms.

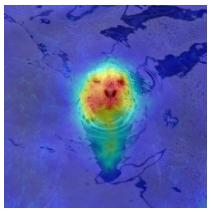

Figure 38: Full-Grad

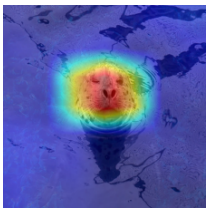

Figure 39: Grad-CAM

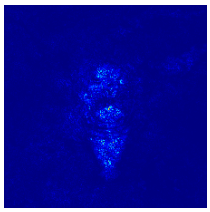

Figure 40: Input-Gradient

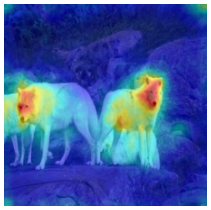

Figure 41: Full-Grad

Figure 42: Grad-CAM

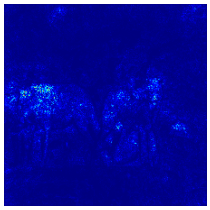

Figure 43: Input-Gradient

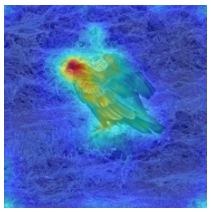

Figure 44: Full-Grad

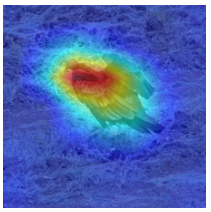

Figure 45: Grad-CAM

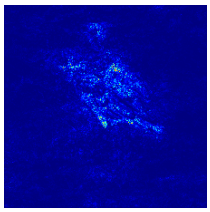

Figure 46: Input-Gradient

Figure 47: Full-Grad

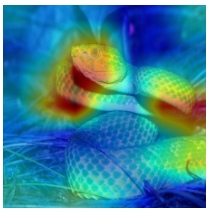

Figure 48: Grad-CAM

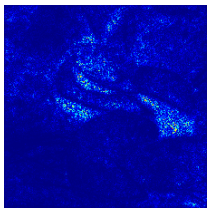

Figure 49: Input-Gradient

A.6 Contribution

Each team member has contributed equally to this report. Anca Vicol and Philipp Lintl implemented the used frameworks, experiments and metrics in Python. They ran all the experiments on Lisa. Radmir Leushuis and Puja Chandrikasingh have written the report, prepared and executed the presentation and worked out the additional metrics and the bias experiment. Moreover, they assisted Anca Vicol and Philipp Lintl with debugging.