



# CBR-LIME: A Case-Based Reasoning Approach to Provide Specific Local Interpretable Model-Agnostic Explanations

Juan A. Recio-García<sup>(✉)</sup>, Belén Díaz-Agudo, and Victor Pino-Castilla

Department of Software Engineering and Artificial Intelligence,  
Instituto de Tecnologías del Conocimiento,  
Universidad Complutense de Madrid, Madrid, Spain  
{jareciog,belend,vpino}@ucm.es  
<http://gaia.fdi.ucm.es>

**Abstract.** Research on eXplainable AI has proposed several model agnostic algorithms, being LIME [14] (Local Interpretable Model-Agnostic Explanations) one of the most popular. LIME works by modifying the query input locally, so instead of trying to explain the entire model, the specific input instance is modified, and the impact on the predictions are monitored and used as explanations. Although LIME is general and flexible, there are some scenarios where simple perturbations are not enough, so there are other approaches like Anchor where perturbations variation depends on the dataset. In this paper, we propose a CBR solution to the problem of configuring the parameters of the LIME algorithm for the explanation of an image classifier. The case base reflects the human perception of the quality of the explanations generated with different parameter configurations of LIME. Then, this parameter configuration is reused for similar input images.

**Keywords:** Specific explanations · User experience · Model-agnostic explanations · Case-based explanations

## 1 Introduction

With the success of Machine Learning (ML) interpretability for ML systems have become an active focus of research. XAI research tries to solve several questions related to the increasing need for interpretable models, such as: How should interpretable models be designed? How do we evaluate the resulting explanations? What knowledge do we need for building explanations? How does interpretability change interactions between the AI systems and the users? What to explain? When to explain? How to deal with the fact that different users have different

---

Supported by the Spanish Committee of Economy and Competitiveness (TIN2017-87330-R) and UCM Research Group 921330.

© Springer Nature Switzerland AG 2020

I. Watson and R. Weber (Eds.): ICCBR 2020, LNAI 12311, pp. 179–194, 2020.

[https://doi.org/10.1007/978-3-030-58342-2\\_12](https://doi.org/10.1007/978-3-030-58342-2_12)

expectations and explanation needs? From the CBR perspective, research in XAI has pointed out the importance of taking advantage of the human knowledge to generate and evaluate explanations [16, 19].

At a high level, the literature distinguishes between two main approaches to interpretability: *model-specific* (also called transparent or white box) models and *model-agnostic* (post-hoc) surrogate models to explain black box models [12, 13, 24]. Transparent models are ones that are inherently interpretable by users. So, the easiest way to achieve interpretability is to use algorithms that create interpretable models, such as decision trees, nearest-neighbour or linear regression. However, the best performing models are often not interpretable, or they are interpretable only if features are few in number or where the model is sparse, and where the features have a readily understandable semantics [10]. Besides, for the sake of performance, it is typical to use ensembles of several models that cannot be interpreted, even if every single model could be interpreted, like in the random forest algorithm. Model-agnostic interpretation methods propose separating the explanations from the ML model. Although the main advantage is flexibility, as the interpretation methods can be applied to any model, some authors consider this type of post-hoc explanations as limited *justifications* because they are not linked to the real reasoning process occurring in the black box.

LIME [14] (Local Interpretable Model-Agnostic Explanations) is a well-known model agnostic model that attempts to understand the model by perturbing the input of data samples and understanding how the predictions change. The intuition to local interpretability is to determine which feature changes will have the most impact on the prediction. According to its authors, the algorithm fulfils the desirable aspects of a model-agnostic explanation system regarding flexibility. The LIME interpretation method can work with any ML model and is not limited to a particular form of explanation and representation. An essential requirement for LIME is to work with an interpretable representation of the input, like images or bag of words, that is understandable to humans. The output of LIME is a list of explanations, reflecting the contribution of each feature to the prediction of a data sample.

Although LIME is general and flexible, there are some scenarios where simple perturbations are not enough, so there are other approaches like Anchor [15] where perturbations variation depends on the dataset. Either in LIME or Anchor, the configuration variables are set up by default. However, the adequacy of the variables to the input query instance is critical to provide quality explanations. In fact, the type of modifications that need to be performed on the data to get proper explanations are typically *use case* specific. The authors gave the following example in their paper [14]: “a model that predicts sepia-toned images to be retro cannot be explained by presence or absence of superpixels”.

In this paper, we propose a CBR solution to the problem of configuring the default parameters of the LIME algorithm for an image classifier. The case base reflects the human perception of the quality of the explanations generated with different parameter configurations of LIME. Then, this parameter configuration is reused to generate explanations for similar input images.

This paper is organized as follows: Sect. 2 presents related work, whereas Sect. 3 introduces the LIME algorithm and some of its limitations. Section 4 describes the CBR-LIME method and the case base elicitation process. In Sect. 5 we demonstrate the benefits of our approach using both off-line and on-line evaluations. Concluding remarks are discussed in Sect. 6.

## 2 Related Work

CBR can provide a methodology to reuse experiences and generate explanations for different AI techniques and domains of applications. Therefore, we can find several initiatives in the CBR literature to explain AI systems. Some relevant early works can be found in the review by [8]. For example, [19] presents a framework for explanation in case-based reasoning (CBR) focused on explanation goals, whereas [2] develops the idea of explanation utility, a metric that may be different to the similarity metric used for nearest neighbour retrieval.

Recently there is a relevant body of work on CBR applied to the explanation of black-box models, the so-called *CBR Twins*. In [6], authors propose a theoretical analysis of a post-hoc explanation-by-example approach that relies on the twinning of artificial neural networks with CBR systems. [9] combine the strength of deep learning and the interpretability of case-based reasoning to make an interpretable deep neural network. [4] investigates whether CBR competence can be used to predict confidence in the outputs of a black box system when the black box and CBR systems are provided with the same training data. [23] demonstrates how CBR can be used for an XAI approach to justify solutions produced by an opaque learning method, particularly in the context of unstructured textual data. As we can observe, most of these works are post-hoc explanation systems, where CBR follows the model-agnostic approach to explain black-box models. However, there are other works that, instead of explaining the outcomes of the model, they try to explain the similarity metrics [17].

Outside the CBR community, many algorithms follow the same model-agnostic approach than LIME. Partial dependence plots (PDP) show the marginal effect that one or two features have on the predicted outcome of a machine learning model [3]. The equivalent to a PDP for individual data instances is called individual conditional expectation (ICE) plot [5]. It displays one line per instance that shows how the instance's prediction changes when a feature changes. Other approaches, referred to as permutation feature importance, measure the increase in the prediction error of the model after permuting the feature's values [1].

The global surrogate model is an interpretable model that is trained to approximate the predictions of a black box model [13]. In contrast, local surrogates, such as LIME or Anchors [14, 15], focus on explaining individual predictions. Another popular local surrogate model similar to LIME is SHAP [11]. It is based on the game theory concept of Shapley values and explains the prediction of an instance by computing the contribution of each feature to the prediction.

Once we have reviewed the most relevant contributions of CBR to XAI and presented an overview of model-agnostic explanation methods, the next section focuses on the LIME algorithm that is the basis of this paper.

### 3 Background

LIME focuses on training local surrogate models to explain individual predictions given by a global black-box prediction model. In a general way, it analyses the behaviour of the global prediction model through the perturbation of the input data.

In order to figure out what features of the input are contributing to the prediction, it perturbs the input data around its neighbourhood and evaluates how the model behaves. Then, it trains an interpretable local model that weights these perturbed data points by their proximity to the original input. This local model should be a good and explainable local approximation of the black-box model. Mathematically, it is formulated as follows [14]:

$$\text{explanation}(x) = \arg \min_{g \in G} L(f, g, \Pi_x) + \Omega(g) \quad (1)$$

This equation defines an explanation as a model  $g \in G$ , where  $G$  is a class of potentially interpretable models, such as linear models or decision trees. The goal is to minimize the loss function  $L$  that measures how close the explanation is to the prediction of the original model  $f$  given a proximity measure  $\Pi_x$ . This proximity measure defines the size of the neighbourhood around the predicted instance  $x$  that is used to obtain the explanation. Additionally, it is necessary to minimize the complexity (as opposed to interpretability) of the explanation  $g \in G$ , denoted as  $\Omega(g)$ .

Regarding the perturbation of the input data, it depends on its type. For tabular data, LIME creates new samples by perturbing each feature individually based on statistical indicators. For text and images, the solution is to remove words or parts of the image (called superpixels). Here, the user can also configure how these superpixels are computed and replaced. By default, LIME uses the Quickshift clustering algorithm [22] that finds areas with similar pixels using a hierarchical approach. This clustering algorithm depends mainly on the Gaussian kernel used to define the neighbourhoods of pixels considered, that in practice defines the number of clusters. Once the image has been segmented, it is necessary to perturb the image to generate the training set for the surrogate model by removing superpixels randomly. Next, the definition of the proximity measure  $\Pi_x$  should also be chosen carefully to select the neighbourhood of perturbed images. Current implementations of LIME use an exponential smoothing kernel where the kernel width defines how close an instance must be to influence the local model.

Finally, the interpretable surrogate model used by LIME is linear regression, corresponding to the  $\Omega(g)$  function in Eq. 1. Here, the user has to define the number of the top superpixels being considered. The lower top superpixels, the

**Table 1.** Variables used to configure the LIME method.

Clusters size	<b>C</b>	This parameter defines the width of the Gaussian kernel used to define the neighbourhoods of pixels considered
Number of perturbations	<b>P</b>	Number of perturbed images generated through the random removal of parts from the original image
Proximity measure	<b><math>\Pi</math></b>	Width of the exponential kernel that defines how close a perturbation must be to be included in the linear regression model
Number of features	<b>F</b>	Number of superpixels being considered by the linear regression model, representing to the $\Omega(g)$ function in Eq. 1

easier it is to interpret the model. A higher value potentially produces models with higher fidelity.

The use of linear regression makes LIME unable to explain the model correctly on some scenarios where simple perturbations are not enough. Ideally, the perturbations would be driven by the variation that is observed in the dataset. The same authors proposed a new way to perform model interpretation which is Anchors [15]. Anchor is also a local model-agnostic explanation algorithm that explains individual predictions, i.e., only captures the behaviour of the model on a local region of the input space. However, it improves the construction of the perturbation data set around the query. Instead of adding noise to continuous features, hiding parts of the image, to learn a boundary line (or slope) associated to the prediction of the query instance, Anchors improves LIME using a “local region” instead of a slope. Nevertheless, it also uses a generic configuration for every image.

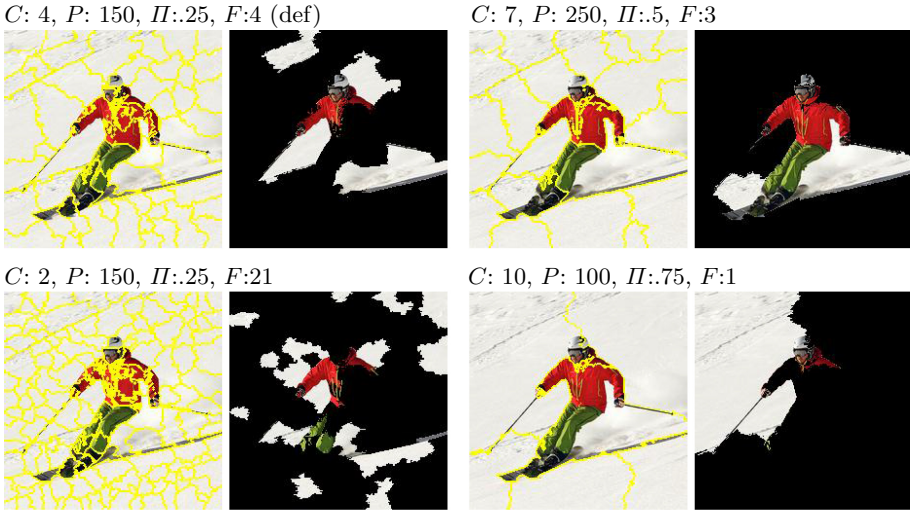
Once we have described LIME and its limitations, the next section introduces the CBR-LIME method that improves its configuration through a case-based reasoning process.

## 4 The CBR-LIME Method

As explained in the previous section, instead of using the default LIME setup, its configuration can be optimized in order to achieve higher performance. Here, an image-specific configuration of these parameters is critical in order to obtain good explanations. In our approach, we will consider the parameters to configure the LIME method listed in Table 1. In this table we have selected those parameters with a higher impact in the final explanation after a preliminary evaluation based on the results obtained by the LIME implementation provided by the authors<sup>1</sup>. Figure 1 illustrates the impact of these parameters, showing the resulting explanations for a given image when applying different LIME configurations. In this case, the underlying neural network classifier identifies the

<sup>1</sup> <https://github.com/marcotcr/lime>.

image as “ski”. However, the visual explanations provided by LIME change significantly depending on its setup. As we can observe, the explanation generated using the default parameters (top-left pair) is not a proper choice to explain the outcome of the classifier.



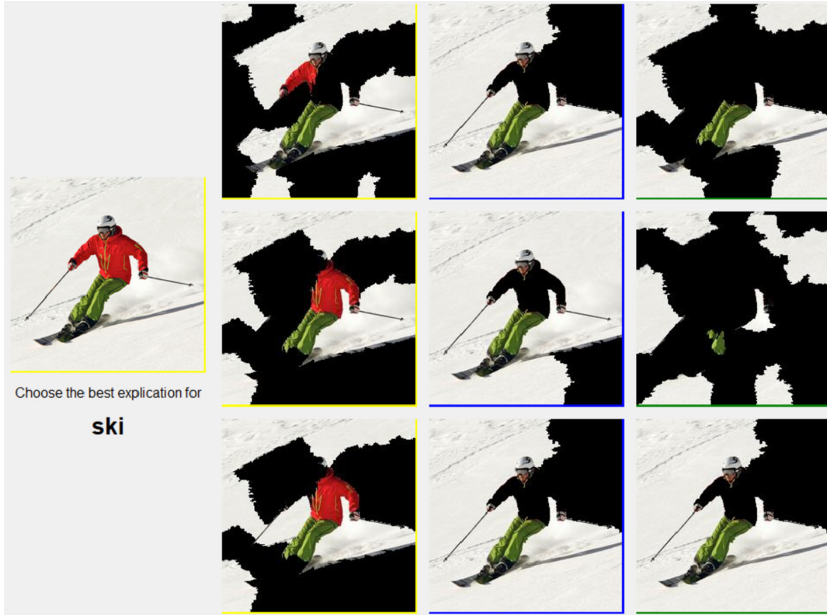
**Fig. 1.** Examples of LIME explanations for the same image using different setups. Each pair shows the image segmentation on the left and the explanation generated according to the parameters above. Top-left pair corresponds to the default values of the LIME implementation.

A straightforward solution is to adjust these parameters according to the predicted instance. However, as explanations depend on their utility to the user, it is not possible to find an algorithmic solution to compute the best setup. Therefore, we propose the use of a CBR approach where a case base of instances and their most suitable configuration for LIME is collected and reused to provide explanations.

#### 4.1 Case Base Elicitation

To ease the evaluation of explanation cases with users, we have focused on the LIME method for images. The case base of images has been obtained from the dataset provided by the Visual Genome project [7]. We selected 200 images that were confidently classified by Google’s Inception deep convolutional neural network architecture [21] with a predominant class (*precision* > 95%). For every image, we generated eight different explanations through the heterogeneous configuration of the variables in Table 1, plus the default configuration of LIME. Then, these nine explanations were presented to users, that could select

the most suitable explanatory image, as illustrated in Fig. 2. Explanations were randomly shuffled, and the corresponding LIME configuration is not displayed to the user. Each time the user selects an explanation, a new image and its corresponding explanations are shown until the 200 images have been voted. Concretely, users were asked to select the most specific explanation, meaning that, in case of two similar images, they should choose the one with less image area.



**Fig. 2.** Application used to vote for the best explanation and generate the case base. The original image and the majoritarian predicted class is shown on the left. Images on the right are generated through 8 random configurations of LIME plus the default setup.

After repeating this process with 15 users we collected a total of 3.000 votes (15 per image) that were used to generate the case base. The description of each case is the image itself (its pixel matrix) plus the feature’s vector returned by the classifier. Then, the solution of each case is the average of the values for  $C, P, \Pi$  and  $F$  from the LIME configurations chosen by the users. This representation of cases can be formalized as:

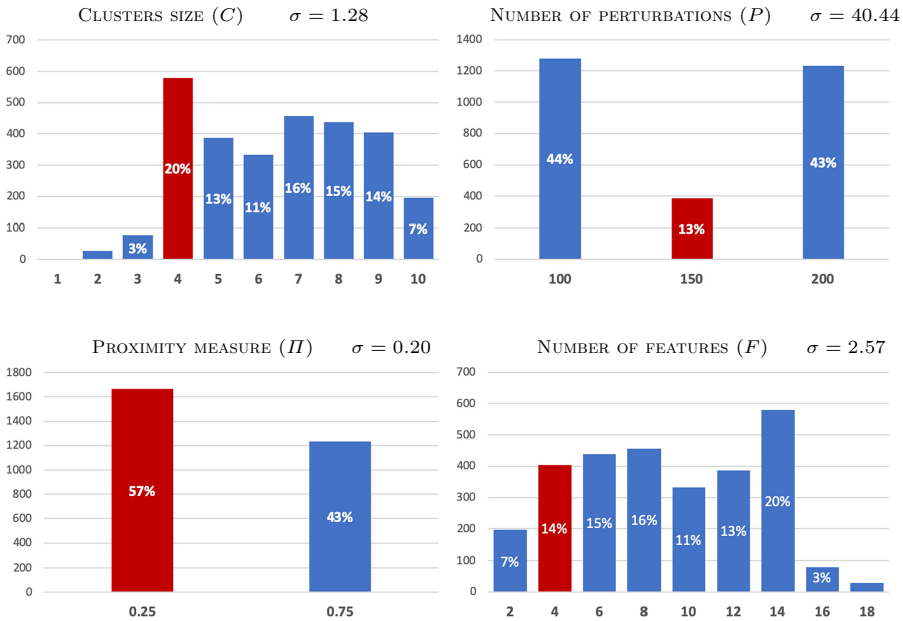
$$Case = \langle D, S \rangle \tag{2}$$

where

$$D = \langle image, \mathbf{f} \rangle$$

$$S = \langle C, P, \Pi, F \rangle$$

The analysis of the configurations voted by the users confirmed our initial hypothesis stating that the default configuration of LIME is not suitable for a general-purpose explanation. As Fig. 3 shows, the default configuration values for each parameter (red columns) are not predominant, and there is significant heterogeneity. This conclusion is also contrasted by the analysis of the variability on the user’s choices. If we compute the standard deviation of the configuration values chosen for every image, we can study if users tend to select a similar configuration for LIME as the best explanation. Through this analysis, we collaterally validate the central hypothesis of this paper, consisting of applying a case-based reasoning solution to generate LIME explanations because similar images should be explained using similar configurations of the algorithm. The corresponding average standard deviation values are also displayed in Fig. 3. As we can observe, this analysis validates our hypothesis as the variability on the configurations chosen by the users is quite low, especially for the  $C$  and  $F$  variables.



**Fig. 3.** Histograms describing the values (x-axis) chosen by users when voting for the best explanation. Red columns highlight the default values in LIME. Numbers inside columns reflect the percentage of explanations chosen by users that were configured with the corresponding value in the x-axis.  $\sigma$  values correspond to the average of the standard deviation for each image. (Color figure online)

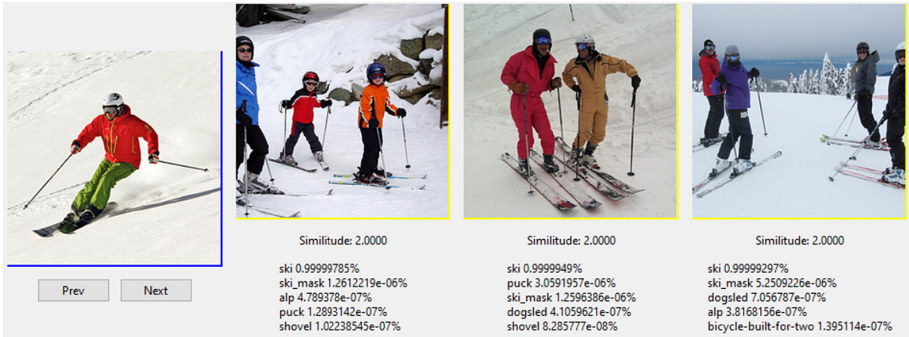


### 4.2 Case-Based Explanation

Once the case base has been generated, we can define the CBR process used to find the most suitable configuration for LIME given an instance and its corresponding classification by the global model. The first step is the retrieval of similar images (and their corresponding LIME configurations) from the case base. A straightforward method to retrieve similar images is the comparison of the pixel matrix. However, in practice, this approach is not a good choice because we must focus on the objects in the image that were identified by the global model. Therefore, we have defined the retrieval process as the comparison of the feature vectors  $\mathbf{f}$  given by the global model. This way, once we have the classification of the query image ( $q$ ), we can compare its feature vector with the vectors describing the cases simply by applying a distance metric such as the Euclidean distance.

$$sim(D_q, D_x) = Eucl\_Dist(\mathbf{f}_q, \mathbf{f}_x) \tag{3}$$

Then, the  $k$  most similar images can be selected. This retrieval process is illustrated in Fig. 4, where the three most similar images (yellow border) to the query image (blue border) are displayed together with their feature vectors  $\mathbf{f}$ . Here we can observe that the feature-based similarity achieves our goal of retrieving related images and avoids problems associated with pixel-based comparisons such as colour or image contrast.



**Fig. 4.** Application used display the similarity between images using the features identified by the global model (Eq. 3). (Color figure online)

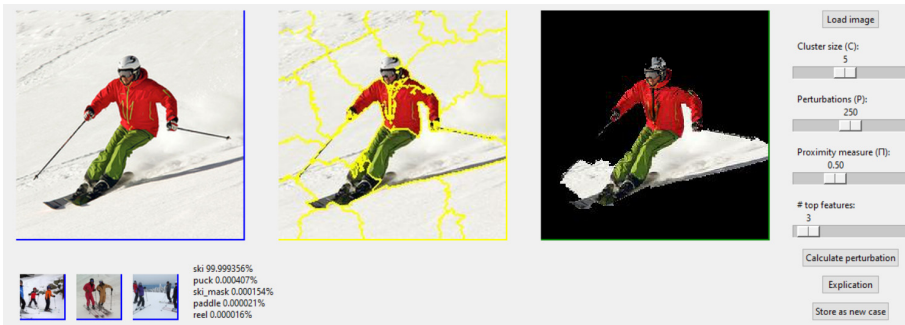
The following step in the CBR cycle is adaptation. Here, the final configuration for the LIME algorithm is calculated as the average of the configurations of the  $k$  most similar cases. This way, we are reusing the user’s experience to generate the explanation instead of applying a setup by default.

$$S_q = \bigvee_{x \in kNN(q)} \langle \overline{C}_x, \overline{P}_x, \overline{\Pi}_x, \overline{F}_x \rangle \tag{4}$$

Then, the generated explanation is presented to the user that can revise the configuration values in order to adjust its quality. Finally, the user can store the new generated case into the case base to close the CBR cycle. Figure 5 shows a capture of the CBR application that implements this process.

## 5 Evaluation

In order to demonstrate the benefits of CBR-LIME we have conducted two complementary evaluations. Firstly, an offline evaluation compares the explanatory images generated by the default LIME setup and our case-based approach using cross-validation. Secondly, we implemented an online evaluation with users similar to the experiment described in Sect. 4.1. This time, explanatory cases are shown, and users must vote the most suitable explanation. Both offline and online evaluations are presented next.

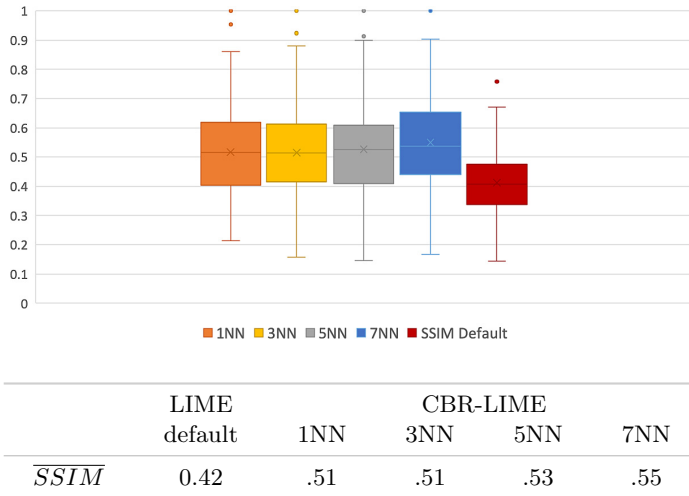


**Fig. 5.** Application implementing the full CBR cycle. It shows the original image, its associated perturbation and the resulting explanatory image given by the configuration obtained by CBR-LIME (Eq. 4). This configuration can be revised by the user, that also can store the generated new case into the case base.

### 5.1 Offline Evaluation

The goal of the offline evaluation is to compare, using an image similarity metric, the explanatory images generated by the default LIME setup and different configurations of our CBR-LIME method. Given any image in a case of our case base, we can compute the “optimal” explanatory image (according to the users’ votes) through the configuration stored in its solution. Then, other explanatory images generated with different configurations of LIME can be compared to this optimal explanation in order to measure their quality. If we repeat this process throughout the whole case base using a leave-one-out approach we can evaluate the performance of the default LIME setup in contrast to the configurations provided by our CBR-LIME method (with different  $k$  values: 1NN, 3NN, etc.).

A key element in this evaluation is the similarity metric used to compare the explanatory images. There is an extensive catalogue of such metrics in the field of Image Quality Assessment (IQA) that must be carefully chosen depending on the nature of the image and the type of comparison that is required [18]. In our case, we need to compare variations of the same original image where some parts have been removed. Therefore, we need a metric that is able to compare the structural changes in the image, such as the Structural SIMilarity (SSIM) index. This metric that has demonstrated good agreement with human observers in image comparison using reference images [25]. The SSIM index can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality. It combines three comparison measurements between the samples of  $x$  and  $y$ : luminance, contrast and structure. In our evaluation, the explanation generated with the (average) configuration chosen by the users is the image of perfect quality to compare with. In contrast, the explanations generated with other configurations of LIME (default, 1NN, 3NN, ...) are the variations that we need to find out their comparative quality.

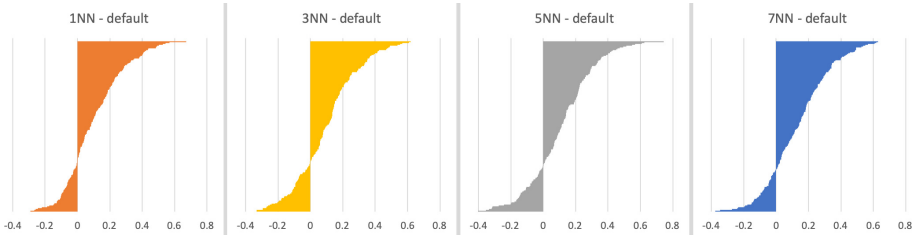


**Fig. 6.** Boxplot (top) and average SSIM values (bottom) obtained when comparing explanatory images generated with different LIME configurations.

Results are summarized in Fig. 6 that shows a boxplot (top) and the average (bottom) of the SSIM values obtained by the explanatory examples generated with different configurations. We can observe that the SSIM index is higher using the CBR-LIME method. As we have computed the SSIM index for the 200 images in the case base we can contrast the resulting series in order to validate this improvement statistically. Therefore, we have run a two-pair Wilcoxon signed-rank test comparing the SSIM indexes obtained by the default LIME setup and the values from the CBR-LIME configurations. In all cases, the improvement was

statistically significant at  $p < 0.05$ . However, there is a little improvement when increasing the  $k$  parameter of the CBR-LIME method, finding only statistical evidence between  $k = 7$  and  $k = 1, 3$ .

These series comparisons are graphically presented in Fig. 7 that plots the difference between the SSIM values obtained by the kNN configurations and the default LIME setup. As we can observe, the positive area (on the right side of the y-axis) is much larger than the negative, indicating that the explanations generated by CBR-LIME are more similar to the optimal explanatory image.



**Fig. 7.** Plots of the differences between the SSIM index obtained by the k-NN configurations minus the default LIME setup for every image.

## 5.2 Online Evaluation with Users

We have also conducted an online evaluation with users to corroborate the results of the offline analysis. In this case, users had to choose between two explanatory images: one is generated with the default LIME setup, and the other generated from the configuration obtained by CBR-LIME<sup>2</sup>. The application used to conduct this evaluation (Fig. 8 left) shows the original image, the classification



**Fig. 8.** (left) Application used in the online evaluation where users have to vote for the best explanation comparing the images generated by the default LIME setup and the CBR-LIME configuration. (right) Percentage of votes given by the users to each alternative (1600 total votes).

<sup>2</sup> Explanations were generated using 3-NN as there are no significant changes with other  $k$  values.

given by the global model, and the two explanatory images. One more time, this application shuffles the images to avoid any kind of bias in the users' choices, and the voting process must be repeated for all the images in the case base.

After collecting 1600 votes, results corroborate the benefits of CBR-LIME, as 76.7% of the images selected by the users as the best explanation were generated using the configuration provided by our method.

## 6 Conclusions and Future Work

This paper presents a Case-based reasoning method that takes advantage of human knowledge to generate explanations. Concretely, we have defined and evaluated a CBR solution to the problem of configuring the well-known LIME algorithm for images. This algorithm attempts to understand a global black-box classification model by perturbing the input of data samples. However, this method applies a generic setup for any image, that leads to inadequate explanations as demonstrated in this paper through an evaluation performed with 200 images and 15 users. This evaluation let us collect a case base of images and their associated "optimal" LIME configurations according to the users. From this case base, we can implement a CBR-LIME method where, given a new query image, similar images are retrieved, and their corresponding configurations are reused to generate an explanation through the LIME algorithm.

To validate CBR-LIME, we have conducted two complementary evaluations. The offline evaluation compares through cross-validation the explanatory images generated by the default LIME setup and the configurations obtained by CBR-LIME to the "optimal" explanation according to the users. To compare the images, we use the SSIM image comparison index, that is a reference method in image quality assessment, able to compare variations of the original image. The results of the offline evaluation demonstrated that CBR-LIME improves up to 13% the similarity of the generated images with the optimal explanation. Then, we conducted an online evaluation with real users in order to corroborate these results. In this case, users had to choose between two explanations for the same image, one generated with the default LIME setup, and the other with CBR-LIME. Again, the results confirmed the benefits of the later as it obtained 76% of the votes.

This paper leaves many open lines for future work. Firstly, we would like to explore the impact of other configuration parameters of LIME that were considered initially as less relevant to generate the explanation. For example, there is a ratio threshold in the Quisckshift algorithm that defines the trade-off between colour importance and spatial importance to create image clusters. This parameter was not included in CBR-LIME because initial evaluations did not demonstrate a significant impact on the performance of the method. However, this must be methodologically validated.

The combination of these parameters as the solution of the cases also requires further evaluation. Obviously, during the case base elicitation process, users did not choose the same best explanation for a particular image. We, therefore,

obtained several LIME configurations for each image, that were averaged to compute the final solution of the case. Thus, other alternatives may be considered and evaluated, i.e., the median value or just selecting the most voted configuration.

We must also analyze the impact of the case base quality in the explanation process regarding cold-start scenarios where no similar images are available in order to find out the minimum similarity threshold and class distributions required to provide good explanations. Also, our evaluation only includes images that are confidently classified by the neural network, so we need to evaluate the impact of incorrect or ambiguously classified images. Additionally, the impact of user bias in the case base elicitation and evaluation must be carefully analyzed too.

Another relevant line of future work is the improvement of the similarity metric. Equation 3 does not take into consideration the pixel matrix of the image to retrieve similar cases. However, it was our initial idea, and we tested the SSIM index and other feature matching methods like FLANN [20] as similarity metrics. Unfortunately, results were disappointing due to the variability of the images in the case base. So we discarded the pixel matrix comparison and focused on the similarity of the image features. Nevertheless, further research is required in order to enhance the similarity metric by including pixel-based comparisons.

An open implementation of CBR-LIME in Python is available at:

<https://github.com/UCM-GAIA/GBR-LIME>.

## References

1. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
2. Doyle, D., Cunningham, P., Bridge, D.G., Rahman, Y.: Explanation oriented retrieval. In: Funk, P., González-Calero, P.A. (eds.) *Advances in Case-Based Reasoning, ECCBR 2004*. Lecture Notes in Computer Science, vol. 3155, pp. 157–168. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-28631-8\\_13](https://doi.org/10.1007/978-3-540-28631-8_13)
3. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29**(5), 1189–1232 (2001). <https://doi.org/10.1214/aos/1013203451>
4. Gates, L., Kisby, C., Leake, D.: CBR confidence as a basis for confidence in black box systems. In: Bach, K., Marling, C. (eds.) *Case-Based Reasoning Research and Development, ICCBR 2019*. Lecture Notes in Computer Science, vol. 11680, pp. 95–109. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-29249-2\\_7](https://doi.org/10.1007/978-3-030-29249-2_7)
5. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. *J. Computat. Graph. Stat.* **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>
6. Keane, M.T., Kenny, E.M.: How case-based reasoning explains neural networks: A theoretical analysis of XAI using post-hoc explanation-by-example from a survey of ANN-CBR twin-systems. In: Bach, K., Marling, C. (eds.) *Case-Based Reasoning Research and Development, ICCBR 2019*. Lecture Notes in Computer Science, vol. 11680, pp. 155–171. Springer, Heidelberg (2019). [https://doi.org/10.1007/978-3-030-29249-2\\_11](https://doi.org/10.1007/978-3-030-29249-2_11)

7. Krishna, R., et al.: Visual genome: connecting language and vision using crowd-sourced dense image annotations (2016). <https://arxiv.org/abs/1602.07332>
8. Leake, D.B., McSherry, D.: Introduction to the special issue on explanation in case-based reasoning. *Artif. Intell. Rev.* **24**(2), 103–108 (2005). <https://doi.org/10.1007/s10462-005-4606-8>
9. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: a neural network that explains its predictions. In: McIlraith, S.A., Weinberger, K.Q. (eds.) *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18*. pp. 3530–3537. AAAI Press (2018)
10. Lipton, Z.C.: The mythos of model interpretability. *Commun. ACM* **61**(10), 36–43 (2018). <https://doi.org/10.1145/3233231>
11. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in neural information processing systems*, **30**, pp. 4765–4774. Curran Associates, Inc. (2017)
12. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. *CoRR abs/1706.07269* (2017). <http://arxiv.org/abs/1706.07269>
13. Molnar, C.: *Interpretable Machine Learning* (2019). <https://christophm.github.io/interpretable-ml-book/>
14. Ribeiro, M.T., Singh, S., Guestrin, C.: “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*, pp. 1135–1144. Association for Computing Machinery, New York, NY, USA (2016). <https://doi.org/10.1145/2939672.2939778>
15. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: McIlraith, S.A., Weinberger, K.Q. (eds.) *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-2018*, pp. 1527–1535. AAAI Press (2018). <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982>
16. Roth-Berghofer, T., Richter, M.M.: On explanation. *Künstliche Intelligenz KI* **22**(2), 5–7 (2008)
17. Sanchez-Ruiz, A.A., Ontanon, S.: Structural plan similarity based on refinements in the space of partial plans. *Computat. Intell.* **33**(4), 926–947 (2017). <https://doi.org/10.1111/coin.12131>
18. Sheikh, H.R., Sabir, M.F., Bovik, A.C.: A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. Image Process.* **15**(11), 3440–3451 (2006). <https://doi.org/10.1109/TIP.2006.881959>
19. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in case-based reasoning—perspectives and goals. *Artif. Intell. Rev.* **24**(2), 109–143 (2005). <https://doi.org/10.1007/s10462-005-4607-7>
20. Stju, D.A., Jose, H.: Flann: Fast approximate nearest neighbour search algorithm for elucidating human-wildlife conflicts in forest areas. In: *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, pp. 1–6, March 2017. <https://doi.org/10.1109/ICSCN.2017.8085676>
21. Szegedy, C., et al.: Going deeper with convolutions. In: *Computer Vision and Pattern Recognition (CVPR)* (2015). <http://arxiv.org/abs/1409.4842>
22. Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *Computer Vision - ECCV 2008*, pp. 705–718. Springer, Heidelberg (2008)

23. Weber, R.O., Johs, A.J., Li, J., Huang, K.: Investigating textual case-based XAI. In: Cox, M.T., Funk, P., Begum, S. (eds.) *Case-Based Reasoning Research and Development, ICCBR 2018, Proceedings*. Lecture Notes in Computer Science, vol. 11156, pp. 431–447. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01081-2\\_29](https://doi.org/10.1007/978-3-030-01081-2_29)
24. Weld, D.S., Bansal, G.: The challenge of crafting intelligible intelligence. *Commun. ACM* **62**(6), 70–79 (2019). <https://doi.org/10.1145/3282486>
25. Zhou, W., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861>