A comparative study of explainer modules applied to automated skin lesion classification

Jia Sun, Tapabrata Chakraborti, and J. Alison Noble

Department of Engineering Science, University of Oxford, UK
jia.sun;tapabrata.chakraborty;alison.noble @eng.ox.ac.uk

Abstract. How to choose the appropriate explainer module for a deep network for interpretability of learned features through visualization? Are the selection criteria specific to a task or can they be generalised? We investigate a set of criteria by which to evaluate and select explainer modules for deep learning based classification. This is of great importance for applications with high human consequence like healthcare, where it is of utmost importance for the automated decision making process to be aligned with clinical expert knowledge. We choose skin lesion classification as the representative classification task. and select three off-the-shelf popular explainer-visualizer modules: LIME, Grad-CAM and Kernel SHAP. We compare these modules on a baseline vanilla CNN model and evaluate them based on several criteria like consistency, fidelity, sensitivity and relevance. The results bring out several interesting insights and are presented with detailed illustrative diagrams. [1]

Keywords: skin lesion classification · explainable artificial intelligence · interpretable machine learing · class activation maps · medical imaging

1 Introduction

Deep learning based vision systems have made giant strides in automated medical image analysis in recent years, due to smarter algorithms, faster computing and increased memory resources. Though the performance of these systems have improved by leaps and bounds over a short span of years, the associated increase in the design complexity of these models have made it difficult even for their designers to explain the decision making process. The need for interpretability in deep learning is crucial for such methods to be trusted in application of high consequence like medicine and healthcare. While recent research in medical imaging [4] has delved into the open problem of transparency of deep networks, there is yet to be a standard set of criteria and tools that a machine learning engineer might refer to while using off-the-shelf explainer modules. This is exceptionally important since the absence of a benchmark in interpretability will further obfuscate the goal of explainability. In fact, it might make the situation worse, if

TC and JAN are funded by UK EPSRC grant EP/M013774/1 (Seebibyte)

the results of explainer modules [20] are at odds, or at least it is unknown why they might differ under varying circumstances.

A case in hand is automated skin cancer classification from visual data. According to the world health organisation [1], between 2 and 3 million non-melanoma skin cancers and 132,000 melanoma skin cancers occur globally each year, which is one in every three cancers diagnosed. However, the European Union's General Data Protection Regulation (GDPR) and the UK's related Data Protection Bill both require a "right to explanation" for any automated decision-making algorithms [18]. Although this right is not legally binding, automated diagnostic systems being developed are likely to face more scrutiny over their explainability. Indeed, for people to put their trust into an automated diagnosis, there is a need for an explanation as the consequence of a misdiagnosis can be catastrophic [15].

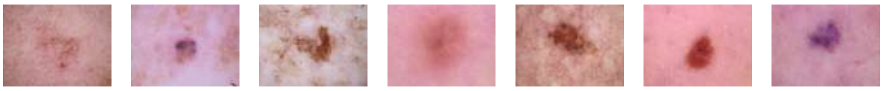
The contribution of this paper is that it formalises the criteria on which to choose explainer modules for different scenarios and for the first time experimentally demonstrates the effects choosing skin lesion classification as the representative application [9]. This will help a machine learning practitioner to confidently select available explainer modules and interpret the results better, which is the call of the hour in interpretable machine learning.

2 Experimental Setup

In this section, we introduce the dataset and deep architecture used in this work. We also describe the data preparation process and training protocol.

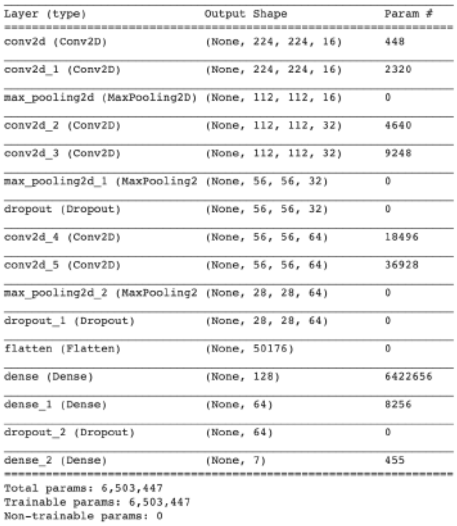
2.1 Dataset and data preparation

HAM10000 dataset [16] is used in this work since it is the most commonly used benchmark dataset by the research community for skin lesion classification. The dataset contains, a total of 10015 RGB dermoscopic images of dimensions $3 \times 450 \times 600$ distributed over 7 classes namely: melanoma (Mel, 1113 samples), melanocytic nevi (NV, 6705 samples), basal cell carcinoma (BCC, 514 samples), actinic keratosis and intraepithelial carcinoma (AKIEC, 327 samples), benign keratosis (BKL, 1099 samples), dermatofibroma (DF, 115 samples) and vascular lesions (VASC, 142 samples). The main challenges of this dataset are class imbalance and the presence of artifacts like dark patches, skin hair, etc.



(a) akiec (b) bcc (c) bkl (d) df (e) mel (f) nv (g) vasc

Fig. 1: Examples of 7 classes of pigmented lesions in the HAM10000 dataset



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 224, 224, 16)      448       
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 224, 224, 16)      2320      
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 16)      0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 112, 112, 32)      4640      
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 112, 112, 32)      9248      
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 32)        0         
_________________________________________________________________
dropout (Dropout)            (None, 56, 56, 32)        0         
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 56, 56, 64)        18496     
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 56, 56, 64)        36928     
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)        0         
_________________________________________________________________
dropout_1 (Dropout)          (None, 28, 28, 64)        0         
_________________________________________________________________
flatten (Flatten)            (None, 50176)             0         
_________________________________________________________________
dense (Dense)                (None, 128)               6422656   
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256      
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0         
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 455       
=================================================================
Total params: 6,503,447
Trainable params: 6,503,447
Non-trainable params: 0
```

Fig. 2: CNN Architecture.

The data is first split in a ratio of 90:10 into the training and test data. The training data is further split in a ratio 90:10 in to training and validation data. It is worth noting that the validation and test set both have the same proportion of each class as the original dataset. Then the images are resized to 224 x 224 pixels, and RGB values are normalised between 0 and 1. All classes except nv are augmented by rotation, flipping, shifting and zooming to yield 2500 images in each training class. This yields almost double the number of training data examples (19389 images). The HAM10000 has multiple images from different angles and lighting of the same lesion, which are all kept out of the validation and test set to avoid false high accuracy due to data leakage. No colour constancy or hair removal algorithms were used since they will make the training images more homogeneous and defeat the purpose of discovering whether CNNs ignored these features that can cause spurious correlations.

2.2 Deep Network

The CNN model is constructed loosely based on a baseline vanilla CNN network described in [9]. The basic structure is two convolutional layers followed by a max-pooling layer. The first two layers contain less filters to reduce the training time. The network here also includes more dropout layers, less fully connected neurons to combat the overfitting present when replicating the original network. Furthermore, it is modified for a seven class classification instead of a two class one. Every layer uses an ReLU activation function except for the last dense layer which uses the softmax activation. The resulting architecture is as shown in Figure 2. The number of trainable parameters is around 6.5 million. For comparison, the VGG16 network contains 41.5 million and ResNet50 26.7 million.

2.3 Training

The training loss is weighted inversely proportional to the number of images in each class in order to further compensate for the highly unbalanced training data. The model is trained using the Adam optimiser [7] for 100 epochs with an initial learning rate of 0.001. The learning rate is halved when the validation accuracy did not improve for three epochs.

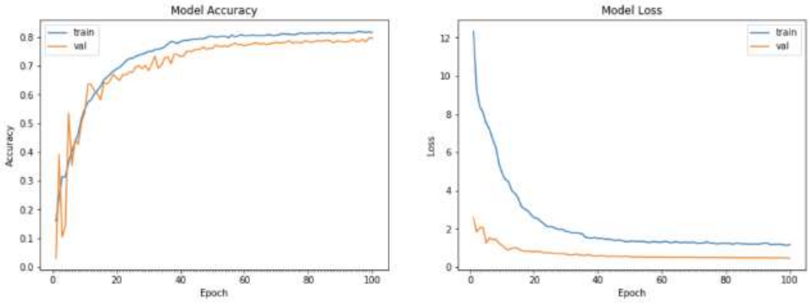
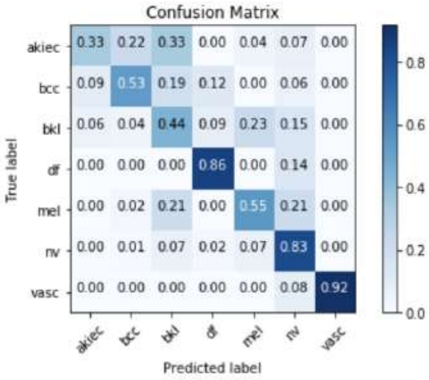


Fig. 3: Training accuracy (a) and validation loss (b) over 100 epochs.

As shown in Figure 3, validation loss and training loss both plateaued around the same number of epochs where the training accuracy is consistently higher than the validation accuracy. The training loss did not improve at a faster rate than the validation loss beyond the 60^{th} epoch. Note that the higher overall training loss is due to the fact that the loss is weighted but not normalised. The model which yields the lowest validation loss is saved during training.



	akiec	bcc	bkl	df	mel	nv	vasc
akiec	0.33	0.22	0.33	0.00	0.04	0.07	0.00
bcc	0.09	0.53	0.19	0.12	0.00	0.06	0.00
bkl	0.06	0.04	0.44	0.09	0.23	0.15	0.00
df	0.00	0.00	0.00	0.86	0.00	0.14	0.00
mel	0.00	0.02	0.21	0.00	0.55	0.21	0.00
nv	0.00	0.01	0.07	0.02	0.07	0.83	0.00
vasc	0.00	0.00	0.00	0.00	0.00	0.08	0.92

Fig. 4: Class confusion matrix.

3 Explainer Modules

In this Section, we briefly describe the three explainer modules (LIME, Grad-CAM and Kernel SHAP) that we have selected to compare in this work, based on their popularity in relevant literature.

3.1 LIME

LIME (local interpretable model-agnostic explanations) [11] aims to strike a balance between interpretability and model fidelity by minimising the following equation:

$$\xi(x) = \underset{g \in \mathcal{G}}{\operatorname{argmin}}\, L(f, g, \pi_x) + \Omega(g) \tag{1}$$

where f is the black box model; g is the explainable model used, in this case a ridge regression model; and $\Omega(g)$ is a measure of complexity of the explainable model; π_x is the kernel function that transforms an array of distances into an array of proximity values between input x and sampled instances in Z:

$$\pi_x(z) = \sqrt{e^{\frac{-d^2}{\sigma^2}}} \tag{2}$$

where d is the distance between the sampled instance and the input and σ is a parameter for the width of the kernel. The kernel function is then used to weight the loss function between the original model and the linear model:

$$L(f, g, \pi_x) = \sum_{z,z' \in \mathcal{Z}} \pi_x(z)(f(z) - g(z^{'}))^2 \tag{3}$$

To put it more intuitively, in order to approximate a non-linear model without making any assumptions about it, the algorithm takes samples of perturbed instances z and $z^{'} \in \mathcal{Z}$ around input x and its explaninable version $x^{'}$. Then use the original CNN model to generate a prediction target $f(z)$ to train the explainable model $g(z^{'})$ using the weighted loss function from equation 3.

When applied to a CNN model, LIME regards an explainable version of an image to be a binary vector containing zeros and ones. The image is first segmented into superpixels, which are groups of pixels with common characteristics like intensity or colour. Hence zero means that the superpixel is "switched off" and one means that it is "switched on". The perturbed instances are obtained by using a fair coin toss to generate sets of zeros and ones, resulting in images with different missing patches. The features (i.e. superpixels) with the highest positive coefficients in the linear model contributes the most to the predicted class and vice versa.

As a demonstration, Figure 5 is generated using the open source LIME implementation with its default parameters. Some drawbacks can be quickly spotted. First there is no indication of the degree of influence that a superpixel can have on the final prediction. If the third image is directly displayed, one cannot be certain that the green patch covering the lesion contributed the most. The number of superpixels included in the explanation is also arbitrary, dependent on the size of the lesion. Moreover, there are the many parameters that need to be chosen heuristically, like the segmentation method, kernel width and distance metric. The default setting here seems to give a reasonable coverage of the lesion, but further investigation is needed on lesions with more complicated pigmentation networks.

3.2 Grad-CAM

Grad-CAM (gradient class activation mapping) [12] is a generalised version of CAM. In CAM, the method is designed for a specific type of CNN architecture where the global average pooling layer directly feeds into a softmax layer. Whereas in Grad-CAM, any convolution layer can be examined by first calculating the gradient using back propagation then using global average pooling to assign weights to each feature map output in that layer (Eq. 4).

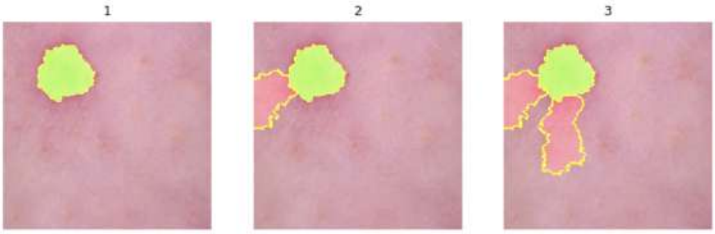


Fig. 5: LIME with 1, 2 and 3 superpixels. Green means positive contribution.

$$a_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{4}$$

$$L_{Grad-CAM}^c = ReLU(\sum_k a_k^c A^k) \tag{5}$$

In the above equation, Z is the number of pixels, y^c is the class score and A_{ij}^k is the feature map activation of feature map k. The feature map outputs can now be weighted and summed before being passed through a ReLU function (Eq. 5). The ReLU function makes sure that only positive contributions to the class are displayed. The level of detail in these saliency maps are determined by the convolutional layer examined (figure 6). The closer the layer is to the fully-connected layers, the more accurate these maps are, but also more blurred.

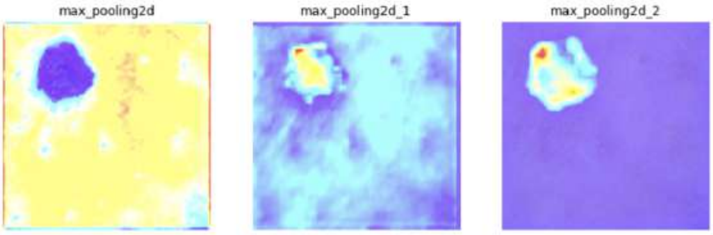


Fig. 6: Grad-CAM from the three max pooling layers in the network.

3.3 Kernel SHAP

Kernel SHAP [8] is LIME plus "Shapley additive explanations". Here, "kernel" refers to the kernel function π_x in equation 3. The difference being that here it is modified to retrieve Shapley values [13] (equation 6) and used to weight the same loss function in equation 3.

$$\begin{aligned} \Omega(g) &= 0, \\ \pi_x(z) &= \frac{M-1}{\binom{M}{|z'|}\,|z'|\,(M-|z'|)}. \end{aligned} \tag{6}$$

Here M is the number of simplified features obtained using a debiased LASSO [10]. The rest of the algorithm works mostly the same as LIME. The theoretical benefits of using Shapley values are extensively explained in the original paper [13]. For this particular application, the immediate improvement from LIME is that there are significantly fewer parameters to fine tune; and the explanation covers the whole image with clear indication of degree of influence. Both characteristics make it easier to compare different explanations against one another (Figure 7). The main downside is that calculating Shapley values are expensive and slow: with the same number of samples taken, kernel SHAP on average took twice as long as LIME.

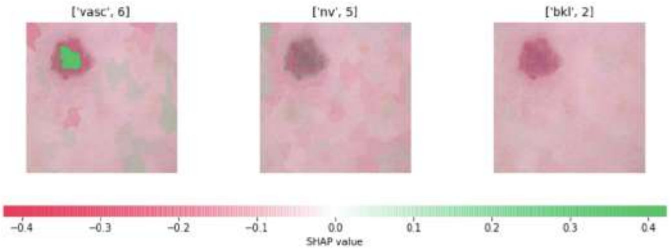


Fig. 7: Kernel SHAP for the top three predictions, taking 1000 samples.

4 Evaluation

There has been mention in recent literature regarding the importance of investigating fidelity, consistency, and sensitivity of deep learned features [19][3]. However, there are currently no standardised metrics to evaluate explainability of deep networks. Furthermore, for medical applications, the clinical relevance of such methods are arguably equally important. In addition, given the model-agnostic and post-hoc nature of the three explainer modules chosen in the current work, they can be examined through grouping similar instances together to gain an intuition regarding their behaviour.

4.1 Consistency

Consistency means that the results can be reproduced under same experimental conditions. LIME and kernel SHAP are both sampling based methods and have multiple parameters that need to be determined depending on the data and the application. These parameters have the most effect on the consistency of the explanations and subsequently the ability to accurately test the fidelity and sensitivity in later sections.

The original open source implementation of LIME has a number of parameters that the user can toggle to best suit the application, while kernel SHAP has a more rigid implementation. Ideally, LIME parameters will match kernel SHAP as close as possible. For example, both methods use segmented superpixels as features. In LIME, the default segmentation algorithm is quickshift [17], while kernel SHAP uses SLIC [2]. In order to compare the two methods, they have to use the same set of features generated by the same segmentation algorithm. However, as findings in [14] show that for LIME, the SLIC segmentation resulted in the lowest weights compared to quickshift and Felzenszwalb [5] when images are segmented into the same number of superpixels. This can create problems in terms of consistency since lower weights are more sensitive to small changes between two sampling sessions. When quickshift is used, the saliency maps generated did not change when repeated multiple times with 1000 samples. But when SLIC is used, the superpixel explanations changed during multiple runs even when 5000 samples were taken (Figure 8).

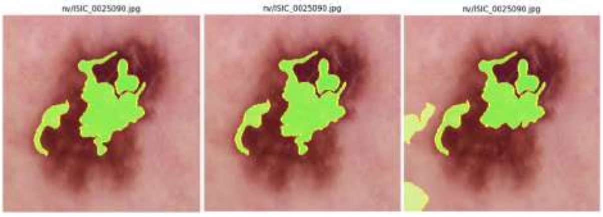


Fig. 8: LIME saliency maps using 100 SLIC superpixels with 5000 samples.

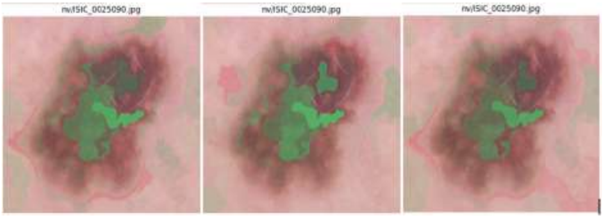


Fig. 9: Kernel SHAP saliency maps using 100 SLIC superpixels, 5000 samples.

This is the same for kernel SHAP (Figure 9), though visually the change is less noticeable. Regions within the pigmented lesion with high Shapley values do not go from having a positive value to a negative one. Regions in the surrounding skin area with low weights do change. Of course the number of samples taken can be increased even higher to reduce the inaccuracies of the linear approximation but that would increase the computational cost. Nevertheless, making the size of the superpixels larger can also guarantee a more stable result. This is because smaller superpixels do not cover enough area in the lesion for them to capture much useful information for classification. As a result, the weights assigned to them become small. Reducing the SLIC segments to 50 in this case have shown to improve consistency when taking 5000 samples for both LIME and kernel SHAP while preserving enough details in pigmentation, shape, etc (Figure 10).

The other important parameter to set is the feature selection method used. The default method used in LIME so far is referred to as "highest weights". It selects the highest product of absolute weight times original data point when learning with all the features. It is quite obvious that this method is not ideal for lesion classification because it will give a massive weight boost for light, non-pigmented superpixels with high values for data points and therefore potentially skew the result. LIME can match kernel SHAP and use LASSO feature selection [10] but it negatively impacted the consistency when all other parameters are kept the same. This is due to the fact that LIME fits the features to a ridge regression model for approximation, which already has built-in regularisation. An extra step of feature selection will reduce the weights even lower and make them unstable. Kernel SHAP uses a weighted least square loss function and hence the feature selection beforehand is necessary. Other options include "forward selection", in which features are added iteratively until the addition of a new feature does not decrease the loss. However, this process is order sensitive and hence inconsistencies are still present in repeated runs. Thus, all features are selected when LIME is used to ensure maximum stability of the result. A larger feature set supposedly will increase the computational cost but the increase observed is far less than the cost of increasing the samples taken.

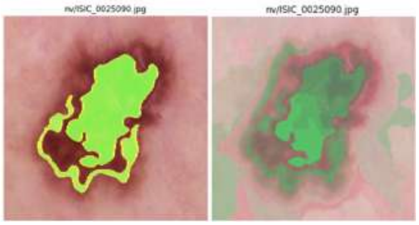
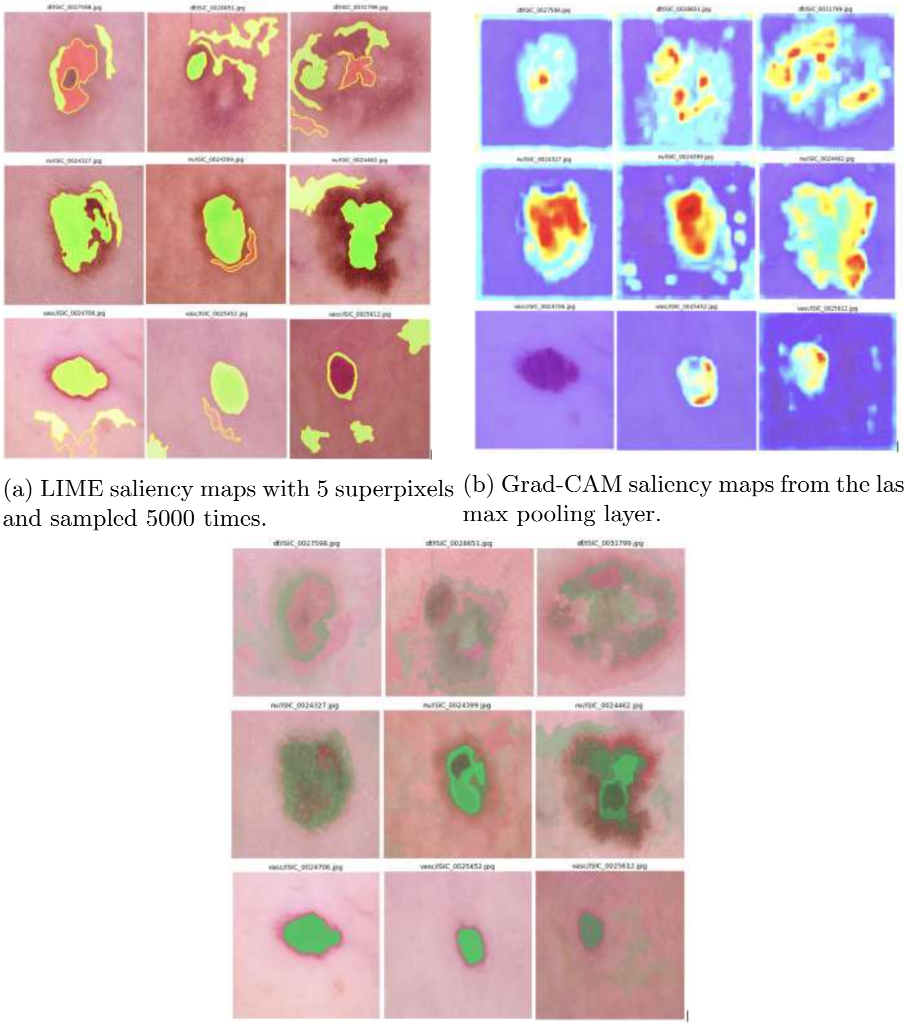


Fig. 10: LIME and kernel SHAP saliency maps generated using 50 SLIC superpixels with 5000 samples.

Various distance metric can also be tested for LIME. In the original paper, the authors suggested using cosine distance for textual data and Euclidean ($L2$) distance for images. When tested, cosine distance generated far more consistent results than euclidean distance. Euclidean distance might have performed well for low dimensional datasets but in this case with 50 features, this no longer holds true. As shown in Figure 16 cosine distances also gives slightly higher feature weights overall and ensures more stability. Lastly, inconsistencies can still occur using these optimised parameters. In those cases, the inconsistency itself signals that a larger area in the image is contributing to the classification (e.g. a large lesion). When this happens, selecting a suitably larger number of superpixels included in the explanation becomes more important.



(a) LIME saliency maps with 5 superpixels and sampled 5000 times.

(b) Grad-CAM saliency maps from the last max pooling layer.

(c) Kernel SHAP saliency maps with 50 superpixels and sampled 5000 times.

Fig. 11: Examples of saliency maps generated in classes with high accuracies and images with high prediction confidence in order to test the fidelity.

4.2 Fidelity

Fidelity commonly refers to the ability of an explanation method to approximate the black box model correctly. It is usually tested by perturbing the model through random initialisation of weights and checking how the explanation changes with it. This is true for all three methods that are being examined. However, it is an insufficient confirmation on the fidelity of these methods. When an explanation is generated, there are three possible reasons for its quality: whether the method used is truthful to the model; whether the model has a high accuracy; or a combination of the two.

Low accuracy means that the model has not learned the underlying data representation properly and the associated explanations will thus be less meaningful and harder to understand. This model has a balanced accuracy of 0.64, but individually some classes have a higher accuracy and some are lower. Average model performance can be artificially increased if only classes with high accuracy are chosen. Among these classes, predictions with high confidence are used to generate saliency maps, to ensure that the chosen instances are highly representative of what the model believes to have the characteristics driving its prediction. Then the only variable in the quality of these explanation is the fidelity of the methods. Therefore, any test on these methods in this section will prioritise images belonging to class df, nv and vasc with at least 99% prediction confidence. Since all the methods are tested on the same model, the explanations across all three methods will be similar if they are indeed accurate approximations. Explanations from the same class should also be similar. Because if the prediction does not change over small input variations, yet the explanations change drastically, it is very likely the explanation method is inaccurate.

An initial inspection of Figure 11a and 11c shows that LIME and kernel SHAP are visually similar simply because the same image segmentation method is used. A closer look reveals that 3 out of the 9 examples do not agree with each other: two from the class df and one from class vasc. When both methods are run repeatedly on these images, the results are very inconsistent, indicating low weights over the entire map. This implies that many features (i.e. superpixels) are equally important for the prediction. The images in class df support this idea as the lesions have light pigmentation and span a large area in the images. Partial feature overlap between methods during repeated runs also back up this intuition. However, this is not the case for the image in class vasc. Multiple kernel SHAP results all included a singular patch on the oval shaped pigmented lesion (similar to the other instances in this class), despite small changes in the surrounding skin area. However, among multiple LIME results, the same superpixel is only activated half of the time (Figure 12). The inconsistency of LIME explanations might be the reason for this behaviour. The fact that a ring-shaped superpixel (not present in other LIME explanations of the same class) is just as likely to be activated shows that the inconsistency is hampering the ability to approximate the black box model accurately.

Grad-CAM is a gradient-based method commonly used for localising points of interest in a CNN when making a prediction. It is shown to be successful

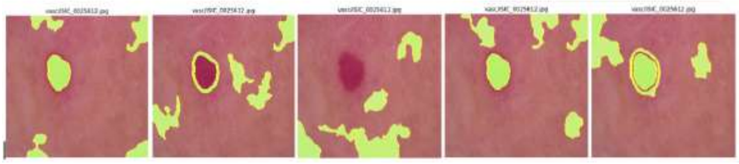
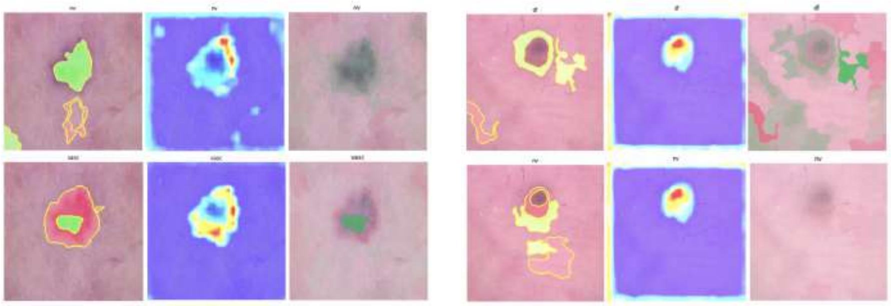


Fig. 12: LIME saliency maps generated 5 more times for image ISIC_0025612.jpg.

in 8 out of 9 images in the examples, highlighting correctly the lesion areas. However, Grad-CAM failed to generate a saliency map for one of the presented images in the figure. The gradient from the last dense layer into the final convolutional layer has become zero because the softmax activation function has become completely saturated. Another cause for concern is that the Grad-CAM saliency maps seem to activate in areas where both LIME and kernel SHAP has deemed to be negatively contributing to the prediction.



(a) Saliency maps generated for predicted class nv and correct class vasc.

(b) Saliency maps generated for predicted class df and correct class nv.

Fig. 13: Examples of saliency maps for misclassified images in order to test the sensitivity of the explanantion methods to change in class labels.

4.3 Sensitivity

Explanations for different instances should be different under the same model, using the same explanation method. If a method offers the same explanation when the user changes either the input image or its predictive class, then the explanation can be considered useless for decision support. For example, image in Fig. 13(a) was misclassified as class nv with a confidence of 0.75, followed by the correct class vasc with confidence 0.24. Its saliency maps (Figure 13a) shows that all three methods generated different looking maps when asked to explain the top prediction and the correct one. LIME changed the location and the color of the superpixels while the other two generated saliency maps that showed a higher activation for the correct class. Again, LIME and kernel SHAP agrees in terms of the superpixels supporting and rejecting the classification. Grad-CAM also shifted its most activated region from the upper right towards the bottom, overlapping the regions covered by the other two methods.

However, if the image is misclassified with a high confidence, like in Frg. 13(b), where the correct class nv is ranked 4th on prediction confidence (0.002) while the top prediction, class df, has a confidence of 0.95, the saliency maps are less helpful. As shown in figure 13b, the Grad-CAM maps are extremely similar and for kernel SHAP the weights on the superpixels are too close to zero for the correct class to provide any useful insight. Only LIME is able to give a distinctly different explanation for class nv.

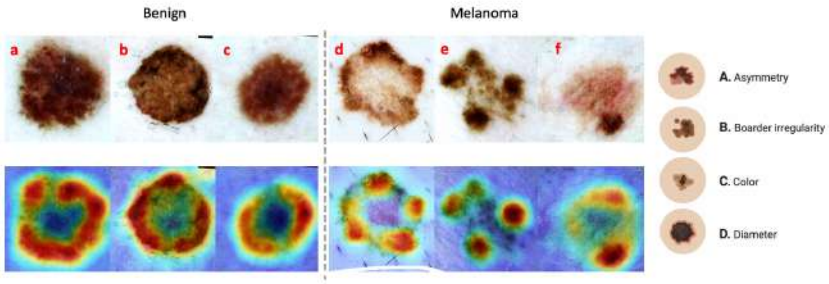


Fig. 14: ABCD features used in diagnosis of skin lesions in dermatology.

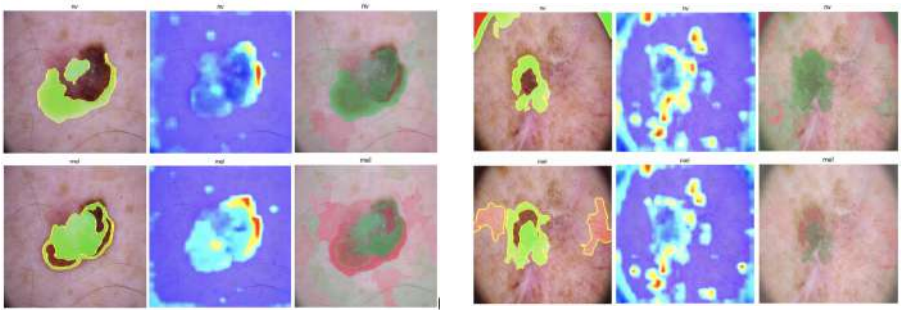
4.4 Clinical relevance

Dermatologists consider certain clinical features during the classification of malignant or benign skin lesions. A popular example is the **ABCDE** features set [6] presented in Fig. 14. In this approach, **A**symmetry, **B**order irregularity, **C**olor variation, **D**iameter and **E**volving or changing of a lesion region are taken into consideration for determining its malignancy. We expect some of these criteria to manifest themselves in saliency map explanations. In HAM10000, class mel and bcc are cancerous and akiec can develop into a cancerous lesion, the rest are all benign.

Recall the confusion matrix (Figure 4), 42% of class mel and 37% of class bcc is misclassified as benign lesions respectively. As the ABCD rule is most commonly used for seperating benign and cancerous lesions, especially melanoma, the instances where class mel is misclassified as class nv are examined to identify how each methods can or cannot reveal potential reasons for failure in the model.

For LIME and kernel SHAP, the image segmentation algorithm dictates the appearance of the explanations. The SLIC segmentation has a parameter that can be chosen to prioritise colour proximity over spatial proximity. Hence the explanation will automatically be made up of uniform coloured patches. As a consequence, the borders of lesions and their overall shape will also be marked out, making it easier to compare to the ABCD rule.

Interestingly, LIME and Kernel SHAP are no longer producing similar results here. There are direct contradictions in Figure 15a, most prominently featured in the explanations for class mel. The observations so far have suggested that LIME and kernel SHAP offer similar level of fidelity with LIME being more inconsistent.



(a) (b)

Fig. 15: Saliency maps for images from class mel misclassified as class nv, with prediction confidence above 90%.

However, by comparing the kernel value in terms of number of features in a sampled instance for the two methods (figure 16), it becomes obvious that kernel SHAP is not the best for local accuracy in this application. The Shapley kernel assigns high values both when there are very few and when there are many features in a sample. The logic behind it being that when a single feature is studied in isolation its contribution to the model can be best measured. To put in context, samples containing only individual superpixels will be regarded just as important as samples containing all but one superpixel. Given that the size of the superpixels are no larger than the whole lesion, the target class used to train the linear approximation is most likely inaccurate because the original model is only fed a partial lesion region. But under kernel SHAP these instances are given the most weight. For figure 15a, the red region marked out by kernel SHAP was fed through the original CNN model and the classifier outcome decided against class mel. Both LIME and kernel SHAP do not account for feature dependence, which makes it impossible for them to capture colour inconsistency and asymmetry when features are examined separately. Therefore, it is no surprise that explanations generated using LIME with activated superpixels covering the entirety or the majority of the lesion region correlate to the correct class mel.

Clearly the CNN has learned the correct representation of class mel and class nv, if the explanation method is telling the practitioner that the model produces the right outcome when the whole area of the lesion is accounted for. However, the model eventually made its decision based on partial information. It signals that a larger receptive field is needed and hence a deeper network can be more effective. On the other hand, image augmentation by zooming and cropping can also have an impact.

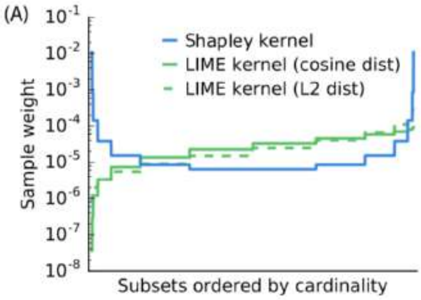


Fig. 16: Comparison of kernel value against the size of the feature set in a sampled instance [8].

Grad-CAM is proven to be hard to interpret. A sweeping generalisation can be made that its saliency maps either fall in to the category of a spotty activation pattern over the lesion region, or a more even activation pattern with continuous edges. However, the extent of clinical relevance halts here as the last max pooling layer is a fairly coarse representation of the original image. This will only be magnified when deeper networks are used. But if layers closer to the input is chosen to generate the saliency map, then Grad-CAM will be reduced to a glorified edge detector. There is also no correlation between activation level and different class labels. The lack of sensitivity (figure 15b) to different class labels also hampers its ability to provide useful information.

5 Conclusion

In this work, we compare and evaluate several explanation-visualization modules against a set of criterion to serve as guideline for machine learning practitioners who wish to add interpretability to deep learning tasks. Experiments were performed using a baseline CNN trained on the benchmark HAM10000 dataset for automated skin lesion classification task. For the first time, detailed experimental results are presented to formalize several criteria like fidelity, consistency, sensitivity and clinical relevance. The authors are of the opinion that the results obtained will be of immediate relevance to readers who value the critical role of explainability in deep learning, particularly in areas of high social consequence like healthcare.

References

1. Skin cancers, Oct 2017. Available at https://www.who.int/uv/faq/skincancer/en/index1.html, accessed on 12.11.2019.
2. Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. page 15, 2010.

3. Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.
4. Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.
5. Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.
6. Emma Harrington, Barbara Clyne, Nieneke Wesseling, Harkiran Sandhu, Laura Armstrong, Holly Bennett, and Tom Fahey. Diagnosing malignant melanoma in ambulatory care: a systematic review of clinical prediction rules. *BMJ Open*, 7(3), 2017.
7. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
8. Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.
9. Pieter Van Molle, Miguel De Strooper, Tim Verbelen, Bert Vankeirsbilck, Pieter Simoens, and Bart Dhoedt. Visualizing convolutional neural networks to improve decision support for skin lesion classification. *Understanding and Interpreting Machine Learning in Medical Image Computing Applications Lecture Notes in Computer Science*, page 115–123, 2018.
10. R. Muthukrishnan and R. Rohini. Lasso: A feature selection technique in predictive modeling for machine learning. In *IEEE International Conference on Advances in Computer Applications (ICACA)*, 2016.
11. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
12. Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, Oct 2019.
13. Lloyd S. Shapley. A value for n-person games. *The Shapley Value*, page 31–40, 1988.
14. Iam Palatnik De Sousa, Marley Maria Bernardes Rebuzzi Vellasco, and Eduardo Costa Da Silva. Local interpretable model-agnostic explanations for classification of lymph node metastases. *Sensors*, 19(13):2969, May 2019.
15. Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Towards medical xai. *ArXiv*, abs/1907.07374, 2019.
16. Philipp Tschandl. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, 2018.
17. Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. volume 5305, pages 705–718, 10 2008.
18. Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *SSRN Electronic Journal*, 2017.
19. Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (in)fidelity and sensitivity for explanations, 2019.
20. Kyle Young, Gareth Booth, Becks Simpson, Reuben Dutton, and Sally Shrapnel. Deep neural network or dermatologist? *Lecture Notes in Computer Science*, page 48–55, 2019.