Evaluating the Usage of Saliency Maps: A Literature Review

Tom Tornieporth
Technische Universität München
Munich, Germany
t.tornieporth@tum.de

***Abstract*—Over the last few years AI changed from being a tool, which requires expensive hardware to run or comes with expensive subscriptions to use, to being a feature easily useable from a phone, influencing hundreds of decisions every second. But while offering great results in many fields, their results are often not reliable and proofreading a Blackbox based AI model results is initially impossible.**
To alleviate these problems there are multiple methods, all aiming give further insight into the decision making. The goal of this paper is to showcase these systems, analyse their effectiveness in past testing environments and try to devise a plan for how further research could be done.

I. Introduction

The dominant presence of AI within public perception has shifted immensely over the last few years. Breakthroughs made by OpenAI and similar companies managed to make complex software products that are able to mimic voices, write texts for users, recognize and interpret surroundings and help with research available for everyone. But these systems don't come without their flaws or limited use cases. While Large Language Models like ChatGPT are be able to produce a coherent explanation and devising a conclusive summary of information, it will only do so warning the user, that these results are not reliable.
Created is a system which, while offering great results, is not trustworthy enough to solely base decisions on and can only be used to help the user with solving the problem themselves. This mainly boils down to the Blackbox-type nature of Convolutional Neural Networks (CNN), giving the user no indication for its inner proceedings and making normal debugging impossible. To counteract this, the process of generation can be extended, implementing principles like backwards propagation or perturbation of the original input. Analysing these factors allows to gain insight on how the AI classifies.
An example to this, in the field of AI image recognition and classification, are saliency maps. These are algorithms used to highlight the areas of significant importance to the AI's decision making. Running conjoined with the CNN, they display their final results in maps meant to be overlayed or viewed alongside the image. The paper "Evaluating Saliency Map Explanations for Convolutional Neural Networks: A User Study" [1] attempts to test if these maps help the user spot errors made by the CNNs, how their confidence about predicting the Neural Networks output was affected by the given information and ultimately if they were able to gain a better understanding of its prediction behaviour. Initial results observe that, while saliency maps have a significant impact onto users, users still struggle with detecting wrong classifications [1]. But this is only tested under the LRP-algorithm and limited research resources are also split on testing a score based system, displaying users probability scores for objects in the image. This paper will try to fill in these gaps, propose improvements and discuss applications for saliency maps.
To further expand on the work already done, different algorithms suitable for saliency map creation will first be discussed. For this, the paper "Ground truth based comparison of saliency maps algorithms" [2] evaluates the performance of saliency map algorithms via comparing their outputs to a predefined ground truth map of the intended result. Another possible improvement are deep ensembles and the utilization of post-processing methods. This is put to the test in "Clinical validation of saliency maps for understanding deep neural networks in ophthalmology" [3], putting the methods to test in a medical environment. Lastly, "A Unified Approach to Interpreting Model Predictions" approaches graphical explainable AI (XAI) solutions via utilizing Shapley values in the KernelSHAP environment as another reliable, but expensive method.

II. An Overview to XAI in Image Recognition

Convolutional Neural Networks (CNNs) can be diagnosed via multiple ways, some limited to image processing, others are more general tools explainable AI tools that are modified to work on images (SHAP, LRP-algorithms). The saliency map is one of these diagnostic tools, which are based on relevance-scores. To generate such a map, an algorithm runs in conjunction with the CNN, reading and adjusting information between the different Layers of said network. These methods mostly fall under the category of perturbation analysis and gradient based.
Perturbation Analysis involves modifying parts of the image (white-noise, other parts of the image, etc.) in between the different layers and observing how these changes influence the probability scores generated. This gives an understanding which image relate to the observed image [8]. An example of this is LIME, which generates masks onto the image (Fig. 1, analysing how these will affect the classification [9].

Fig. 1. Masks generated by LIME to test the significance of each pixel [4]

Gradient based algorithms work under the principle of backwards propagation. After the CNN has propagated the input through all of the convolutional layers (forward passes), the final output scores are then passed backwards and compared to the score values calculated on earlier layers. The gradient of this different or loss is then calculated and out of these losses, relevance values can be gained. Two popular examples for this kind of backwards propagation are LRP-algorithms, used in the base paper [1] and Grad-CAM [2], a saliency map specific algorithm which defines a broad arrangement of variations. LRP on the other hand is an explainable AI-tool not limited to saliency maps. It continuously propagates backwards passes, passing the generated relevancy values to the previous layer until the first convolutional layer is reached. The relevancy is then computed, giving highly detailed maps, emphasizing contours [7]. Grad-CAM sacrifices these contours in favour of creating more performant heatmaps. For this, the relevance values are only passed back to the last layer, generating less detailed maps, that are very quick to create. Grad-CAM inspires a lot of variations, trying to improve on its design and flaws. SmoothGrad, another highly used principle [2] [3], introducing noise to multiple images and then averaging the maps created by a gradient based algorithm [8]. While normally only positive values are calculated for these scores, there is also specific algorithms, that are able to generate negative relevance, points of no interest to the CNN. One example for this is the Epsilon-LRP algorithm.

In the referenced works multiple algorithms are used, with each algorithm introducing small changes, attempted improvements or performance optimizations. But this paper attempts to evaluate the usage of saliency maps in general, lessening the importance of the individual functionalities of each algorithm. As such this paper will not mention any more algorithms' technical capabilities in detail.

Another approach to making complex systems approachable, is the concept of Shapley values. Originating from game theory, observed subjects are broken down into their individual contributing factors. These are then analysed for their effect in every exponentially generated subset of factors, to evaluate each factor's contribution in detail. A common example for this are medical analysis tools, estimating patients health via generating the Shapley values for their age, lifestyle, smoking habits, etc. to get as detailed of an analysis as possible. But the amount of subsets grows exponentially with the amount of factors, making brute-force approaches quickly lose viability. Because of this, algorithms must be created to approximate these values or eliminate irrelevant subsets to avoid exponential runtime [4].

III. Related work

A. Methods

1) The experiment conducted in the user study: The base paper [1] now goes about testing both saliency map algorithms and the score system, to answer three research questions: Do Saliency Maps allow users to better understand CNN classification behaviour? Does the presence of probability scores for different object classes affect the user similarly? And finally with either scores or saliency maps present, do users actually pay more attention to details of an image?

In preparation for the project the authors trained a neural network on the PASCAL Visual Object Classes Dataset, based on the Keras library in Python. Trained under the ImageNet dataset in the VGG16 architecture, this network was purposefully under-trained, to not meet industry standards, in an attempt be properly able to observe its behaviour.

During the survey, 64 Participants were split into four groups, divided by the amount of information they would later have access to. These included the group having access to either saliency maps, scores, both explainable AI Systems, or neither, to test the first two study goals at the same time. It is also important to notice, that the standard requirements for study participants were set to at least guaranteed to have a technical background, as well as also bolstering a very good reputation on the survey system used by the authors, this service being Prolific, an online survey host.

For the generation of the saliency maps, the authors chose to use the LRP-algorithm only. In the scope of their paper, they only name LIME and LRP algorithms as possible candidates, citing their broad usage. Ultimately it was decided, that they would end up only utilizing the LRP algorithm, as it is more capable of tracing the contour of an object, a preferred kind of saliency map feature as discovered in the papers pilot study. These saliency maps came in the form of a gray-scaled copy of the original image, in which areas with a positive/high relevance would be marked as red, while their counterparts were being marked in blue.

As the Neural Network trained in this kind of environment is able to generate classification scores for multiple object classes, all these detected classes were quantified into a value, that was then normed to the interval [0,1]. Based on this, the user was created with a table, showcasing the 10 highest possibility matches, alongside their normed values.

During the survey, each participant had to solve 14 tasks, each requiring the user to observe and understand the given CNN, to then predict future results it would give. For this they would have to review twelve example images of the AI receiving an input and generating a classification for it. These 12 example images always included six True Positives, examples of the CNN correctly recognizing the image, another three False Negatives, the CNN not classifying the object, as well as three False Positives in cases where the AI incorrectly categorized

parts of the image. Notably the Study did not include any examples of False Negatives.

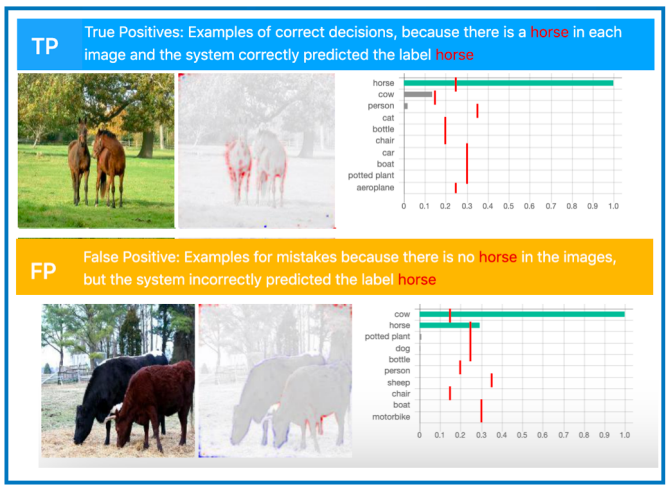

Fig. 2. Two example images of a total of twelve the user had access to as well as the classification by the network. In this case, the user is also presented with both a saliency map and scores [1]

Besides the first group of 16, getting no additional information, the other groups were shown the scores, a saliency map for each image, or both. These examples were intended for the user to gain an understanding on the CNN's behaviour, this insight would then be tested, by having the participants predict the CNN's output for another image, without any other information accessible. Besides just wanting to test, if the users were predicting correctly more often, the paper also tried to capture how the users confidence in their answer was affected by the given information, via having them rank their confidence over the decision on a scale from 1 to 4. Lastly, in regards to the last study goal, the participant was also asked to name few features, they thought the Neural Network would, and would not recognize.

2) Evaluating different Saliency Map algorithms: To introduce more algorithms that can be used for future works, this paper will take the paper "Ground truth-based comparison of saliency maps algorithms" [2] into consideration. It presents and test different type of saliency map algorithms, that are, contrary to the LRP-algorithm, all unmodified algorithms that were from ground up intended for creation of these maps. It is tested in the VGG16, ResNet and SCNN architectures, the former already being used previously [1].

As a way of measuring the performance of each algorithm, they are compared to a previously generated or manually created ground-truth (GT) map, a binary map marking the object of interest. From the saliency map, the n most relevant pixels are then selected, overlapped onto the GT-mask. The resulting overlap is then divided by n to generate a value $m_{tg} = overlap/n$ between 0 and 1.

This kind of evaluation was chosen, as it is quick to compute and simple to implement, but only works under the assumption, that there is an unambigious area of interest with no hidden relations. This is achieved via using specialized image sets, that connect the CNNs output with only one factor. These specialized sets are then tested in four experiments.

For the first experiment, the CNN is trained to recognize green rectangles in the bottom right corner of images of traffic signs. These stem from the German traffic sign recognition benchmark, but only serve as a distraction without any hidden relations. Exaclty half of these images are fitted with the green rectangle, a GT-mask is easily created, as the exact dimensions and position of the rectangle is known.

Afterwards, to simulate a more realistic use-case, the image is conducted on cartoon figures, groups of villains and heroes purely differentiated via their eye color. Done in an attempt to create more complex structures with no hidden relation present, this makes for a clear GT-mask that is manually created for each eye.

In the third experiment, the authors use images from the LISA database containing traffic lights. These are cropped to mostly just consist out of the actual lights in different weather conditions and are cut down to only include green and red phases for simplification. As a lot of these images are quite blurry, and a clear GT-mask cannot be created, the authors used multiple segmentation algorithms, and would use all of them to classify every algorithm for its best m_{gt}-value between every created GT-mask.

Lastly, to analyse the algorithms performance in more complex sample datasets, the fourth experiment uses the ImageNet-S dataset containing random images of 50 classes of objects. This is by far the most realistic testing environment, in which the saliency maps are used on over 700 images. The GT-masks in this experiment can again be created quite easily, as none of the images are particularly blurry like in the previous experiment. Any remaining margins of errors are compensated for by the sheer size of the dataset.

3) Using Saliency Maps in a medical environment: The paper "Clinical validation of saliency maps for understanding deep neural networks in ophthalmology" [3] now puts the so far mostly theoretical work into a more practical appliance. One of the most high-stakes appliance in which Image Recognition CNNs have been used in recent years, is the medical field. But both from a practical and ethical point of view it is questionable at the least to use Blackbox-based systems in this kind of environment. This is due to the fact, that AIs in general are overconfident, as the probabilities given to the user for a certain object to be part of an image, or another kind of result, often does not relate to the actual probability for this event. Combined with the inability of the medical personnel to gain insight on the CNN's inner doings, their implementation has been slow and cautious. To combat this, [3] tries to evaluate the effectiveness of saliency maps via measuring their effectiveness in classifying and recognizing two prevalent eye-diseases: diabetic retinopathy (DR) and neovascular age-related muscular degeneration (nAMD). In an attempt to improve upon the already existing tools, they also try to introduce deep-ensembles to the process and implement their own post-processing method for saliency maps. It is noteworthy that [3] does this from a very technical standpoint,

both explaining algorithms in detail and outlining mathematical formulas, which this paper will mostly ignore, as it is not in the scope of highlighting recent advances in saliency maps or necessary for the discussion.
Similarly to the previously discussed paper [2] the researchers for [3] used the Resnet50 architecture, but also utilized newer architectures InceptionV3 and EfficientNet. Their CNNs were run in via the Keras library in python and trained with the ImageNet Libraries. For the classification of DR, the CNNs were trained on a public set of fundus images (a fundus photos being images of the rear of an eye). This dataset was manually reviewed and each image was classified as "healthy" with a rank applied of 0, or "diseased" and then ranked on whole numbers between 1 and 4. For the nAMD dataset over 3500 scans were graded by retina specialists, for the presence of active nAMD, which is recognized by either intraretinal or subretinal fluid, similar to a ground-truth map [2]. Another 71 images were then annotated by experts as a base comparison for the final evaluation of the maps' efficiency. The material for both diseases were then also varied via introducing rotation, vertical and horizontal translation, different brightness levels or mirroring to the image to diversify the samples, as well as also cropping the images to a unified size. Contrary to the CNNs used in the User Study [1], all Networks in this work were trained up to an industry standard, with the validation score being well above the ones in earlier experiments [1].
To then evaluate the CNN, the authors decided to utilize

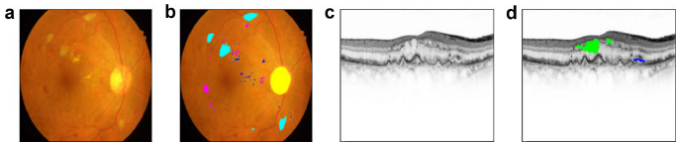

Fig. 3. a) and c) are scans used for diagnosing DR and nAMD, b) and d) are export annotated scans. The different colours highlight the different diagnosing features (microaneurysms, hemorrhages, etc.) relevant for diagnoses [3]

a variation of LRP algorithms and Gradient-based methods, bolstering a total amount of 14 saliency map algorithms. Most noteworthy these include Integrated Gradients and SmoothGrad, already performing well in ground-truth based tests earlier [2], Guided Backprop as well as an ensemble of LRP variants. These algorithms were then assembled into so called Deep Ensembles.
Deep Ensembles are collections of Neural Networks meant to better represent the actual probabilities of a certain event via observing multiple machines. For this, different initial variables are introduced to to the Networks or even different training sets, which leads to results of higher quality. In combination with saliency maps this leads to different parts of the ensemble recognizing and marking different symptoms of each disease. Combined with running on 3 different architectures, each ensemble consisting out of 5 CNNs and 2 diseases being tested, the study uses a total of 30 Neural Networks.
In an attempt to simplify the massive amounts of results produced by the Deep Ensembles, [3] introduces their own post-processing method. As simple linear addition or similar combination methods would make the resulting combined saliency map sparse and without significant points of relevance, which would diminish the point of introducing the ensembles in the first place, the paper utilizes non-linear scaling methods to individually present the distinct highlights of each saliency map. This scaling is meant to not just emphasize individual maps, but also "grow" them represented by a value between 0 and 1. With these methods created, the paper now tests the effect of these combined measures via comparing the created and processed saliency maps with expert-annotated versions of these pictures highlighting the symptom areas. This is quite comparable to ground-truth based comparisons [2].

4) Approximating Shapley Values: Predictions explained with Shapley values continue to be some of the most intuitive and most understandable approaches to XAI. But calculating vanilla Shapley values for an image quickly becomes uncomputational, as even low resolution saliency maps with $5x6$ zones would already require the relevancy of $2^{30} > 1.000.000.000$ different subsets to be calculated. This kind of runtime behaviour necessitates the introduction of an approximation method, the one in focus being KernelSHAP [6]. For this, four different approximations are mainly proposed: LinearSHAP, TreeSHAP, Deep SHAP and KernelSHAP that all aim significantly decrease these runtimes [6]. While these methods were all proposed, in the final testing only KernelSHAP was utilized, as such the focus will be on its functionality.
The goal of KernelSHAP is, to calculate the relevance scores based on a smaller subset of features as well as using linear regression to approximate individual values. For this, KernelSHAP initially cuts down the amount of features to a manageable size and introduces perturbation to copies of these features [4]. Afterwards, a formula can be devised that puts the three desirable properties of SHAP, Local accuracy, Missingness and Consistency, in relation to a relevance value. Then, a weighted linear regression model to these perturbed samples is fitted, creating relevancy scores for each feature [4] [6]. As values for this regression, Linear LIME values are used, as their are equivalent to the approximation of the SHAP-values [6].
To test the newly proposed method, two experiments were devised. Firstly, the both LIME and SHAP were compared to participant's performance in classifying fevers and coughs based on given symptoms. Afterwards the individual performances were presented to the participants, SHAP, LIME and Deep Lift (another studied, but less relevant system), to hand out a budget based on the performance of the individual system. Lastly a smaller scale experiment was conducted on the MNIST digit set. Based on a collection of handwritten digits, both LIME and SHAP created saliency map for a CNN classifying the numbers. This experiment is a common evaluation method for saliency maps, C. Garbin also tested on the same environment [10].

B. Results

1) Results of the user study: The authors [1] of the original paper initially work to answer 3 research questions, whether scores or saliency maps help the user understand CNNs further, and if the presence of either helps the user recognize features in an image. Additionally they also try to gather the level of confidence the participants have about their prediction which, while not an initial research questions, is also evaluated. To measure the results, the users are split into four groups, the baseline being shown neither scores nor saliency maps, the next to groups being shown either technique and the last group being shown to both.

"The utility of saliency maps exists, but it is limited[.]" This is the result stated by [1] after going through the participants predictions. Collecting the probabilities of these, they can be described with a normally distributed function, which is significantly affected by the presence of saliency maps for the user. But it becomes apparent, that the expected value merely rises from 55.1% to 60.7%, with no significant change in the variance. It is also clearly visible, that while users have an almost 80% chance of correctly identifying True Positive cases, they struggle with properly pointing out False Positives (46.9%) and even more so when the example is a False Negative (36.7%). In general the users tend to be overconfident in the CNNs ability and for half of the total examples being incorrect classifications, for 67.3% of the cases, the user assumes the machine to correctly identify the image. Continueing it becomes apparent, that this effect cannot be achieved as significant with scores. While still achieving slight improvements, from 56% of correct classifications without scores to 60%, the authors deem the difference as too insignificant.

Though seemingly able to increase the user's ability to properly classify the CNN's output, the paper clearly states that their confidence in doing so wasn't helped by both saliency maps and scores. Measured at "slightly confident (3)" on a forced 1-4 confidence scale, neither technique managed to increase these values, with again false classifications by the CNN generally causing confusion and lowering the confidence.

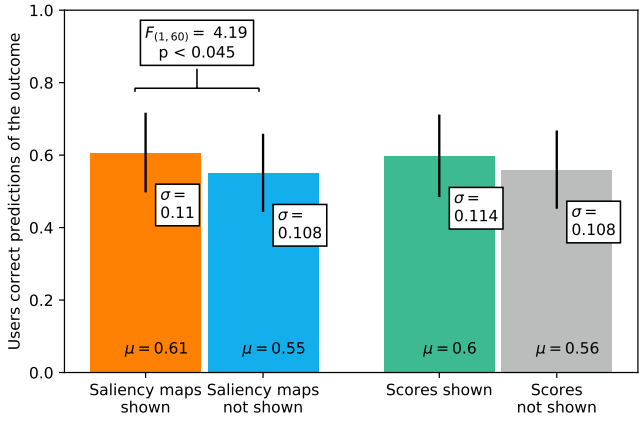

Fig. 4. Users ability to predict the CNNs output, dependant on scores and saliency maps being given. Users with saliency maps perform significantly better, scores increase the performance less [1]

Bigger successes were observed in the research towards named features. To avoid participants naming countless features in an attempt to gather as many as possible, a saliency feature ratio was calculated, displaying the amount of named features being relevant saliency features. In this category saliency maps managed to majorly improve the users ability to name such features, rising from an average ratio of 0.55 to 0.84. This is contrary to the score system, where a clear decrease from 0.73 without scores to 0.66 with scores can be observed. But even with an increased amount of features named, the paper emphasizes, that these features did not necessarily help users predict, seemingly confusing the user in some cases.

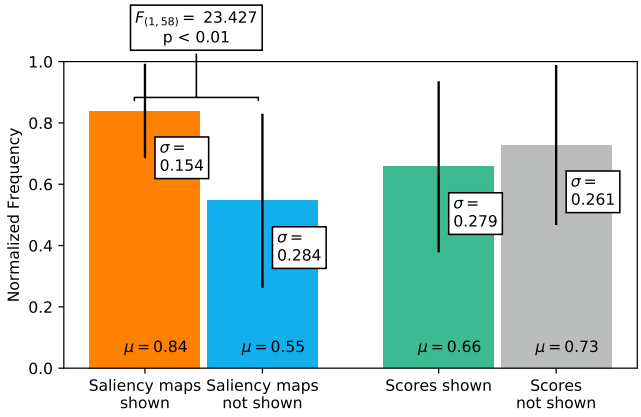

Fig. 5. Users ability to name features, dependant on scores and saliency maps being given. Participants perform clearly better upon being given saliency maps, while scores don't seem to positively affect the results [1]

2) Evaluating different Saliency Map algorithms: Conducting the four experiments discussed in [2], now aims to deliver an algorithm, that manages to increase this performance, via being more accurate in its coverage of the ground-truth masks discussed earlier. Examples of the tests can be seen in Fig. 6

During the first experiment, conducted via using German traffic signs as a distraction from squares in the bottom right

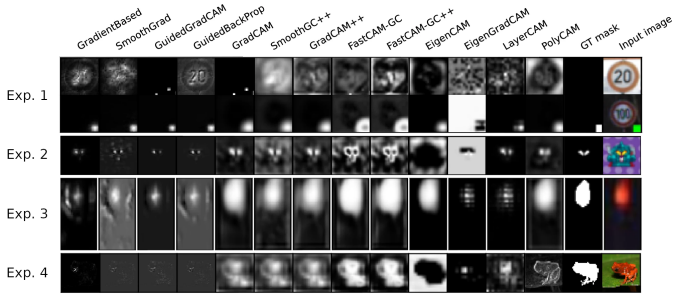

Fig. 6. Performance of different algorithms during each experiment. For simplification purposes, only one test per experiment is shown. All of the experiments shown are under VGG16 architecture [2]

corner of the screen, the CNN behind the saliency map predicted the result with a 99% accuracy, meaning all m_{gt}-scores are solely based on the saliency map algorithm's performance, not potential misclassifications. Evaluating this quite simple example, most algorithms managed to gain a high accuracy, most m_{gt}-scores being far above 70%. This excludes the notable exception of FastCAM and FastCAM++, algorithms not named specifically earlier. Their poor performance streaks through all experiments; as such their output won't be further relevant. While EigenGrad-CAM achieved best scores in the first experiment, this is a win by a small margin, as given by the easy task.

Continuing to the second experiment, the heroes and villain cartoon drawings, being defined by their eye color, we quickly see GradientBased, SmoothGrad, Guided Backprop and GuidedGrad-CAM achieving consistently good results. This is again due to their raw performance, as in Experiment 2 the average accuracy is again over 92%, even going up to 99.95% for the third experiment.

Furthermore the experiment in 3 also benefits the same algorithms again. This is mostly, as other saliency map algorithms will often focus on the border or contours of an object, via this creating quite usable maps, that fail to achieve good scores on the ground-truth based comparison system though. This streaks into the 3rd experiment, with the traffic lights being picked up by almost all algorithms, but blurriness again hinders algorithms from performing at full level. Following the results of the second experiment though, all four algorithms previously managed, perform quite admirably during this task as well, solidifying their position as worthy alternatives.

In the last example, during the process of tracing actual objects, the CNN itself performes a lot worse for the first time, dropping to a medium accuracy of around 75% and further lowering m_{gt} values further. This, as well as the more complex nature of the problem, had the m_{gt} values drop significantly. Again both Gradient Based and Grad-CAM perform consistently well, but surprisingly besides the more niche LayerCAM and the initially underwhelming FastCAM performed well. Furthermore PolyCAM, while achieving lower m_{gt} values, managed to highlight the contour of relevant objects incredibly well, but failed in actually marking the object itself.

Concluding, the algorithms performed quite different depending on the task and CNN architecture, as such there is no clear winner or objectively most performant algorithm, but it becomes apparent that GradientBased, SmoothGrad, Grad-CAM and PolyCAM perform on a consistently high level. Still the outputs of algorithms like EigenCAM, EigenGrad-CAM, LayerCAM and PolyCAM, while not performing well under the given measuring system, still manage to often give insightful information and while their individual performance might not be high, their diverse results often help understand a CNNs process better.

3) Performance of saliency maps in medical environments: To properly evaluate the performance of the saliency maps, it is necessary to look at their individual performance for both Diabetic Retinopathy and neovascular age-related muscular degeneration. For DR, the CNNs picked up on different symptoms of the disease, depending on both the algorithm used and which CNN in the Deep Ensemble was used. This shows in SmoothGrad preferring to pick up on optic discs and soft or hard exudates, while Guided Backprop preferred microaneurysms and hemorrhages. Furthermore the individual CNNs of each Deep Ensemble also tended to focus on different symptoms, with in one example different machines would pick up on only microaneurysms, while others also utilized hemorrhages, soft exudates as well, showing the clear effect of different starting values for the individual network. In general the CNNs picked up on smaller symptoms more frequently, with large exudates or bigger hemorrhages being picked up a lot less. Generally thought, most features were, while picked up, not properly covered by the saliency map and only partially marked.

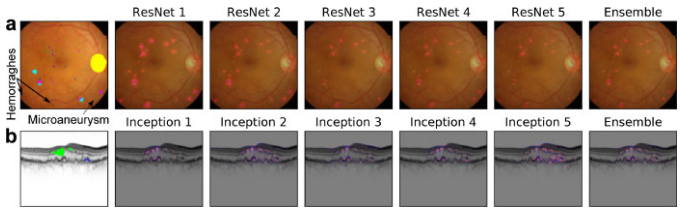

Fig. 7. Merging of saliency maps. The ensemble still shows the intricate details of the individual maps, while not appearing sparse or blurred [3]

Continuing, the post processing method, introduced by the authors of [3] was then also evaluated. Initially, the results of the Deep Ensembles when the maps were just overlayed on each other ended up being sparse and not highlighting the spots of interest strong enough. The post processing method now combats this and excels at concentrating on and growing the highlight spots of each saliency map, even if the highlighted feature is not picked up by every map. Consequently its tendency to pick up small features benefits algorithms that concentrate these symptoms initially, promoting as an example the Guided Backprop algorithm, reducing the difference between its dice loss to lessen the gap towards SmoothGrad in the first experiment.

In the second experiment, analysing nAMD symptoms, which is commonly diagnosed via small batches of sub- or intraretinal fluids in the back of the eye. Contrary to the quite diverse ways that the CNNs diagnosed DR in the first experiment, DNNs and their respective saliency maps diagnosed on a more uniform set of symptoms, while again preferring smaller features over bigger batches of fluid. As such, most machines mostly focussed on subretinal fluid instead of intraretinal, as the former appears in quite smaller amounts. Additionally to expected detected symptoms, the saliency maps showed, that CNNs also used the fovea, a point inside of the Macula where the most photoreceptor cells lie, to diagnose the eye disease. As this was not intended, the Dice-loss values decreased for all saliency maps. Still, Guided Backprop emerged as the

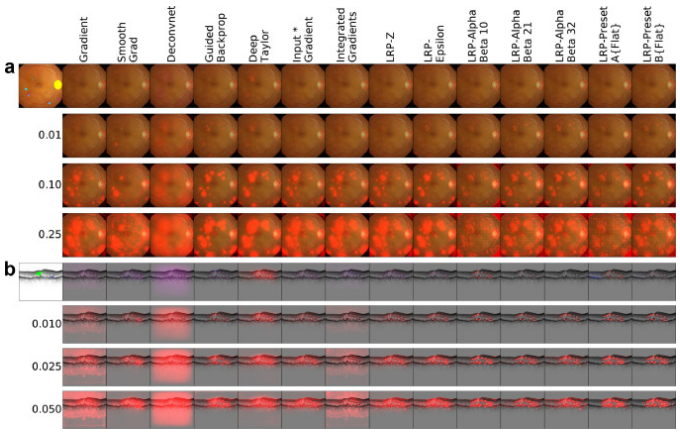

Fig. 8. Behaviour of the post-processing method dependant on the scaling factor. Adjusting it especially grows small parts of the saliency map, but can invalidate smaller details [3]

most clinically useful algorithm with SmootGrad performing worse, but still bringing forth acceptable results with slightly different results. Besides the CNN picking up characteristics of the fovea, the Dice-loss was also increased, as part of the training set was incorrectly classified. In the expert annotated scans these images were correctly classified and interestingly enough, the saliency maps pick up on some of the very small amounts of fluids used for this diagnose. These small spots combined with the rest of small batches of subretinal fluid were again enhanced by the post processing method, with it correctly identifying relevant spots across the different saliency map and growing small spots of interest.

In conclusion the author name Guided Backprop as the clinically most viable algorithm, consistently performing on a high level, even though it is insensitive to some symptoms. The usage of Deep Ensembles also proved to be successful, with individual CNNs deciding on their own set of symptoms and increasing the amount of captured symptoms as well as the quality of probability predictions significantly. Lastly, the post processing method proved to be a legitimate algorithm, highlighting, emphasizing and growing relevant spots on maps, with a tendency to benefit smaller features.

4) Performance of Shapley Approximation: For both experiments involving participants, SHAP consistently aligned to the judgments given out by users, both in the medical environment as well as when handing out budgets. MNIST experiments were mostly not analysed, mostly paying attention to the algorithms ability to properly point out the differences between 3s and 8s, with SHAP correctly highlighting the differences [6]

The final product of implementing KernelSHAP and other SHAP variations have been available ever since initial proposing as a Python implementation, offering JupyterNotebook entries for implementations, tests and examples. This accessible form of publication has since then inspired a significant amount of work utilizing KernelSHAP, but, due to the amount of features in image-recognition, almost all of these experiments are based on non-image applications. An exception to this are the experiments conducted by C. Garbin, in which he further tests the performance of SHAP. This is measured upon the MNIST handwriting dataset, evaluating SHAP's performance for both a high accuracy dataset (97%) as well as a more inaccurate network (87% accuracy). In both experiments, the SHAP methods used achieved highly intuitive solutions, highlighting how the network distinguished numbers based on missing features, correctly marking missing features and conveying a clear understanding of the topic [10].

Fig. 9. SHAP classifying an instance of 8 from the MNIST dataset. In comparison is also the 3. Negative relevance is clearly visible for the classification of 3, pointing towards points that clearly make out the 8 [10]

IV. Discussion

A. Effect of saliency maps

The effect of saliency maps is clear. Providing insightful and effective results, the authors of the user study [1] first proved their overall capabilities in helping users predict CNN outputs. Furthermore the ground-truth based tests conducted [2] also showed the potential of different algorithms, showcasing a broad amount of algorithms with high reliability and some more experimental algorithms with very unique use-cases.

B. Low results initially measured in the user study

But the effect observed in the initial paper [1] is underwhelmingly small with only 5% more users actually predicting the correct result, even though the quality of saliency features named by users increased significantly. So why is the measured effect of saliency maps so small in this case? One possible explanation is simply that of a wrong measuring metric being used. The strength of saliency map comes in gaining an insight on its previous predictions, gaining insight on the features used by a CNN to predict and helps spotting errors that could be fixed with dedicated training. This can be seen in the experiments conducted [2] with the saliency maps clearly pointing out the eyes of the villain cartoon figurines and traffic lights, while also highlighting the weaknesses of the nAMD detection CNN, that also used the fovea to diagnose said disease, even though this was completely unintended. But this is only indirectly being picked upon by [1] as their paper is more interested in user's predictive capabilities. Prompts like "Can users detect if a CNN is sufficiently trained by looking at its generated saliency maps?" or "Can participants spot unintended feature classification by a CNN?" would generally focus more on these analytical strengths of saliency

maps. And while these study goals would substantially differ from the original prompt, it would be meaningful research to counter the participants obvious struggles in classifying false classifications as observed in the initial user study [1].
Another, more casual use-case of saliency map algorithms would also be the simple double-checking of results generated. While the authors of [1] intend on analysing saliency maps influence on non-ML-experts, their study standards are still quite high (participants with technical backgrounds) while they also propose conducting future experiments with even higher standards, but it has to be mentioned, that this defect also doesn't get addressed by the earlier named prompts. Instead another direction, future research could be headed into, would be more simplified studies to measure non-tech-users ability to utilize saliency maps to double check CNN outputs. Especially with both Apple introducing AI features in newer versions of IOS and Samsung marketing their new devices almost exclusively with AI features, it is important to see if these day-to-day decisions could be improved upon via presenting these users explainable AI features. Equally important would be the willingness of users to work with these tools; a applicable experiment for this could have users find the cause for misclassification of a given image and either select or take a new photo avoiding the irritating feature.
In combination with saliency maps, the utilization of score-based systems was also attempted, but failed. Neither did scores increase the participant's accuracy in predicting the CNN's results by a significant margin nor did it help them identify more saliency features. But this should not come as too much of a surprise, with the task being completely visual and the score system counter-intuitively being the exact opposite. From observing the scores alone, the user gains no insight on why the CNN recognized these features, rather only seeing probability scores generated by only one CNN. A deep ensemble based score system could help this experiment somewhat via producing much more realistic probability scores [3], but would probably still not increase the merits of scores to a significant margin.

C. How should saliency maps be used?

While the base paper [1] refrains from using multiple saliency map algorithms, related works [2] [3] use a broad arrangement of these. It quickly becomes apparent, that there is no clear winner for either an algorithm nor an architecture, but there is still preferable algorithms. In both papers algorithms like SmoothGrad and Guided BackProp perform on a consistently high level with many others also being viable options. A lot of the used algorithms also bear quite unique use cases, with PolyCAM and LayerCAM producing situational but interesting results in earlier tests [2] and some LRP-variations performing well in the nAMD-detection environment. Contrary, the deep ensembles utilized in a medical environment [3] perform admirably well, both creating more realistic probability scores, showcasing more differentiated saliency maps and doing all of this with the only major drawback being increased computing in both machine training and execution. The post processing method proposed also performs on a high standard, its highlighting of individual spots also avoiding sparse saliency maps, another possible drawback of deep ensembles. But both papers struggle with establishing proper metrics for their evaluation, both terms of fairness for the algorithms as well as accuracy.
The ground-truth masks utilized require the object to completely fill out said map, which in turn lowers ranking values of algorithms that would rather focus on outlines of objects. This can be observed while looking at the PolyCAM results in later tasks [2]. LRP-algorithms were also selected in the user study for their nature of highlighting object's contours rather than mark them in whole, as LIME does. Even though this kind of highlighting was preferred by users, LRP-algorithms would have performed badly in the ground-truth based tests [1][1].
During the evaluation in a medical environment, the dice-loss was again calculated dependant of how an algorithm would highlight the given features, potentially worsening some of the results for the utilized LRP-algorithms. Additionally the A generalised approach to displaying saliency maps to users could now be a more interactive solution: Allowing the user to switch to more experimental algorithms, while initially displaying more reliable ones, as well as allowing them to scale post processing methods described in the medical-analysis paper [3] could allow users to familiarize themselves more with a topic that has no best method of approach, but rather requires individual understanding.

D. SHAP

SHAP continues to be a relevant procedure for explainable AI. Its results are some of the most significant and logical, giving far-reaching insights of some of the less obvious relations between objects in an image, relations that even the most advanced saliency map algorithms won't pick up on. But this is not due to the fact, that SHAP utilizes some highly efficient algorithm or bringing forth advanced analytical algorithms, but rather via the sheer amount of information given, that other saliency map algorithms simply do not have access to. On the other hand, SHAP variations like TreeSHAP try to cut down on this runtime, utilizing less and less subsets. But TreeSHAP does not feature the same intuitive results as KernelSHAP. This creates a careful balance between keeping the runtime of SHAP somewhat low, but also not losing its intuitive design. With its runtime far beyond the acceptable spectrum for day-to-day tasks, KernelSHAP will for now find no application in use-cases involving double checking outputs or the use alongside the CNN to just highlight relevant regions for a user. The unique capabilities of SHAP still hold its merits though and, while most use-cases don't justify the amount of resources being used, high-stakes applications could still benefit from these. A possible task for SHAP could be the analysis of CNN failures, trying to find unintended behaviours, as done with saliency maps in [3], as a more or less last

[1]Mentioned as an observation during the pilot study

resort tool with high amounts of reliability. But the technology itself just ends up being a dead-end. Neither parallelisation, nor optimizations in the runtime change the fact, that the base behind Shapley values is brute-force with an exponential amount of subsets, stopping it from ever becoming viable on a grand scale or trading off some of that runtime to diminish its capabilities.

E. Designing standards suitable for a medical environment

Even though ResNet50 or ImageNet applications are regularly used in saliency map environments [1] [2] [3] [5], this adheres to no standard, but rather a matter of broad usage. But especially for medical appliances, with trust and reliability being necessary to protect both clinics and patients, some kind of standard will need to be generated for the future. And while standards regarding individual algorithms or architectures might be hard to formalize, the authors of [3] showed clear benefits of using both post processing methods and deep ensembles. A possible standard could involve required accuracy scores, as well as requiring ML-experts to analyse the saliency maps, generated via deep ensembles, to meticulously search for unintended characteristics being used to diagnose, requiring deep ensembles to pick up each individual feature instead of avoiding certain features like observed in experiments conducted with DR and nAMD [3] (larger features like intraretinal fluid for nAMD). Furthermore the creation of a proper dataset will be required. To train their datasets, the authors of [3] use scans and images from the "University Eye Hospital Tübingen"[2], but it is clear that individual hospitals might not have the resources or permission to create these datasets. To open the possibilities for these hospitals to work with CNNs, utilizing reliable and capable networks, a unified and international dataset must be created and made available for these hospitals.

V. Conclusion

Saliency maps are one of the most unique techniques to analyse image-recognition AI. The widespread implementation of these tools still faces challenges, both because of the situational nature of most algorithms, performing well on very specific tasks while failing to properly solve others, as well as the lack of defined standards for the entire field. Future research into the capabilities of deep ensembles, post processing methods and the combination of saliency map algorithms must first be done, before any breakthrough in usability can happen. All these deficits must not distract from the already achieved goals though, with saliency maps proving to help users understand Neural Networks, performing admirably well in medical environments and achieving consistently accurate results in most tasks. And while the topic of saliency maps appears to be niche, work done on their field will only increase in relevance in the future, paving the way for reliable AI systems, that both protect manufacturers and consumers.

[2]see "2.1. Diseases and datasets", [3]

Acknowledgment

As there is no clear best saliency map algorithm during the creation of this paper, it would be unnecessary to explain every algorithm in detail. Besides some highly relevant base algorithms like Grad-CAM or LRP, most names are mainly used like a pseudonym. Their individual implementations and intricacies are not relevant for this paper, but the general diversity of algorithms is.

References

[1] A. Alqaraawi, M. Schuessler, P. Weiß, E. Costanza, N. Berthouze "Evaluating saliency map Explanations for convolutional neural networks: a user study" UCL Int. Cen. London, Tec. Uni. Berlin, February 2020

[2] K. Szczepankiewicz, A. Popowicz, et al. "Ground truth based comparison of saliency maps algorithms" University of Technology, Warsaw, October 2023

[3] M. Seçkin Ayhan, L. Benedikt Kümmerle, et al. "Clinical validation of saliency maps for understanding deep neural networks in ophthalmology" Uni. of Tübingen, Helmholtz Cent. Munich, January 2022

[4] C. Molnar "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable", Chapter 9.6 "SHAP (SHapley Additive exPlanations)", February 2022

[5] Image examples - SHAP, https://shap.readthedocs.io/en/latest/image_examples.html, accessed 16.07.2024

[6] Scott M. Lundberg, S. Lee "A Unified Approach to Interpreting Model Predictions", University of Washington, Seattle, May 2017

[7] W. Samek, "Layer-wise Relevance Propagation", https://www.hhi.fraunhofer.de/en/departments/ai/technologies-and-solutions/layer-wise-relevance-propagation.html, accessed at 16.07.2024, Heinrich-Hertz-Institute, Berlin

[8] C. Molnar "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable", Chapter 10.2 "Pixel Attribution (Saliency Maps)", February 2022

[9] C. Molnar "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable", Chapter 9.4 "Local Surrogate (LIME)", February 2022

[10] C. Garbin"Exploring SHAP explanations for image classification", https://cgarbin.github.io/shap-experiments-image-classification/, April 2021