Machine Learning
https://doi.org/10.1007/s10994-022-06204-w

LIMEcraft: handcrafted superpixel selection and inspection for Visual eXplanations

Weronika Hryniewska[1] · Adrianna Grudzień[1] · Przemysław Biecek[1,2]

Received: 18 November 2021 / Revised: 15 April 2022 / Accepted: 26 May 2022

Abstract

The increased interest in deep learning applications, and their hard-to-detect biases result in the need to validate and explain complex models. However, current explanation methods are limited as far as both the explanation of the reasoning process and prediction results are concerned. They usually only show the location in the image that was important for model prediction. The lack of possibility to interact with explanations makes it difficult to verify and understand exactly how the model works. This creates a significant risk when using the model. The risk is compounded by the fact that explanations do not take into account the semantic meaning of the explained objects. To escape from the trap of static and meaningless explanations, we propose a tool and a process called LIMEcraft. LIMEcraft enhances the process of explanation by allowing a user to interactively select semantically consistent areas and thoroughly examine the prediction for the image instance in case of many image features. Experiments on several models show that our tool improves model safety by inspecting model fairness for image pieces that may indicate model bias. The code is available at: http://github.com/MI2DataLab/LIMEcraft.

Keywords Explainable AI · Superpixels · LIME · Image features · Interactive user interface

Editors: Dana Drachsler Cohen, Javier Garcia, Mohammad Ghavamzadeh, Marek Petrik, Philip S. Thomas.

✉ Weronika Hryniewska
w.hryniewska@mini.pw.edu.pl

Adrianna Grudzień
a.grudzien@student.mini.pw.edu.pl

Przemysław Biecek
przemyslaw.biecek@pw.edu.pl

[1] Faculty of Mathematics and Information Science, Warsaw University of Technology, Koszykowa 75, 00-662 Warsaw, Poland

[2] Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland

Published online: 14 July 2022

1 Introduction

Artificial Intelligence (AI) is rapidly becoming applicable in a variety of domains. Deep learning (DL) has already achieved significant results in many areas concerning computer vision. However, despite these remarkable results achieved by DL, the decisions made by black-boxes still remain unclear for humans, due to difficulty in understanding the reasoning process of the neural network.The lack of interpretability results in critical issues considering model fairness and safety.

For this reason, explainability methods have begun to attract researchers' attention. They have started to create various approaches to explain neural networks' decision process. One vastly used method is Local Interpretable Model-Agnostic Explanations (LIME). It appears that explanation which marks important regions in the image is easily understandable for humans, and therefore used in many scientific studies.

However, so far, most of these methods consider it sufficient to mark only the area that affects the model prediction. The construction of explanations does not take into account the individual factors that contribute to the significance of a region in the model prediction.

In this paper, we propose a new process of explanation based on LIME with the possibility of inspection image features, such as: color, shape, position, and rotation for creation of Visual eXplanations. LIMEcraft also allows handcrafted superpixel selection, which eliminates non-interaction problems with explanation methods and improves the explanation quality of complex image instances. The human interaction process is described in Fig 1.

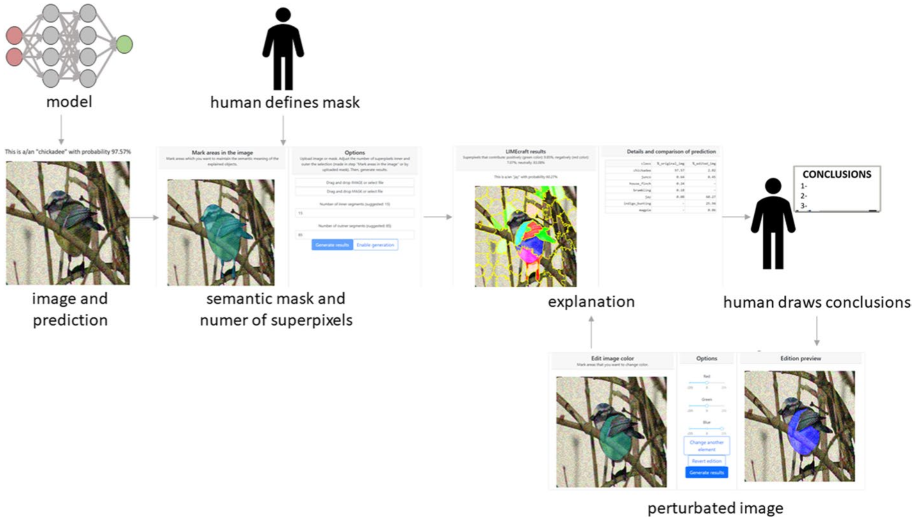


Fig. 1 Diagram summarizing the explanation process using LIMEcraft

2 Related works

2.1 Perturbation-based explainable algorithms

Perturbation-based explainable algorithms use a technique of iteratively removing or changing parts of image features. The variety of perturbations shows how many types of distortions can be applied to images. Among such techniques, we can distinguish: occlusion (e.g. LIME by Ribeiro et al., 2016 or RISE by Petsiuk et al., 2018), DeConvolution and Occlusion Sensitivity (see Zeiler and Fergus, 2014), blurring (Fong & Vedaldi, 2017), conditional sampling (Prediction Difference Analysis by Zintgraf et al., 2017), adding noise (Noise Sensitivity by Greydanus et al., 2018), substitution of existing features (IRT and OSFT by Burns et al., 2019) or superimposing another image (Ribeiro et al., 2018). Based on a model response, the importance of those image features is calculated. Then, the attribution of each feature is computed and the results are shown. However, such techniques do not measure the importance of particular image features, only the location of parts of the image is important. The lack of investigation which image features (color, position, shape, brightness) play the most crucial role, makes these methods prone to errors.

One of the most popular techniques that use occlusion to check the importance of regions in an image is called Local Interpretable Model-Agnostic Explanations Ribeiro et al. (2016 [LIME]). LIME is a model-agnostic explanation algorithm. Model-agnostic means that the architecture of the model does not have an influence on the possibility to explain the model. The explanation is local—it focuses on one specific prediction rather than considering the model globally. The LIME algorithm works with tabular data, text, and images. As for images, it divides them into superpixels based on the quick shift algorithm (Vedaldi & Soatto, 2008). Quick shift is a fast mode-seeking algorithm that segments an image by localizing clusters of pixels in both spatial and color dimensions. Then, a dataset with some superpixels occluded is generated. Each perturbed instance gets the probability of belonging to a class. On this locally weighted dataset, the linear model is trained. The highest positive and negative weights for a specific class are presented in the original image by addition, respectively, a green or a red semitransparent mask on the most important superpixels.

2.2 Methods based on LIME

Following the success of LIME algorithm, many scientists started to be interested in developing this method to make it even more efficient and effective. The greatest number of LIME modifications are for tabular data, e.g., DLIME (Zafar & Khan, 2019), GraphLIME (Huang et al., 2020), Tree-LIME (Li et al., 2019), ALIME (Shankaranarayana & Runje, 2019), LIME-SUP (Hu et al., 2018). However, methods based on LIME for the images are also developed and published in scientific journals, namely: Anchor LIME (Ribeiro et al., 2018), LIMEAleph (Rabold et al., 2020), KL-LIME (Peltola, 2018), MPS-LIME (Shi et al., 2020), and NormLIME (Ahern et al., 2019).

Anchor LIME (Ribeiro et al., 2018), instead of hiding some superpixels from the original image, superimposes another image over the rest of the superpixels. Authors stress that the method might seem unnatural, but it allows to predict the model's behavior on unseen cases.

A LIME-based approach Kullback Leibler divergence, called KL-LIME (Peltola, 2018), is designed for explaining the predictions of Bayesian predictive models. The proposed method combines methods from Bayesian projection predictive variable selection with LIME algorithm. In KL-LIME, parameters of the interpretable model are found by minimizing the Kullback-Leibler divergence from the predictive model.

LIME-Aleph (Rabold et al., 2020) combines an explanation generated by LIME with logic rules obtained by the Inductive Logic Programming system Aleph. The authors claim that in LIME it is not clear if the classification decision is made due to the presence of specific parts of the image or because of the specific relation between them. Their approach is capable of identifying the relationship between elements as an important explanatory factor.

The method of superpixels selection is replaced in MPS-LIME (Shi et al., 2020) with Modified Perturbed Sampling (MPS) operation. MPS-LIME converts superpixels into an undirected graph. The authors claim that their method does not ignore the complicated correlation between image features and improves the algorithm efficiency.

LIME is a method for a local explanation, while NormLIME (Ahern et al., 2019) tries to aggregate local explanations and create a global, class-specific explanation.

2.3 Limitations of existing algorithms

There are some weaknesses of the LIME method. The definition of superpixels does not take into account the semantic meaning of objects in the image, and consequently, sometimes different objects are located within a single superpixel. This is especially visible in images with many overlapping objects and in medical images.

Moreover, despite many attempts to improve the LIME algorithm, it is often considered non-robust. Alvarez-Melis and Jaakkola (2018) show that perturbation-based methods are especially prone to instability. Small changes in the input image, such as adding Gaussian noise, should not significantly affect explanations. However, due to the fully automatic selection of the superpixels, LIME depends strongly on nonsemantic input image features and is particularly sensitive to noise.

A suggestion that the existing explanation techniques are vulnerable to attacks of adversarial classifiers is made by Slack et al. (2020). They claim that LIME is not sufficient for ascertaining the discriminatory behavior of classifiers in sensitive applications and is not reliable. Their approach can be used to scaffold a biased classifier. Predictions of the classifier on the input data still remain biased, but the post hoc explanations of the scaffolded classifier look innocuous.

Moreover, Rahnama and Boström (2019) claim that LIME suffers from data and label shift. Their experiments show that the instances generated by LIME's algorithm are distinctly different from training instances drawn from the underlying distribution. Based on the obtained results, they conclude that random perturbations of the features of the explained instance cannot be considered a reliable method of generating data in the LIME method.

Schallner et al. (2020) stress that the selection of a suitable superpixel algorithm should be considered. They conduct several experiments comparing different superpixel algorithms. Finally, they say that for each problem, the superpixel selection algorithm should be consulted with domain experts. In some situations, it is important to generate superpixels of significantly different sizes depending on the semantic meaning of the content.

To sum up, LIME is sensitive to small changes in the input image, such as adding noise or an adversarial attack. Random feature perturbations are unreliable because LIME suffers from data and label shift. The choice of an optimal segmentation algorithm affects the LIME results and should be consulted with a domain expert. As we will show below, for partially covered or noisy features, automatic segmentation leads to worse results than expert-assisted segmentation.

3 LIMEcraft

3.1 Motivation

During our work with medical images, we discovered that LIME prefers sharp boundaries between superpixels. It also needed further clarification on which kind of image features the prediction of a specific part of the image was made. Our motivation was to create a solution that not only works better on LIME's corner cases, but also gives more detailed insight into the model's strengths and weaknesses.

The weakness of LIME is that the division into superpixels is unacceptable when objects are partially covered or the image is noisy. LIME also fails when there are many elements that interfere with the way superpixels are constructed, e.g. zebra stripes. The lack of semantic understanding of the image makes LIME unsafe for many medical images and for images taken in the natural environment when some objects are partially obscured by others, as presented in Fig. 2. For this reason, we create an opportunity to manually select superpixels.

The next problem that we wanted to address is the lack of understanding of the image features that mostly contribute to the model prediction. The location of the most important superpixel does not provide us with complete information about whether the model has learned the correct features. Without careful verification, we cannot be sure that a car of any color will be correctly recognized by the model as a car and classified into the correct class. Image features' inspection may also help us to investigate the possibility of bias based on, e.g., skin color.

The LIME algorithm is prone to the presence of noise. Noise can greatly interfere with how superpixels are formed, and thus, due to the large differences in superpixel sizes, also change the areas that are marked by LIME as relevant to the model. Although our solution

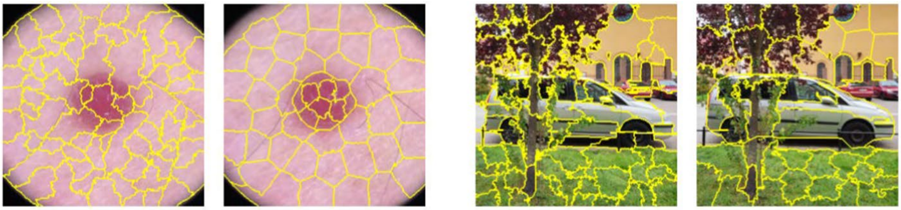


Fig. 2 Examples of division into superpixels by LIME and by our algorithm of skin lesion (Vargas et al., 2020) and car photo. For each pair, the left image shows the automatic LIME segmentation and the right one a LIMEcraft segmentation supported by the user who outlined the skin lesion of interest (first pair) or the car (second pair). LIME uses quick shift algorithm, and LIMEcraft uses manual or predefined superpixel selection, and then, segmentation based on K-means clustering algorithm

cannot fully eliminate this instability, it limits an uncontrolled splitting into superpixels and improves the reliability testing capabilities of the model.

In summary, LIME's weaknesses are: (1) uninformative/misleading segmentation, due to the lack of semantic understanding of the image during the creation of superpixels, (2) lack of understanding of the image features that mostly contributed to the model prediction, (3) and sensitivity to the presence of noise.

3.2 Different types of superpixels' selection

In LIMEcraft there are two main ways of selecting segments called "superpixels": semantic and non-semantic. The first one can be done by a user using a tool, which allows drawing an irregular path of shape, or by uploading a prepared mask of superpixels. Such functionalities lead to greater influence on proper image analysis.

After manual or predefined superpixel selection, the next step is the non-semantic selection of superpixels. In this step, only previously selected areas are divided into smaller pieces. Such automatic segments are generated using image segmentation based on the K-means clustering algorithm. Moreover, we can determine into how many segments the areas selected will be divided. LIMEcraft suggests how many superpixels will be optimal for each case. It calculates how many superpixels should be inside and outside the selected areas to maintain the same size of the superpixels. However, in some cases, the user may want to increase the number of superpixels inside the selected areas to obtain more detailed results. It might be helpful when the object inside the selection has small details and the rest of the image is just a little diverse background.

The potential use-case scenario is to manually define objects that should not be combined in the same superpixel. Such an image could be, for example, a complex cityscape partially obscured by tree branches, an X-ray image in which there are naturally small differences in the brightness of areas while these areas belong to other internal organs, a photograph of undergrowth in which there are many objects similar in color.

Checking the responsibility of a network to classify lung lesions can be an example of using the mask loading functionality. The neural network was trained on the data with lesions label, and then validated on an external database, as recommended by Hryniewska et al. (2021). For external validation, the dataset for lesions detection was chosen, so the database contained not only the names of lesions present on the images but also their location. The masks of lesions can be easily uploaded into LIMEcraft to verify whether they are important for the model's prediction. The dashboard shows how much of the whole image has a positive impact on the prediction of the model (green color), negative (red color), and neutral (without color). It might be useful to assess the importance of the selected areas (when the superpixels are not exactly the same size).

3.3 Inspection of image feature importance

The interface we have created makes it possible to analyze the impact of image perturbation on the prediction of the model. The user can edit the color, shape, and position of the selected area, and then subject the edited image to the LIMEcraft algorithm.

Color edition enables to manipulate the values of individual channels of the image (RGB), so the brightness can also be adjusted. Moreover, we can rotate the selected area and change its position. The moved piece can also be completely removed. In the edited area, the inpainting algorithm based on"biharmonic equation" (Damelin &

Hoang, 2018) is applied. The selected area may also be expanded according to a user-defined“power”value. For values greater than 1, it will be enlarged, and for values less than 1, it will be shrunk.

In order to be able to better understand the changes that have occurred as a result of the perturbation of the image, the report in a form of a table is generated. It compares the percentages of probabilities for the predicted classes.

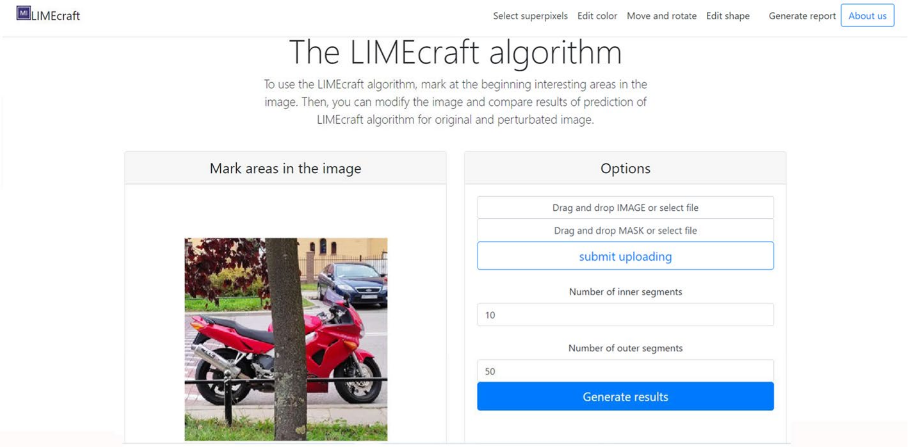
It is crucial to see how the model responds to a change in individual image elements. For example, for microscope images, the shapes and colors of cells may be relevant to a classification task. However, position and rotation should not significantly affect the prediction of the trained model. By running such experiments using LIMEcraft, we can observe whether the model has learned to recognize objects by the correct features.

3.4 Interactive user interface

In contrast to the fully automated approach of the LIME algorithm, our LIMEcraft algorithm incorporates the human into the process of explainability. It gives them the ability to influence the division into superpixels, the choice of the number of superpixels, and a more detailed analysis of the model by comparing the prediction results for the original image and the one subjected to perturbations.

The undeniable advantage of the dashboard is that it can be used by people unfamiliar with programming, because the interface is very intuitive and user-friendly. The interface is presented in Fig. 3.

The interactive User Interface gives the human more control over the quality of the model and, as a result, the safety of the created models. An important aspect that cannot be ignored is the variety of biases that a model may have. With an interactive interface, different possibilities (image modifications) can be tested to ensure safety and fairness. A good example to consider here would be the possibility of changing the skin color of a person in a photo.



The LIMEcraft algorithm

To use the LIMEcraft algorithm, mark at the beginning interesting areas in the image. Then, you can modify the image and compare results of prediction of LIMEcraft algorithm for original and perturbated image.

Mark areas in the image
Options
Drag and drop IMAGE or select file
Drag and drop MASK or select file
submit uploading

Number of inner segments: 10

Number of outer segments: 50

Fig. 3 An example of the user interface for code available on http://github.com/MI2DataLab/LIMEcraft allowing to outline features of interest and analyze the explanations yourself

3.5 Algorithm in details

LIMEcraft algorithm is based on LIME. However, there are some key differences between both of them. As presented in Algorithm 1, LIMEcraft algorithm includes the possibility to define mask for the selected image, and then, to choose the number of superpixels inside and outside the mask. The next important innovation is a functionality to edit image, which provides the insight into the model's robustness. Moreover, LIME and LIMEcraft have different segmentation algorithms. LIME uses quick shift algorithm. In LIMEcraft, besides manual or predefined superpixel selection, segmentation is based on K-means clustering algorithm.

Algorithm 1 LIMEcraft algorithm

Input: black-box model f, input sample $image$, mask of superpixels M, number of superpixels n_1 (inner of selected area), n_2 (outer of selected area), number of features to pick m

Output: explainable coefficients from the linear model

```
if you want then                ▷ $Define\ mask$
    $mask \leftarrow UploadMask(M)$
else
    $mask \leftarrow MarkAreasInTheImage$
end if
if you want then                ▷ $Choose\ number\ of\ superpixels$
    $x_1 \leftarrow segment.KmeansClustering(mask, n_1)$
    $x_2 \leftarrow segment.KmeansClustering(image - mask, n_2)$
    $x \leftarrow x_1 + x_2$
end if
if you want then                ▷ $Change\ image\ features$
    Edit image
end if
$\overline{y} \leftarrow f.predict(x)$
for $i$ in $n$ do
    $p_i \leftarrow Permute(x)$    ▷ $Randomly\ pick\ superpixels$
    $obs_i \leftarrow f.predict(p)$
    $dist_i \leftarrow \mid \overline{y} - obs_i \mid$
end for
$simscore \leftarrow SimilarityScore(dist)$
$x_{pick} \leftarrow Pick(p, simscore, m)$
$L \leftarrow LinearModel.fit(p, m, simscore)$
return $L.weights$
```

4 Method evaluation

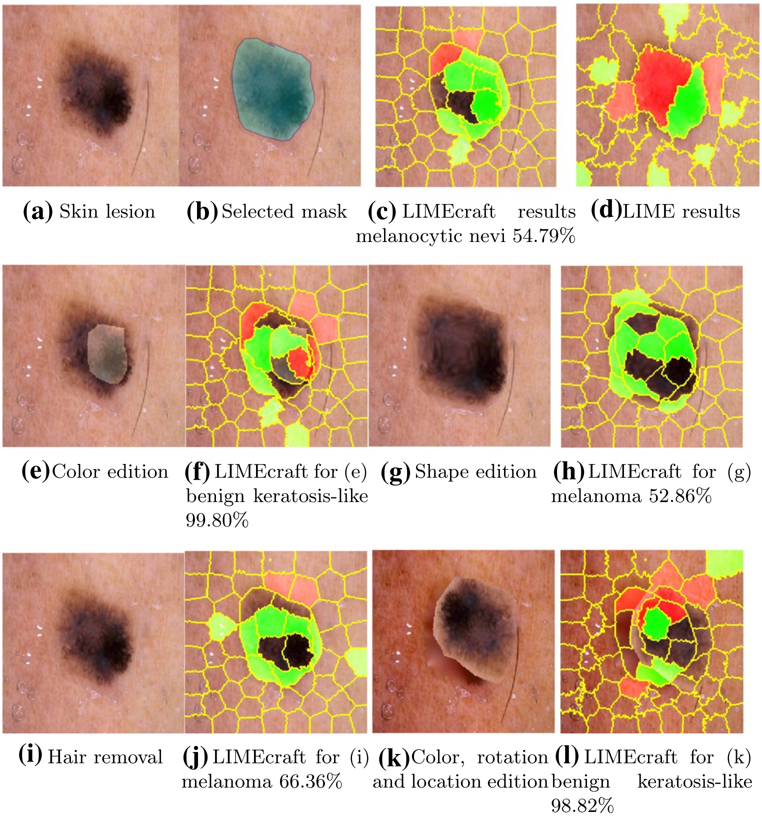
4.1 Relevance of image features to correctness of prediction

We test the ability to evaluate a model for skin lesion classification using LIMEcraft. First of all, we select a database for classification of skin lesions (Mader, 2019). Then, we choose model architecture: MobileNet and image input size: 224 × 224 × 3. The neural network uses the base pretrained on Imagenet. While classifying into 7 classes, it achieves 64.6% of sparse categorical accuracy.

To investigate if the model is predicting class label based on skin lesions, not artifacts, such as hair, we conduct several tests using LIMEcraft. We select a mask (presented in Fig. 4b), and we run several experiments of feature importance.

The color edition, visible in Fig. 4e, changes the model prediction from class melanocytic nevis (54.79%) to benign keratosis-like (99.80%).

LIMEcraft results obtained after shape edition (power of expansion: 1.4), in Fig. 4g, do not change model's confidence so drastically, because it drops only 8.5%. However, it changes the most probable class to melanoma with 52.86% of probability.



(a) Skin lesion (b) Selected mask (c) LIMEcraft results melanocytic nevi 54.79% (d) LIME results

(e) Color edition (f) LIMEcraft for (e) benign keratosis-like 99.80% (g) Shape edition (h) LIMEcraft for (g) melanoma 52.86%

(i) Hair removal (j) LIMEcraft for (i) melanoma 66.36% (k) Color, rotation and location edition (l) LIMEcraft for (k) benign keratosis-like 98.82%

Fig. 4 Model inference results for skin lesion classification using the LIMEcraft algorithm. The input image shows a lesion called melanocytic nevi. Superpixels colored to green mean that this part of the image contributes positively to the prediction, while red parts show negative impact (Color figure online)

In Fig. 4i, we remove hair and the prediction of class melanocytic nevi decreases to 33.54%. The most probable class for this case is melanoma.

The last conducted experiment covers many different image editions, namely: 10px shift into right and down direction, 180° rotation, and change of patient's skin color.

Based on (Stieler et al., 2021)'s work, it can be presumed that changes such as rotation and shift of the skin lesion should not change the prediction of the model. On the other hand, changing the color and boundaries may change the features of the lesion.

Figure 5 partially confirms our assumptions. The color of the lesion had a large effect on the prediction of the model. Hair removal does not drastically change the model's prediction. Nevertheless, there are some disturbing findings. A combination of several changes: skin color, shape, and rotation had a great influence on model classification result. This result may become the basis for speculating that the model could be biased.

Nevertheless, it is very important to note that results obtained by using the LIMEcraft algorithm should be evaluated by a domain specialist. It is up to the specialist to determine whether a given change in the input image should cause a change in the model response.

4.2 Noise example-based sensitivity analysis

Slack et al. (2020) state that LIME can be easily manipulated to hide biases. It stresses the fact that such explanations are untrustworthy and not safe in usage.

We want to examine if LIMEcraft improves the robustness of model explanations. Alvarez-Melis and Jaakkola (2018) add Gaussian noise to the input image and check how strongly the output generated by explanation varies. They show that the LIME explanation is highly vulnerable to small changes in input image due to the presence of sparse superpixels.

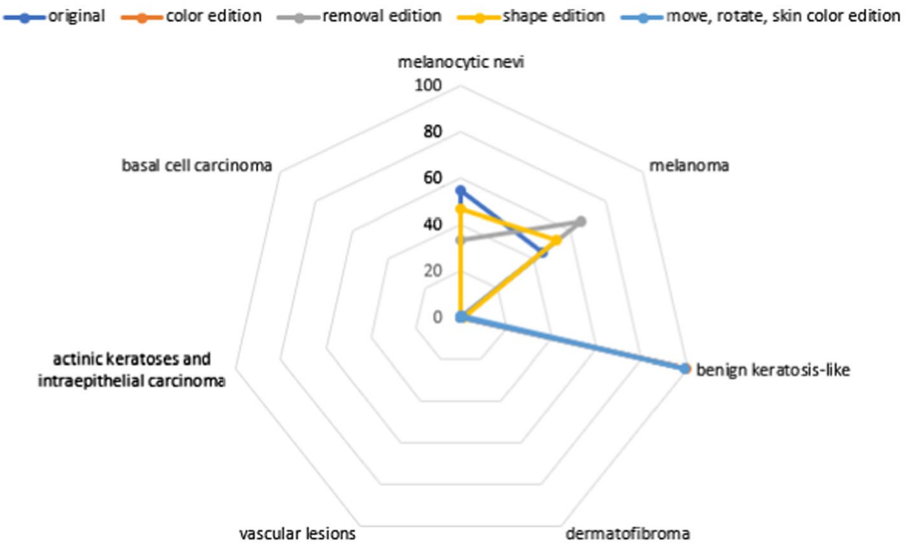


Fig. 5 Radar plot of the model's confidence in class selection for a skin lesion called melanocytic nevi

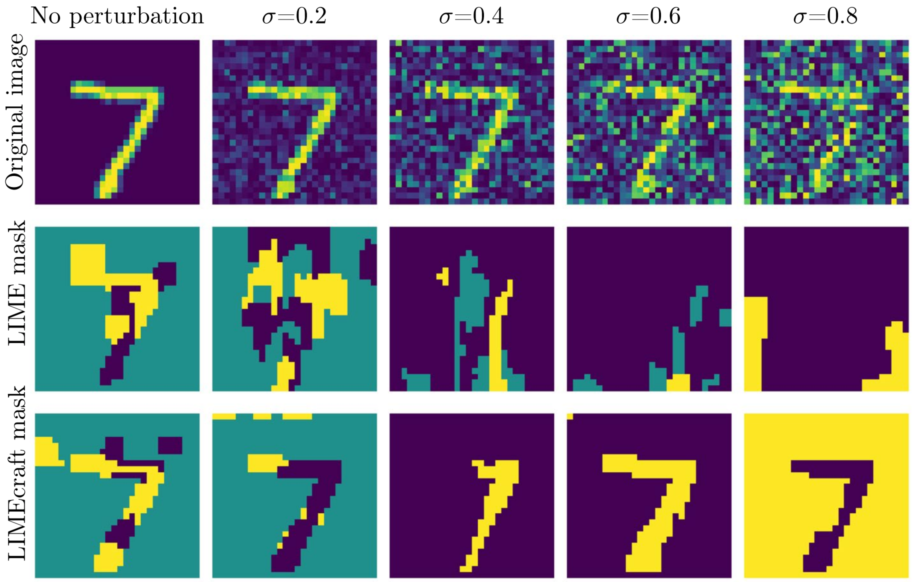


Fig. 6 Images of digit 7 perturbed with Gaussian noise of σ strength and images showing the significance of superpixels on the classification of digit 7 (-1 negatively affects prediction, 0 has no significant effect, 1 positively affects) generated by the LIME and LIMEcraft algorithms

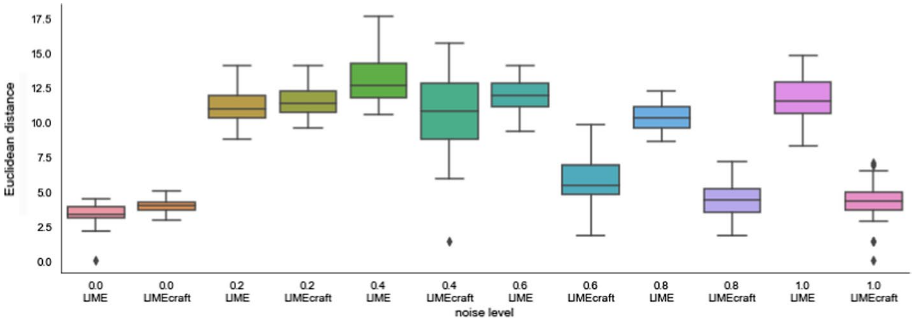


Fig. 7 Box plot of the noise level dependence of the 10 images from the MNIST database on the similarity of the masks generated by LIMEcraft and LIME computed by Euclidean distance for each pair of images

To test LIMEcraft safety, we train on MNIST dataset small neural network with only one convolutional layer with relu activation, max-pooling, flattening, and one dense layer. It receives 98.07% accuracy on the test set.

Then, we perturb the image with number 7 by adding Gaussian noise that varies in intensity. The input image has pixel intensities between [0, 1] and the values out of this range are clipped. In Fig. 6 saved masks of superpixels generated by both LIME and LIMEcraft are presented. In order to obtain more comparable evaluation results for both algorithms, the same segmentation algorithm is chosen - based on K-means clustering.

In the experiment presented in Fig. 7, it appears that, thanks to the manual selection of superpixel covering the number, LIMEcraft is more stable in case of the presence of noise. The Euclidean distance for strongly noisy images is not as high as for LIME algorithm. Not changing the outer border of the 7 number in LIMEcraft leads to an improvement in robustness to perturbations in the input. It is also worth noting that for no noise or low noise values, the masks generated by LIMEcraft are comparable to the results achieved by LIME. Thus, it can be concluded that LIMEcraft improves the repeatability of the obtained explanations.

4.3 Evaluation with human subjects

The user study consisted of a pilot user study, and then a formal user study of 20 people. After the pilot study, the remarks reported by users were used to improve LIMEcraft and any ambiguities in the survey were clarified before proceeding with the formal user study.

The aim of the participants was to follow the instructions to complete a task and fill a questionnaire. Participants tested ImageNet model of Inception v3 architecture with 5 different non-medical images available in Github repository of LIMEcraft: https://github.com/MI2DataLab/LIMEcraft. They had to assess a model quality by seeing what a model has learned and to check the model's safety using the LIME and LIMEcraft explanations. Using a questionnaire, we wanted to assess whether people think LIMEcraft enhances their explanatory abilities and what features of LIMEcraf are most important to them.

In the formal user study, 15 men and 5 women of varying age participated. No strong correlation was discovered between sex and responses. Users' experience in using AI models and LIME differed. The AI experience declared by users had a Gaussian-like distribution. 40% of the people did not know LIME, and the remaining number of users was roughly equally distributed between those who knew the method poorly, moderately, and well. The median time required to participate in the experiment and respond was 38 minutes and the interquartile range was 34 minutes.

In Fig. 8, in all questions, LIMEcraft was rated better than LIME. The bigger differences are visible in questions about how well the superpixel boundaries were chosen and how would the ability to change the number of superpixels be rated.

The most appreciated functionalities provided by the LIMEcraft tool were abilities to select own superpixels and interact with the explanation method. Suprisingly, the users disagreed on whether the image modifications made it easier to detect model weaknesses. Also, opinions were divided on whether users would like to use LIMEcraft in the future.

In examining the correlation between responses, it was noted that the greater the familiarity with LIME, the lower ratings users gave in response to the questions:"How would you rate the automatic superpixel division?", and"How well were the superpixel boundaries chosen?". It might lead to the conclusion that experienced LIME users have already discovered its weaknesses. Users more experienced in AI and LIME did not rate the ability to catch model weaknesses by image modifications so highly.

The user study has shown that LIMEcraft is a self-sufficient, end-to-end tool for model exploration. The functionality related to superpixel selection is especially valuable. The ability to edit the image is less desirable than choosing own superpixels. The majority of people appreciated LIMEcraft under each of the criteria studied.

4.4 LIMEcraft versus other explanatory methods

Various explanation algorithms are used to explain image models (Samek et al., 2019). Apart from explanations that work by perturbing input data, e.g., LIME and

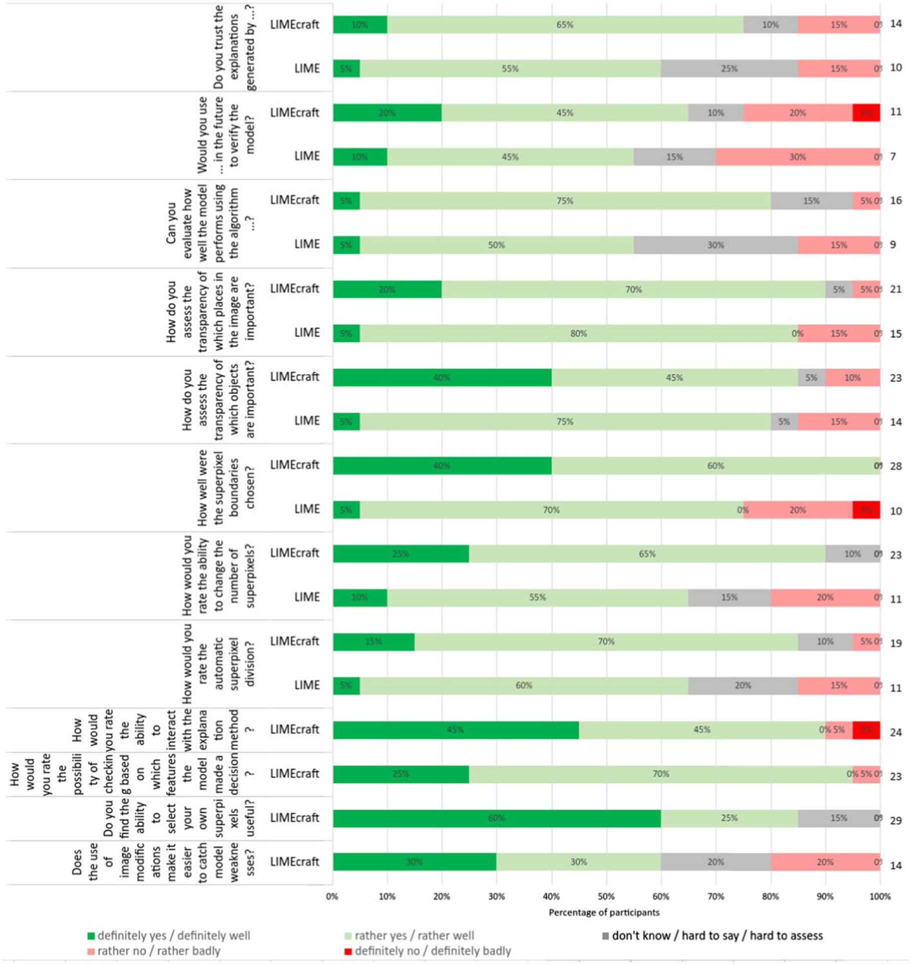


Fig. 8 Stacked column chart showing the responses of 20 users to a survey designed to evaluate and compare LIMEcraft and LIME. Each row is related to a separate question. The colors encode the respondents' answers and are then converted to numbers according to the legend. On the right side of the chart is the sum of the ratings of all respondents (Color figure online)

LIMEcraft, there are other popular methods that are based on gradients, e.g., Grad-CAM and Grad-CAM++.

Grad-CAM and Grad-CAM++ produce heatmaps that are calculated as a result of the features extracted from the final convolutional layer of the model. The heatmaps are coarse, as presented in Fig. 9. While comparing them to perturbation-based approaches, it is important to stress that they do not produce results with sharp boundaries of attention maps. Moreover, gradient-based algorithms show only areas where the attention of a model is present or absent, and do not take into account negative influence in predicting a class.

Since LIMEcraft and LIME are both perturbation-based algorithms, it is worth noting the differences between them. As presented in Table 1, LIMEcraft offers more possibilities for exploring specific regions than LIME and is more robust to noise. However, it cannot be applied directly to text or tabular data. Taking a human-in-the-loop of explanation makes

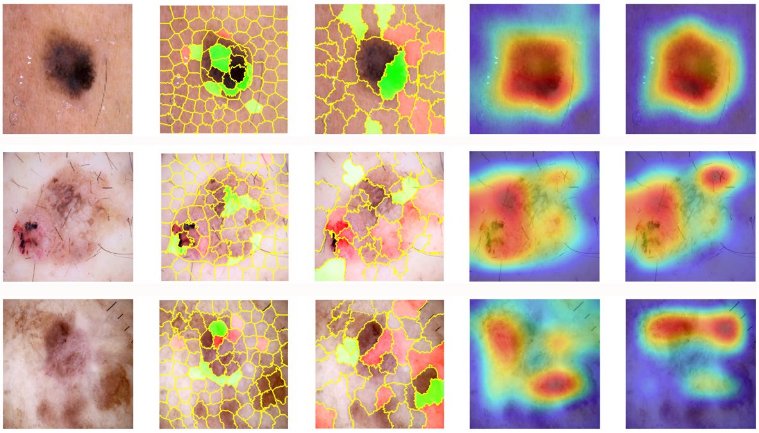


Fig. 9 Model inference results for skin lesion classification using following explanation's algorithms: LIMEcraft, LIME (Ribeiro et al., 2016), Grad-CAM (Selvaraju et al., 2019), Grad-CAM++ (Chattopadhay et al., 2018). In LIMEcraft and LIME, superpixels colored green indicate that this part of the image has a positive impact on prediction, while red parts have a negative one. Grad-CAM and Grad-CAM++ are typically presented in"hot-to-cold"color scale. The red color shows the areas that have the most positive influence on model prediction, while blue presents the most negative impact. The first row shows the lesion with sharp borders. In the second one, only the wounded part was selected in LIMEcraft. The lesion in the last row has blurred borders (Color figure online)

Table 1 Comparison of LIMEcraft and LIME

LIMEcraft	LIME
Human-in-the-loop process	Fully automatic process
Image data, audio data (transformed into a spectrogram)	Image, text, tabular data
Segmentation can be corrected	Segmentation cannot be corrected
Required masks (ready mask can be used)	Not required masks
Opportunity to focus explanations in a specific area.	Lack of ability to focus explanations in a specific area
Opportunity to check what kind of image features contribute the most in explanation	Lack of opportunity to check what kind of image features contribute the most in explanation
More robust to noise	Less robust to noise
Object of interest don't have to have sharp boundaries	Object of interest should have sharp boundaries
Opportunity to check how important are colored superpixels	Lack of opportunity to check how important are colored superpixels

LIMEcraft not work as an automatic tool, but it provides the possibility to add semantic meaning to the explanation process.

As shown in Fig. 9, in the first row, the sharp boundaries are easily identified by LIME. However, LIME divided the lesion into only two parts. The superpixels in the rest of the image are much smaller, which means that the most important part of the image cannot be examined in detail. LIMEcraft offers a solution to overcome this limitation. The ability to manipulate the number of superpixels into which an image is divided in selected region provides the possibility to fine-tune the size of the superpixels. When we want small details, we divide the part of the image into more superpixels; whenever we want a more global result, we divide it into fewer superpixels.

In the second row of Fig. 9, in comparison to LIMEcraft, LIME does not deal well with a lesion of varying colors. In some places, the lesion and the unchanged skin are in the same superpixel. The lack of semantic information in the explained image is also shown in the third row.

Gradient-based methods versus perturbation-based methods are different approaches in explainability. LIMEcraft can be used to combine those two perspectives. The heatmap generated by gradient-based methods can be changed into masks and then used as ready input masks into LIMEcraft. It brings us a closer look at the hidden space of the explained model.

5 Conclusions

To our knowledge, LIMEcraft is the first LIME-based tool that allows the user to be directly involved in the construction of an explanation. In this explanation tool, the user can influence: (1) the superpixel constructions of the interpretable space on which LIME is based, (2) the specific aspects of the image to study how the operation affects the prediction of the model and its explanation, (3) the level of detail of the obtained results. Moreover, operations such as perturbations of colors, shapes, and positions allow the user to study the sensitivity of the model to changes.

The ability to select a mask for LIMEcraft allows domain knowledge to be introduced into the process of explaining. In addition, it makes it possible to apply the method to cases where the object under study is partially obscured or blends in color with its surroundings. By checking the relevance of each image feature, we can verify that the model is not biased and that it has learned the correct features, which will result in increasing the model's safety and trustworthiness.

LIMEcraft might be applied to other inputs, such as audio data, because audio is often transformed into a spectrogram that is an image. Using LIMEcraft, rectangular-shaped superpixels can be created, which would allow the audio to be divided by frequency. Also, the concept of bringing human-in-the-loop can be applied to the explanation process of tabular data, and humans can define the thresholds for specific data transformations, e.g. grouping and discretization of variables.

Incorporating the human into the process of explainability allows to benefit from the knowledge they have. However, if a person does not expect an element to be important, e.g., if they do not notice a lesion, they will not mark it. For this reason, it is also important to be aware of the risk of introducing bias into the model. This is still an open problem that needs further research.

We provided code and a web application that can serve others to evaluate the safety and robustness of their models.

Acknowledgements We would like to thank Dariusz Parzych, psychologist from Center for Innovation and Technology Transfer Management at Warsaw University of Technology, for a valuable help in preparing and conducting user experience analysis, Piotr Piątyszek for helping in deployment of a LIMEcraft web application, and 20 volunteers for participating in the user study.

Author contributions Conceptualization: WH and PB; Methodology: WH; Implementation: AG and WH; Formal analysis and investigation: WH and AG; Writing—original draft preparation: WH and AG; Writing—review and editing: PB and WH; Funding acquisition: PB; Supervision: PB.

Funding The research leading to these results received funding from NCN Sonata Bis-9 Grant 2019/34/E/ST6/00052.

Availability of data and material The data and models used in this work are available at: https://keras.io/api and https://www.kaggle.com/kmader/deep-learning-skin-lesion-classification

Code availability The code is available at: http://github.com/MI2DataLab/LIMEcraft.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Consent to participate All participants of the user study gave their informed consent that they: received and understood the information concerning the test, understood the purpose of the test and their involvement in it, understood that they may withdraw from participation at any stage and understood that their personal results will remain confidential and they will not be quoted or damaged if the information will be made public.

References

Ahern, I., Noack, A., Guzman-Nateras, L., Dou, D., Li, B., & Huan, J. (2019). Normlime: A new feature importance metric for explaining deep neural networks. *CoRR* abs/1909.04200 . Retrieved from http://arxiv.org/abs/1909.04200

Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. In *Proceedings of the 2018 icml workshop on human interpretability in machine learning*. Retrieved from http://arxiv.org/abs/1806.08049

Burns, C., Thomason, J., & Tansey, W. (2019). Interpreting black box models via hypothesis testing (pp. 47–57). Association for Computing Machinery, Inc. Retrieved from https://arxiv.org/abs/1904.00045v3. 10.1145/3412815.3416889

Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. Retrieved from http://dx.doi.org/10.1109/WACV.2018.00097. 10.1109/wacv.2018.00097

Damelin, S. B., & Hoang, N. S. (2018). On surface completion and image inpainting by biharmonic functions: Numerical aspects, vol. 2018. Hindawi Limited 10.1155/2018/3950312

Fong, R. C., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 3449–3457). 10.1109/ICCV.2017.371

Greydanus, S., Koul, A., Dodge, J., & Fern, A. (2018). Visualizing and understanding Atari agents. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning*, vol. 80 (pp. 1792–1801). PMLR. Retrieved from https://proceedings.mlr.press/v80/greydanus18a.html

Hryniewska, W., Bombinski, P., Szatkowski, P., Tomaszewska, P., Przelaskowski, A., & Biecek, P. (2021). Checklist for responsible deep learning modeling of medical images based on COVID-19 detection studies, vol. 118 (p. 108035). Pergamon. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0031320321002223. 10.1016/j.patcog.2021.108035

Hu, L., Chen, J., Nair, V. N., & Sudjianto, A. (2018). Locally interpretable models and effects based on supervised partitioning (LIME-SUP). *CoRR* abs/1806.00663. Retrieved from http://arxiv.org/abs/1806.0066

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., & Chang, Y. (2020). Graphlime: Local interpretable model explanations for graph neural networks. *CoRR* abs/2001.06216. Retrieved from https://arxiv.org/abs/2001.06216

Li, H., Fan, W., Shi, S., & Chou, Q. (2019). A modified lime and its application to explain service supply chain forecasting. In Natural language processing and Chinese computing (pp. 637–644). Springer International Publishing.

Mader, K. (2019). Deep learning skin lesion classification—kaggle. Retrieved from https://www.kaggle.com/kmader/deep-learningskin-lesion-classification

Peltola, T. (2018). Local interpretable model-agnostic explanations of bayesian predictive models via kullback-leibler projections. In *Proceedings of the 2nd workshop on explainable artificial intelligence (XAI 2018) at IJCAI/ECAI 2018*.

Petsiuk, V., Das, A., & Saenko, K. (2018). Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British machine vision conference (BMVC).*

Rabold, J., Deininger, H., Siebers, M., & Schmid, U. (2020). Enriching visual with verbal explanations for relational concepts–combining LIME with aleph, vol. 1167 CCIS (pp. 180–192). 10.1007/978-3-030-43823-4_16

Rahnama, A. H .A., & Boström, H. (2019). A study of data and label shift in the LIME framework. *CoRR* abs/1910.14421. Retrieved from http://arxiv.org/abs/1910.14421

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 2016 conference of the North American chapter of the Association for Computational Linguistics: Demonstrations* (pp. 97–101). Association for Computational Linguistics (ACL). 10.18653/v1/n16-3020

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In S. A. McIlraith & K. Q. Weinberger (Eds.), *Proceedings of the thirty-second AAAI conference on artificial intelligence, the 30th IAAI-18, and the 8th AAAI symposium on EAAI-18* (pp. 1527–1535). AAAI Press.

Samek, W., Montavon, G., Vedaldi, A., Hansen, L., & Müller, K. R. (2019). Explainable AI: Interpreting, explaining and visualizing deep learning. 10.1007/978-3-030-28954-6

Schallner, L., Rabold, J., Scholz, O., & Schmid, U. (2020). Effect of superpixel aggregation on explanations in LIME-A case study with biological data. In *Communications in computer and information science*, vol. 1167 CCIS (pp. 147–158). 10.1007/978-3-030-43823-4_13

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2019). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision, 128*(2), 336–359.

Shankaranarayana, S. M., & Runje, D. (2019). ALIME: Autoencoder based approach for local interpretability. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11871 LNCS (pp. 454–463). Springer. 10.1007/978-3-030-33607-3_49

Shi, S., Zhang, X., & Fan, W. (2020). A modified perturbed sampling method for local interpretable model-agnostic explanation. *CoRR* abs/2002.07434. Retrieved from https://arxiv.org/abs/2002.07434

Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (2020). Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM conference on AI, ethics, and society* (pp. 180–186). New York, NY, USA: Association for Computing Machinery. 10.1145/3375627.3375830

Stieler, F., Rabe, F., & Bauer, B. (2021). Towards domain-specific explainable ai: Model interpretation of a skin image classifier using a human approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops* (pp. 1802–1809).

Vargas, P., Cárdenas, R., Cullen, R., & Figueroa, A. (2020). Eruptive disseminated spitz nevi-case report, vol. 95 (pp. 71–74). Retrieved from www.sciencedirect.com/science/article/pii/S0365059619301540. 10.1016/j.abd.2019.01.010

Vedaldi, A., & Soatto, S. (2008). Quick shift and kernel methods for mode seeking. In D. Forsyth, P. Torr, & A. Zisserman (Eds.), *Computer vision—ECCV 2008* (pp. 705–718). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zafar, M. R., & Khan, N. M. (2019). Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. In *Proceeding of ACM SIGKDD workshop on explainable AI/ML (XAI) for accountability, fairness, and transparency*. Anchorage, Alaska: ACM.

Zeiler, M.D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8689 LNCS (pp. 818–833). Springer, Cham. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53. 10.1007/978-3-319-10590-1_53

Zintgraf, L. M., Cohen, T. S., Adel, T., & Welling, M. (2017). Visualizing deep neural network decisions: Prediction difference analysis. In *5th international conference on learning representations, ICLR 2017—Conference track proceedings. International conference on learning representations, ICLR*. Retrieved from https://arxiv.org/abs/1702.04595v

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.