UTRECHT UNIVERSITY

BACHELOR ARTIFICIAL INTELLIGENCE

THESIS (7.5 EC)

Analysis on Local Model Explanations by LIME

Author
J.E. VAN BUSSCHBACH
6800831

Supervisor
Wijnand VAN WOERKOM

Second Examinor
Jan BROERSEN

June 30, 2023

Abstract

In the field of eXplainable Artificial Intelligence, many algorithms have been proposed to try to get an understanding of black box machine learning models. LIME is one of these proposed algorithms, and since its inception has been heavily discussed in the literature. To verify its functionality, our research aims to reproduce an experiment from the paper in which the algorithm is originally proposed [7]. This experiment tests the functionality of LIME explaining the predictions of a black box classifier trained on a biased image dataset. After performing the experiment, we found our results portrayed the bias of the image dataset similarly to the results of the original experiment, and were useful for determining the trustworthiness of the classifier. However, we also find the algorithm is volatile and heavily dependant on parameters for local sampling. These limitations prevent the algorithm from accurately finding nuanced biases, and deter the technique from being trustworthy for serious applications in the grander scheme of Artificial Intelligence.

1 Introduction

The usage of artificial intelligence has increased substantially over the last few decades, being applied in many different job sectors. The main utilization of these new techniques is automatization for what was previously done by human workers. Repetitive tasks for data science and accounting are already being automised by machine learning models, but this technology is also being deployed for more complex tasks such as aiding health practitioners and aiding in the judicial system. With these tasks being more complex and having substantial consequences for human lives, it becomes important for us to be able to trust the decisions being made by complex machine learning models. At the inception of machine learning, this was not yet a problem as the models used were of small enough complexity for humans to understand. However, with the rise of neural network based models, the machine learning models we deploy for tasks are black-box based. These models are not easily interpretable by the people determining the correctness of their predictions. This gap in interpretation caused a movement for eXplainable Artificial Intelligence (XAI) [6] trying to provide interpretations for the black-boxes generated by complex machine learning models. XAI now forms an important discipline in the study of artificial intelligence and its importance is widely understood.

Machine learning models are not uninterpretable per definition, as linear classification models do provide a model from which an explanation for

a prediction can be inferred. For example, an algorithm using linear classification for determining whether a 32 by 32 pixel image is a representation of the number 9 can be interpreted by humans by looking at how much weight each pixel was assigned to. If the model works correctly, the pixels with the highest weights will form something partially presenting a nine. As per the example, this works well with linear models as the weights can be directly looked at. However with deep learning, much of the weights are contained in a black box model and correspond to more difficult to interpret patterns. XAI algorithms such as LIME and SHAP [8] form a method to look beyond the black box, and try to find interpretable explanations for these deep learning models.

This paper will mainly focus on the Local Interpretable Model-agnostic Explanations (LIME) technique as proposed by Ribeiro et al. in 2016 [7]. The reason LIME is chosen for this research is that the robustness and accuracy of LIME is heavily discussed in literature [2, 4, 8], with one source bringing up theoretical proof that LIME should work on simple datasets of words [4] and images [3]. However, other sources are more critical of LIME, and point out that it is highly unstable with regard to changes in its parameters [8].

We will test the accuracy of the algorithm originally proposed by writing our own implementation focused on generating explanations for image classification models. We will use this implementation to recreate an experiment done by Ribeiro et al. [7] and verify that we get similar results. In this experiment, a highly biased image dataset is used to train a logistic regression classifier, causing predictions of the classifier to be based mostly on the bias from the image dataset. The LIME algorithm is then used to generate explanations which accurately depict the trained bias of the logistic regression model. We hypothesize that the LIME algorithm will generate explanations as expected and give similar clarity to the bias as the original experiment. After performing the experiment, our hypothesis is confirmed with the generated explanations clearly presenting that the predictions of the classifier are affected by a bias. However, our implementation of the LIME algorithm also highlights some concerns with the stability of the algorithm making its use for more critical or substantial use cases unreliable.

This paper will start off by giving a general overview of the LIME algorithm and the experiment which will be reproduced, along with a justification for the expected results. Then, a detailed report is given of our implementation, after which the results of this experiment are presented with an analysis of the effectiveness of the algorithm.

2 Relevance in Artificial Intelligence

This papers main connection to the broader field of artificial intelligence is the discussion of an eXplainable Artificial Intelligence technique. We provide an analysis on the LIME algorithm, which allows humans to get an insight in predictions from black-box machine learning models. As these models tend to be very complex and thus not easily graspable by humans, it becomes beneficial to acquire explanations from machine learning models which accurately portray their inner workings. Aside from increasing trustworthiness of the prediction by giving a justification, the explanation can also be used to evaluate the performance of a model as well as quality of the data set. For example, a study has found that an automated judicial system possesses a bias towards historically marginalized groups, prompting the system to assign a high risk of recidivism for convicts of those groups more often compared to other ethnical groups [1]. XAI techniques revealed this bias, highlighting the problematic dataset that the system used, which contained a disproportionate number of offenders from marginalized groups due to historical issues. Because of these findings, the systems prediction were found to be much less trustworthy and future systems can be made with these limitations in mind.

3 Preliminaries

Before discussing our reproduction, it is important to have an understanding about what an explanation for a machine learning model would entail, the general concept of LIME and a detailed overview of the experiment we will reproduce.

3.1 Interpretable Explanations

To properly evaluate and trust the performance of a machine learning model, explanations are created which try to convey the *process* a model used to get to its prediction. Phillips et al. refer to two approaches to explain this process [6]. *Local* explanations try to explain a single input/output pair of a model. This is useful to understand what parts of an input the model used to get to its prediction of a specific instance. The other approach is a *global* explanation, which tries to give information about the entire algorithm. These explanations help understand the model's general biases and reliability across various input contexts.

As its name suggests, LIME (*Local* Interpretable Model-agnostic Explanations) is a technique for generating local explanations. Ribeiro et al. define an explanation g as a model which covers the domain of presence/absence of interpretable features of an input [7]. These interpretable features come in many forms, but have as most important criterium that they must be understandable to humans, regardless of whether they are used by the model they are supposed to explain. As an example, for image classification, an interpretable representation of an image might be a vector containing the presence or absence of a contiguous patch of similar pixels (superpixel), while the classifier might represent the image as a tensor with three channels per pixel [7].

In addition to the interpretable features, the complexity of the explanation model also needs to be taken into account. An explanation model with a high complexity is not beneficial, as the explanation becomes overwhelming for humans to comprehend. For instance, were the explanation model a decision tree, the depth of the tree would greatly influence how much information humans could grasp from the model. For our image classification example, this means that the amount of superpixels needs to be limited as not to have the image divided in too many regions. Overly small regions are more difficult to interpret and thus some limit needs to be set. This comes with a trade-off however, as acting over the presence/absence of large patches of pixels will lessen the finesse of the explanation, because only large difference can be evaluated. Thus a great explanation model tries to get the complexity low enough to be easily interpreted by humans, whilst minimizing its error in assessing the impact of the presence/absence of the interpretable features.

Lastly, note that the explanations discussed here are explanations of a black box model for a single instance of prediction. That means that the explanation g might cover explaining one instance accurately, whilst not correctly conveying information about another instance, the explanations are thus *local* to the instance being explained and do not reflect on all choices of the black box (some intuition behind this can be seen in Figure 1). As black box models often contain many more input variables than humans can comprehend, capturing all the intricacies of its decisions becomes a most unfeasible task. With these local explanations, accurate insight can be obtained whilst keeping the explanations human understandable.

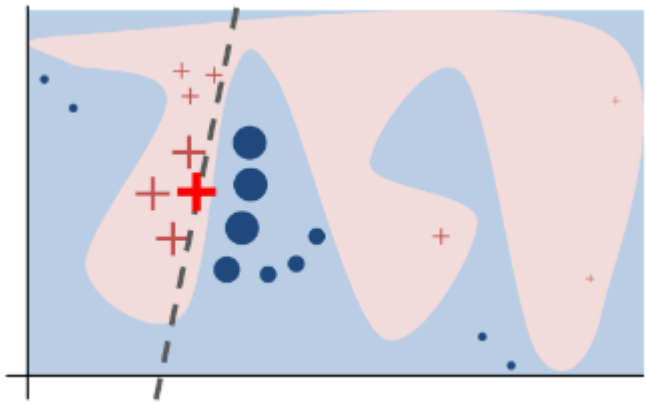

Figure 1: An example from the original paper to explain the intuition behind LIME. The blue/pink background represent the complex decision function f of a black box model. The bold red cross is the instance being explained. Perturbations are generated and weighed according to their proximity to the instance being explained, after which predictions are gotten using f (represented by the crosses and dots, proximity represented by size). The dashed line is the learned explanation that should be faithful to the instance we wanted explained (but not globally to f) [7].

3.2 Overview of LIME

The LIME technique allows the output for complex machine learning models to be evaluated in an interpretable manner. Let f be a model represented by a function mapping input of the model to output given by the model. Let x be an input for model f. The aim of LIME is to generate an explanation ϕ for $f(x)$. This explanation should be human interpretable, meaning that a (trained) person should get insight from ϕ what features f used of x to output $f(x)$. To achieve this, we create an interpretable version x' of x which tries to represent x in a form that would be human interpretable. The paper [7] mentions for example that were x a data resource and f a text classifier, x' could be a vector containing every word present in x and ϕ could be a binary vector indicating whether each word in x' influenced the prediction $f(x)$ or not. To generate ϕ, the LIME technique goes through the following algorithm.

First, a data set of $n \in \mathbb{N}$ perturbations locally sampled around x is created. A perturbation is created as follows. From x', a random amount of entries are turned off resulting in an interpretable version of x with a

random amount of features removed. This version is then converted to its original representation. Let the perturbation in the interpretable representation be called z' and in the original representation be called z.

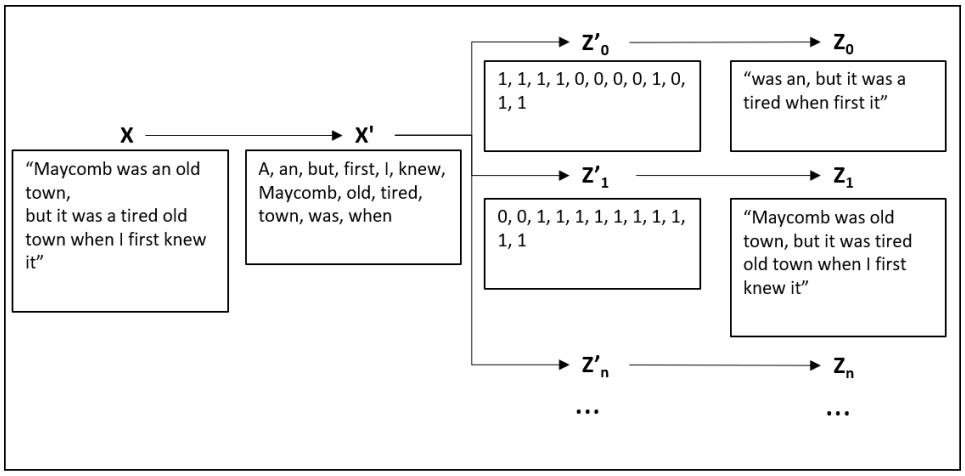

Figure 2: An example of how perturbations could be created of a text source. An interpretable version is made, from which a random amount of features are turned off for each perturbation.

Next, a simple classifier model is created, g. This classifier is trained on a dataset Z' which is filled with perturbations z' of the datapoint x. The label of a datapoint $z' \in Z'$ is given by $f(z)$, the label assigned by the model f to the perturbation in the original representation. Each $z' \in Z'$ is weighted according to the distance between z and x. By training g, we get the explanation ϕ by evaluating the learned weights g has for each feature in x'.

3.3 The Experiment

The experiment which will be implemented for this paper is described under Section 6.2 of the original proposal for the LIME algorithm [7]. In this experiment, a logistic regression classifier is trained with a highly biased image data set. The data set consists of images of huskies and wolves, with every picture of a wolf specifically chosen with snow visible in the lower half of the picture. A picture of a wolf from the trainings data is chosen to generate an explanation for. Using a segmentation algorithm, this image is divided into superpixels (continuous patches of similar pixels) which will

be used as the interpretable features of this image. The LIME technique is then used to generate an explanation for this model. These highlight that the lower half of the image is primarily being used to assess whether the picture is of a wolf or a husky, see Figure 3 for an example taken from [7].

(a) Husky classified as wolf (b) Explanation

Figure 3: Example of a husky classified as wolf with explanation from the original paper [7].

We hypothesize that LIME will generate explanations similar to explanations of the original experiment. Overall, this would mean that the explanations show the predictions of the logistic regression classifier are biased and classifier should not be trusted to differentiate wolves from huskies like humans do. Specifically, we should see that when the logistic regression classifier predicts "wolf" for an image of a husky with snow in the lower half of the picture, the explanation should look similar to Figure 3. The explanation should show that snowy segments in the picture influence the prediction of the logistic regression classifier.

We justify our hypothesis with the following. The training data will have consistent pixel values in the lower half of the pictures for the wolf photos and inconsistent pixel values for the husky photos. The logistic regression classifier will set each pixel as an individual feature and can thus discover that wolf photos have a consistent pattern whilst husky photos do not, and thus will overfit to the lower half of the image. Then, for local sampling of perturbations, perturbations with the lower half of the picture removed will cause a prediction of a husky, and perturbations with the

lower half of the picture present will cause a prediction of a wolf. The linear regression classifier trained on these data and labels will fit and assign bigger weights to the superpixels in the lower half of the picture compared to upper half, which we can use as proper explanation for viewing the bias in the original logistic regression classifier.

4 Method

In this section we set out what we used to reproduce the experiment. We follow the LIME technique from the original paper by Ribeiro et al. [7], however some modifications are made to specific details which will be highlighted in the Analysis & Discussion section.

4.1 Experiment Setup

For our implementation of the experiment Python version 3.10 was used as the main coding language, making heavy use of the scikit-learn (version 1.2.1) library [5] to handle the machine learning models and scikit-image (version 0.19.3) library [10] for image manipulations.

4.1.1 Image Dataset

For the image dataset, a variety of images were acquired from Google Images which were scaled to 128 by 128 pixels using the scikit-image library. These images cover two classes: photos of wolves in a snowy area and photos of huskies. The pictures where chosen such that every picture of a wolf contained snow in the lower half of the picture and every picture of a husky did not. No further criteria were given to the pictures, so the animals are depicted in varying poses and in varying positions on the photos. Fifteen photos in each class were chosen, with 10 being used for the training set and 5 for the test set. The logistic classifier used as the black box for the experiment was taken from the scikit-learn library, as well as the linear regression classifier used for the explanation. Three pictures were chosen to create an explanation for, shown in Figure 4. Pictures (a) and (b) are taken from the wolf and husky dataset respectively. Picture (c) contains a husky with snow in the lower half of the picture.

Figure 4: The images used for the experiment.

4.1.2 Sampling Perturbations

An explanation for each picture was generated following the LIME algorithm. The image in original representation x was gotten by loading the image and resizing it to be 128 by 128 pixels using the scikit-image library tools. This array was used to divide the picture into segments using the SLIC segmentation algorithm as provided by the scikit-image library. We set the algorithm to find 100 segments. Each segment found represents a superpixel. From this we get interpretable representation x' which is a one dimensional vector containing every segment number. In Figure 5 the process is shown for a picture of a wolf.

Figure 5: Instance of the original wolf picture (left) and the image after resizing (right) with the segments found by SLIC denoted with a yellow border. SLIC was used with $n\_segments = 100$ and found 72 unique segments.

After creating the interpretable representation, a collection of 5000 per-

turbations are generated. A set of vectors Z' in which each vector has the same size as x' is created with each vector only containing true values. For each vector z' in Z', a uniformly random amount of values are turned to *false*, with at least one value in z' being *true* after finishing. These vectors represent which segments found in x should stay on, *true*, and which should be turned off, *false*. Each vector in Z' is also converted to the original representation z by copying image x and turning each segment which is denoted *false* by z' gray. These images are collected in set Z. We also keep track of the distance between each perturbation z and the original image x. This was done by turning both z and x grayscale, flattening the arrays and then calculating the euclidean distance between the two. These weights are then collected in W.

Figure 6: A couple of perturbations in original presentation. The color of the turned off superpixels is gray.

4.2 Generating Explanations

After following the setup, the explanations were generated as follows. First, the logistic regression model f was trained using the biased image dataset. After fitting to the dataset, the sampled perturbations Z were used to get predictions $f(Z)$ which were used as labels for the training data set Z'. The linear regression model is trained using Z' as training data and $f(Z)$ as training labels. The weights collected in W are used as sample weight to adjust each entry in Z' by. After the linear regression model was fitted, the weights were gathered. With each weight corresponding to a segment from x', an explanation image was generated by copying x and giving the five segments with the highest weight a green color in the image.

5 Results

The aim of the experiment was to test if we get similar results compared to the original experiment as done by Ribeiro et al. [7]. To evaluate this, it is important to have an understanding on both the performance of the logistic regression model and the explanations provided by following the LIME algorithm, as the logistic regression model directly influences the generated explanation.

5.1 Black Box Performance

After fitting the logistic regression model to the biased training image data, it scored an accuracy of 70% on test images from a dataset with the same built-in bias. Two instances of a husky were predicted to be a wolf, and one instance of a wolf was predicted to be a husky.

Figure 7: Three images of the test set which were classified incorrectly by the logistic regression classifier after being fitted on the train set. From left to right, the logistic regression model classified the pictures as wolf, wolf and husky.

5.2 Explanations

Following the LIME technique, perturbation images were generated from which linear regression classifiers were trained. The six highest positive weights of each linear regression classifier were gathered, and the corresponding superpixels were colored green to form the explanations in Figure 8.

Figure 8: Explanations generated for predictions of the logistic regression classifier on the images in Figure 4. The pictures depict (a) a wolf, (b) a husky and (c) a husky in a snowy landscape. The logistic regression model predicted for these pictures respectively (a) wolf, (b) husky and (c) wolf. The green areas show the six superpixels with the highest weight after training the (linear regression) explanation model for each picture.

Explanation (a) and (c) contains some green areas in the lower half of the picture. All spots that were colored green were originally mostly white, with the exception of the leg of the wolf in explanation (a). Explanation (b) does not contain any green segments. Furthermore, we see that the logistic regression model correctly predicted the classes of the wolf and husky pictures, but was fooled by the picture of a husky in a snowy environment.

6 Analysis & Discussion

Whilst evaluating the results, it is important to understand some differences between the original experiment and our reproduction. Additionally, some issues were found during testing which puts on further elements to discuss about the performance of the LIME model. These topics, including a conclusion to our research question and proposals for follow up research will be adressed in this section.

6.1 Implementation Differences

While trying to stay true to the original experiment from Ribeiro et al. [7], we found some changes were useful to implement in our environment. For the experiment from the paper, superpixels of the images that were to be explained were found using the QuickShift algorithm [11]. In our testing,

we found this algorithm to produce too many segments, with most segments consisting of only a couple of pixels. As we wanted to minimize the complexity of the explanation model, the amount of superpixels found needed to be limited. We also did not want too few superpixels, as each segment would still need to be indicative of a specific informative part of the picture. For example the black part of the nose of a wolf should be contained in one segment. The SLIC algorithm provided functionality to specify the amount of segments we wanted as well as segment based on color, which was sufficient for the purposes of our reproduction.

Another part our implementation differs is that the logistic regression model is trained on a different source. Ribeiro et al. used a logistic regression model which was fitted to the features of a pre-trained neural network model, the *Inception V3* network [9]. Features of each image were extracted by putting them in the network and extracting values of the first soft-max layer. For our reproduction we wanted a controlled environment for both the images and the model used. We opted to fit the logistic regression classifier to less complex data, the rgb color values of all pixels of the image.

Lastly, no information on some specifics of the original experiment were given in the paper describing the experiment [7]. This meant that we did not have access to the images used to train the logistic regression classifier of the original experiment. Moreover, no specific parameter values were given for fitting the logistic regression and linear regression model, and no threshold value was given for which weight-values of the linear regression model should be included in the explanations.

6.2 Explanation Analysis

By performing this reproduction, we wanted to generate explanations which would portray a similar bias as can be seen in Figure 3. Particularly we wanted the explanations to show that the logistic regression classifier had a bias towards white areas on the lower half of the picture. As we can see in Figure 8, explanation (a) was most in line with this expectation. Clearly, snowy areas on the lower half of the picture are given a high weight value, meaning they had a high influence on the prediction of 'wolf' for the image. Notable is the brown leg of the wolf that was found to be significant for the explanation. Upon evaluating our image dataset, we found that most wolf pictures had a part of the wolf found in that similar spot. Since all wolf pictures were of brown wolves, there is a consistent brown spot across the wolf picture dataset at that part of the picture, which could explain why the explanation assigned a high weight to that area.

An unexpected issue with the LIME algorithm came to light with analysing explanation (b). No segments are marked green on the explanation. Upon evaluating the results more, it was uncovered that each segment of the image was given a weight value of zero, meaning no segment was eligible to be colored green in the explanation. Trying to generate explanations for other husky images from our dataset resulted in similar results. To explain why this happens, we need to look at our black box model. Recall that the logistic regression model will label each picture without white at the bottom as a husky. With the biased husky images, almost none contained white at the bottom. For the locally sampled perturbations of these images, parts of the image were turned gray, meaning no white parts were ever added to the picture. This meant that no perturbation generated contained white at the bottom of the picture causing the logistic regression model to label each perturbation as 'husky'. As a result, the linear regression model was trained with a dataset containing only one class, resulting in each superpixel being given the same weight. Specifically for explanation (b), we can see that not much white is found in the lower half of the picture. However, there is a white area in the center of the picture. Compare this area to explanation (a), where we notice that the logistic regression classifier expects to find brown in this area to predict 'wolf' for this image. As the perturbations also never add brown areas to the picture, we can see that the same phenomenon happens causing all weights to be given a weight of zero.

Lastly, we evaluate explanation (c). The context of this explanation, a husky in a snowy landscape, is exactly the same as the example from the original experiment (Figure 3), and thus we expect a similar result. However, explanation (c) depicts that the logistic regression classifier was also influenced by a white segment in the upper left area of the image. This disparity was also caused by a flaw with our image dataset. As we wanted to choose pictures of wolves with white in the lower area of the image, our 'wolf' dataset contained some pictures of wolves in a snowy landscape which could be seen entirely around the wolf. This may have caused the white segment in the upper left area of the husky in snowy area image to be included in the highest weighted segments used to draw the explanation. Nevertheless, explanation (c) still portrays that the snowy areas in the picture had the most influence on 'wolf' being predicted.

These explanations also give some insight in why the logistic regression classifier predicted some images of the test data wrong, as seen in Figure 7. The picture of the wolf, which was predicted to be a husky, only has a small area of white at the bottom of the image. This would indicate that the

white area is not big enough for the classifier to predict 'wolf' for the image. Similarly, the two pictures of huskies which were predicted as 'wolf' might have big enough white areas at certain spots in the picture to fulfill the requirements of the 'wolf' prediction.

With all the explanations evaluated, it is important to keep in mind that the LIME algorithm generates *local* explanations. By viewing these explanations we found out certain nuances of our image dataset which influenced the logistic regression classifiers weight values. However, the algorithm is not intended to provide explanations for the thought process of the entire black box. These explanations only portray what the black box used in that specific instance to generate its prediction. As the original intention of proposing the LIME algorithm was giving insight in predictions made by black box models [7], the LIME algorithm did give proper insight in why this model is unsuitable for human-like husky/wolf classification, and thus usage of the algorithm was a success.

6.3 Complications with the LIME Technique

Besides our results showing the LIME algorithm can be used to reflect the bias the logistic regression classifier was fitted towards, some evidence of the reported instability of the algorithm [8] were also found during implementation and testing. Volatility of the results became evident after running the algorithm multiple times with different random seeds and obtaining slightly different explanations. Figure 9.I shows some examples of this phenomenon. As the amount of superpixels chosen to be turned off are determined randomly for each sampled perturbation, the dataset for the linear explanation model should change a bit with a different random seed. Upping the amount of locally sampled perturbation might lessen this effect, although it would be difficult to determine how many perturbations is enough. We do see that the area that is colored green stays somewhat consistent between the experiment runs, so we can still generally deduce the same bias from each explanation, dampening the disturbance of this randomness. However, with more nuanced biases in a dataset, this might become more of a problem. We can also see that the amount of perturbations generated affects the resulting explanations which can be attributed to the dataset of the linear regression model having fewer training data with less perturbations generated, and as such has less information to fit to. This additionaly could be solved with generating more perturbations, however that also comes with its problems, as each extra perturbation generated means one more call to the black box for getting the label used to

train the explainer model. Making predictions from a black box model can be quite computationally expensive, so ideally these prediction calls should be limited. It could be suggested that more perturbations means the explanation model will overfit to the perturbations, however we would argue that this only comes to benefit the accuracy of the explanation. As a proper XAI model should show what a black box model used for its prediction [6], we want an explainer to be trained on as many combinations of the interpretable segments as possible to view the most important ones. The explainer model is only used to explain one instance, and as such the error of that explainer should be minimized as much as possible.

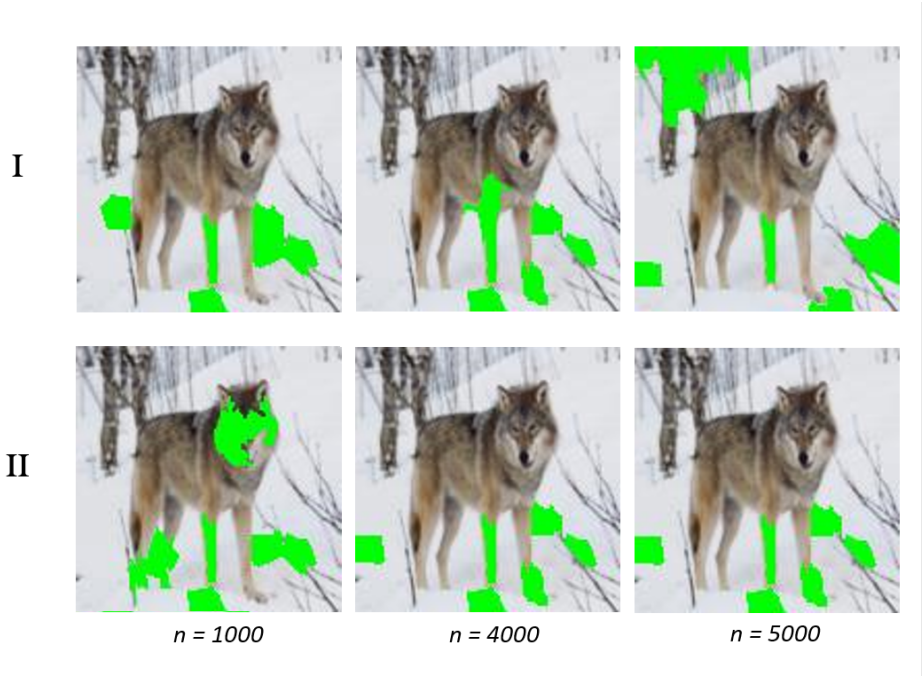

Figure 9: (I): Different explanations generated after running the experiment with the same parameters (same amount of locally sampled perturbations, same area from which perturbations are sampled) and different random seed. (II): Different explanations generated after running the experiment with a differing number of generated perturbations (from left to right respectively 1000, 4000 and 5000) using the same randomness seed.

Another drawback of the LIME algorithm also came to light during the implementation of the reproduction. In the experiment the bias of the im-

age dataset is based on color. However, the deactivated superpixels of the perturbations also need to have a color assigned to them, as the logistic regression model only accepts inputs equivalent in size to the original image. This means that the color chosen for the turned off superpixels has a large effect on the labels given to the perturbations as they are predicted by the black box model, and thus influence the generated explanation. To verify this, we ran the experiment for picture (a) of Figure 4 again and changed the perturbations to show *off* segments with a black color instead of gray. As can be seen in Figure 10, the change in color caused the weights given to the segments to be altered slightly, which led to a different part of the image having the heighest weights. In this context, the bias of the image dataset can still be concluded from both explanations, however with more complex biases this might have a significant impact on the generated explanation.

This phenomenon is mainly the result of choosing colored superpixels for the interpretable representation of the input. Changing the interpretable representation might be beneficial to prevent this, however, it is quite difficult to find a better suited interpretable representation as the superpixel representation is very clear to visualize.

trained with gray perturbations *trained with black perturbations*

Figure 10: Different explanations created by changing the color of the 'off' segments in the perturbation training images to gray and black respectively.

6.4 Conclusion and Further Studies

Before implementing the algorithm, we hypothesized that the LIME algorithm will generate explanations which give similar clarity to the bias of the image dataset as explanations generated in the original experiment done by Ribeiro et al. [7]. Based on our analysis on the generated explanations

and our observations testing the LIME algorithm, we have arrived at the following conclusions. The results of our reproduction reflect the results established by the original experiment. Two of the three explanations we generated properly visualized that the logistic regression classifier had a bias towards snowy areas for generating its predictions. Especially comparing the explanation example from the original experiment (Figure 3) to the explanation we generated within a similar context (Figure 8.c) shows that the explanations depict a similar bias for the same context. Furthermore, for every explanation we generated, we can conclude that the logistic regression classifier did not make its prediction similar as a human would. Thus the LIME algorithm did help give insight that the predictions generated by the logistic regression classifier were not trustworthy for differentiating huskies from wolves.

Even though the LIME algorithm did help enlighten the bias of the logistic regression model in the context of the experiment, our observations indicate that the algorithm is not suitable to discover more nuanced biases of black box models or declare a model unquestionably trustworthy. We confirmed the instability of the algorithm as mentioned in the literature [8] and showed that changing parameters of the algorithm could influence the resulting explanations. In the grander scheme of Artificial Intelligence LIME is a great step in making machine learning models more approachable for the layman, but its lack of consistency makes it an improper fit for any use cases with serious consequences from the explanations.

Possibilities for further studies would be to investigate if the optimal amount of perturbations necessary to eliminate the randomness found in this study can be theoretically calculated. There is already a technique suggested called *focused sampling* [8], and it would be interesting to study if its application would lower the randomness of our LIME implementation. This could also be combined with more research on the distance metric used for sampling perturbations. In our implementation we do not limit the distance at which perturbations can be generated, the intuition of which can be seen in Figure 1. Some experimentation can be done on sampling perturbation sets with a differing maximum allowed distance to the original image, and visualizing if a significant difference can be seen between each perturbation set.

References

[1] Julia Angwin, Jeff Larson, Lauren Kirchner, and Surya Mattu. Machine Bias. ProPublica, 2016. `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing` [Accessed 2-June-2023].

[2] Damien Garreau and Ulrike Luxburg. Explaining the explainer: A first theoretical analysis of LIME. In *International Conference on Artificial Intelligence and Statistics*, pages 1287–1296. PMLR, 2020.

[3] Damien Garreau and Dina Mardaoui. What does LIME really see in images? In *International conference on machine learning*, pages 3620–3629. PMLR, 2021.

[4] Dina Mardaoui and Damien Garreau. An analysis of LIME for text data. In *International Conference on Artificial Intelligence and Statistics*, pages 3493–3501. PMLR, 2021.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] P Jonathon Phillips, Carina A Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. Four principles of explainable artificial intelligence. *Gaithersburg, Maryland*, page 18, 2020.

[7] 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*, pages 1135–1144, 2016.

[8] Dylan Slack, Anna Hilgard, Sameer Singh, and Himabindu Lakkaraju. Reliable post hoc explanations: Modeling uncertainty in explainability. *Advances in neural information processing systems*, 34:9391–9404, 2021.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

[10] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[11] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10*, pages 705–718. Springer, 2008.