Crowd-Sourced Evaluation of Explainable AI Techniques with Games

Mayank Jain

CMU-CS-21-146

December 2021

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Adam Perer, Chair
Kenneth J. Holstein

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Keywords: XAI, Machine Learning, LIME, Grad-CAM, GWAP

For Libby 

Abstract

Image Classification is a fairly mature domain in Machine Learning (ML) today. All the way from the automobile industry to retail supply chains, image recognition and classification enable industry processes everywhere. The one big drawback when it comes to using ML in a lot of industries is the black-box nature of ML algorithms. Historically, it's been almost impossible to figure out *why* a neural net classifies a particular image as something.

On the other hand, Explainable AI (XAI) is an emerging domain in ML that aims to give people more insight into why an ML algorithm does something particular. This allows for more transparency into AI-made decisions, in turn allowing them to enter industries like healthcare and criminal justice, where a black box with 99% accuracy is just not enough. In recent times, a lot of XAI techniques have been proposed to help explain the image classification problem in specific, but few have been evaluated beyond anecdotal evidence. It usually just comes down to the authors saying that the explanations "look good". Many of these XAI techniques are designed for people with the intuition of a data scientist or ML engineer, with very few ways to evaluate them for non-experts.

In this work, we present a novel method for human evaluation of XAI techniques. We do this via a Game With a Purpose (GWAP) called *Eye into AI* that will allow researchers to crowd-source human evaluations of XAI techniques focussed on explaining deep learning models trained for image classification. In addition, we use this game to evaluate LIME, Grad-CAM, and Feature Visualizations, the first evaluation of its kind. We find that our game is able to provide a clear ranking of these XAI techniques, and provide meaningful insights into the kind of use cases they would each be most useful in.

Acknowledgments

I would like to thank my advisor Dr. Adam Perer for guiding me all along the project, and helping me with both the practical and theoretical aspects of this project.

Contents

List of Figures

Chapter 1

Introduction

Machine Learning, in recent times, has seen some fascinating innovations. The design and composition of neural network architectures is ever-changing, achieving state-of-the-art accuracy for numerous tasks. In particular, deep neural nets (DNNs) have been performing remarkably well in a lot of real-world applications, ranging from natural language processing [7] [18], to image classification [6] [23] [21].

DNNs have gotten exceptionally good at extracting features from large high dimensional data, making them highly desirable in industry. However, a big drawback that has emerged in recent times is that as DNNs get more complex, they function more and more as a high-level black box. It is almost impossible to derive relationships between their inputs and outputs. In low-risk environments, like some sort of recommendation system, this is not a big deal since the impact of making an error is not very high. However, in industries like healthcare and the judicial system, where predictions can be life altering for someone, the impact of a single error is much higher. This is a big reason why adoption of ML in these industries has been extremely slow, there is a demand for explainability - a meaningful way to understand why the model is making the prediction it is.

In addition, even in non mission-critical industries, trust is very important for fast and widespread adoption of intelligent systems. Previous work has shown that meaningful explanations can greatly increase trust in intelligent systems [16] [19].

To satisfy this recent demand for XAI (Explainable Artificial Intelligence), a lot of techniques have emerged that aim to provide explanations and interpretations for black-box predictions made by complex DNNs. However, while a variety of XAI techniques have been suggested, there is no standardized way to evaluate them. It is surprisingly hard to answer the question, ”Given a list of XAI techniques, which will provide the best explanations”, even when the dataset and the model in question are fixed. While there is an emerging body outlining new XAI techniques and ways to implement them, there have been far fewer contributions focused on evaluating these techniques (less than 5% of the total body of work, as of 2018) [1].

In addition, of the evaluation techniques that have been suggested for XAI techniques in the

past, a lot of them have taken an automated approach at evaluating these techniques [15] [20] [8]. However, recent work has shown that an explanation classified as highly interpretable by a machine might not necessarily be interpretable at all by humans [14]. This motivates the need for human-based XAI evaluation techniques.

In this work, we propose a scalable, easy to use way, to crowd- source quality metrics about an XAI technique, specifically for image classification problems, allowing XAI techniques to be evaluated on an even playing field with real humans and data. We do this by proposing a "Game with a Purpose" (GWAP) called *Eye into AI*. The goal of these games is to make boring tasks more interesting, allowing easier data generation [29] [30].

We build off of an initial version of this game created in a previous body of work [5]. Our focus here is to alter the game to allow it to work with more XAI techniques in addition to making it more scalable and easier to run trials with. In addition, we also use this game to evaluate two state-of-the-art XAI techniques, LIME [19] and Grad-CAM [22], in order to meaningfully compare the two. We believe this to be the first such comparison between LIME and Grad-CAM.

Chapter 2

Background and Related Work

2.1 Image Classifiers

Image classification, in the last decade, has evolved at a dramatic pace. With the inception of convolutional neural networks [12], and the creation of datasets like ImageNet [4], ML image classification algorithms are performing extremely well right now.

From the first LeNet architecture in 1998, there have been several milestones in CNN development for Image Classification over the last decade, like AlexNet [11], VGGNet [24], GoogleNet [26], and ResNet [6].

For the purposes of our research, we chose GoogleNet for all classification predictions. While GoogleNet is not the absolute state-of-the-art anymore, it still performs quite well, with an error of just 6.7% on ImageNet. The good performance, coupled with the relative ease of running our explainibility techniques on GoogleNet, compared to something like ResNet, was our reasoning behind using GoogleNet. Any compatible architecture could have been used instead.

2.2 Explainibility Techniques

In this section, we've provided brief introductions to the XAI techniques that have been used in our evaluation of *Eye into AI*.

2.2.1 LIME

Local Interpretable Model-Agnostic Explanations (LIME) [19] is a technique that approximates any black box machine learning model with a local, interpretable model to explain each individual prediction.

On a high level, LIME perturbs the original input data points for a model, feeds them back into the black-box model, and then observes the corresponding outputs. LIME then weighs those new data points as a function of their proximity to the original point. Ultimately, it fits a surrogate

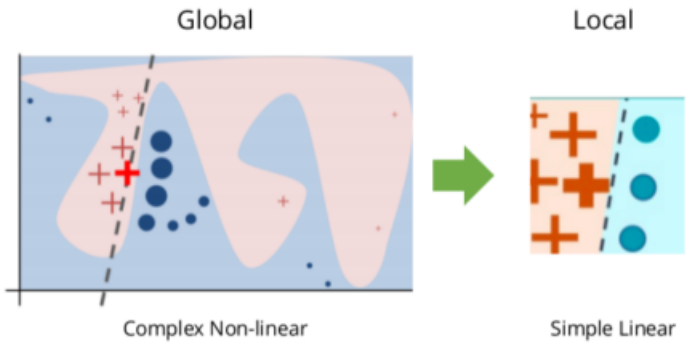


Figure 2.1: An example of how LIME fits a surrogate model on the dataset.
(Image from Ribeiro et al. [19])

model such as linear regression (see figure 2.1) on the dataset with variations using those sample weights. Each original data point can then be explained with the newly trained explanation model.

LIME outputs, for the image classification task, is in the form of overlays on the the input image, highlighting positive and negative superpixels of the image (see figure 3.4d).

2.2.2 GradCam

Grad-CAM [22], like most saliency map XAI techniques, attribute the output of a DNN to parts of its input.

Unlike the simpler Vanilla Gradient [25], in Grad-CAM, the gradient is not backpropagated all the way back to the image, but instead to the last convolutional layer, in order to create a coarse localization map that highlights the important regions of the image.

Unlike traditional gradient based approached, Grad-CAM relies on both, the gradients and the feature maps of the convolutional layer, resulting in a less ”edge-driven” explanation as opposed to other saliency based approaches. We recieve the output of Grad-CAM in the form of a heatmap overlayed on the input image, highlighting the important areas of the image (see figure 3.4b).

2.2.3 Feature Visualizations

Feature visualizations [17] aim to make a neural net’s interpretation of images visible. CNNs are able to learn abstract features and concepts from raw image pixels, radically reducing the need for feature engineering.

Making these learned features explicit is called feature visualizations. For CNNS, this usually

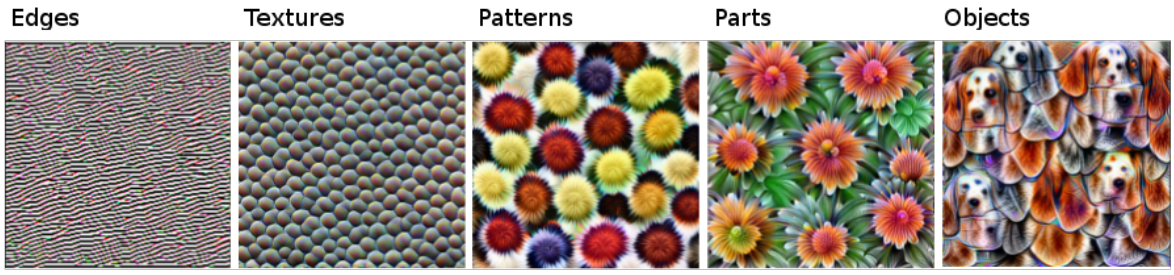


Figure 2.2: Visualizations of features learned by a CNN (Inception V1)
(Image from Olah et al. [17])

involves visualising an entire layer, since a CNN usually contains millions of neurons and visualizing each neuron would quickly become futile.

Figure 2.2 illustrates feautres learned by a CNN on ImageNet Data [4]. We can see that the features range from simpler features in the lower convolutional layers (left) to more abstract features in the higher convolutional layers (right).

2.3 Human-based evaluation of XAI techniques

A variety of crowd-sourced techniques for evaluating XAI techniques have been suggested in the past. Jeyakumar et al. [10] asked participants to directly choose between explanations generated by XAI techniques for a variety of tasks across domains. Hutton et al. [9] took a different approach, asking user to compare human- and computer-generated explanations and indicate which they preferred and why.

Few past works have taken a human computation game focused approach for crowd-sourced evaluation of XAI. One past body of work that did take this approach focused solely on saliency maps [13] and found encouraging results.

Chapter 3

The Setup

Our research involved two major components. The first involved designing the actual *Eye into AI* game to crowd source data on the quality of these XAI techniques, and the second involved creating the actual explanations for our dataset using some state-of-the-art XAI techniques.

3.1 EyeIntoAI - a tool to crowd-source explanation quality

Before diving into the explanation generation process, let's do a quick overview of the actual game itself.

3.1.1 The Game - Overview

The game was built in Javascript, using the React library. We built off of an earlier iteration of the game that was also built here at CMU [5]. The current code can be found here.

The game has 2 major rounds for each XAI technique being evaluated.

- **Explainer Round:** In this round, the "explainer" selects the top explanations, in a ranked manner, that they believe will allow someone to guess the entire image as quickly as possible. This can be seen in 3.1. Note that we only ask the user to pick the top 4. We found that ranking all the explanations shown can get cumbersome since they all become equally non-useful after a point.

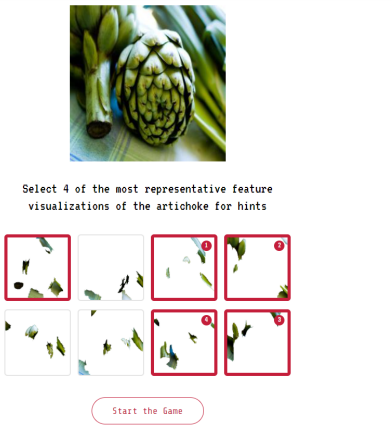
  The explanations being shown are picked randomly from the top 5 and the bottom 5 explanations, as ranked by the XAI technique in order of importance for the image prediction.

  The goal is to be able to identify if a human's ranking of importance of the explanations for the image prediction correlate with that of the XAI technique.

- **Guessing Round:** In the guessing round, the user receives one visual explanation to start with. A new explanation is revealed every 15 seconds (with a total of 4 explanations revealed in total), and is super imposed on the older explanations, so that the user can easily contextualize the newly revealed visual information. This can be seen in 3.2

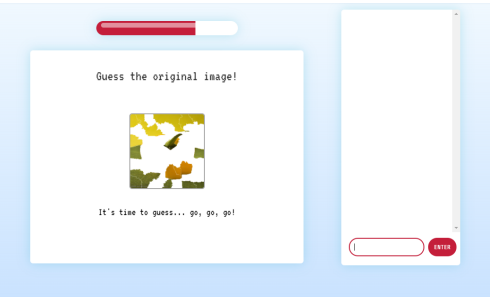
  Note that in this round, the explanations are always revealed in the order that the XAI technique ranked them. Specifically, the user starts of with the most important explanation

(a) Explainer picks a category for the explainer round.

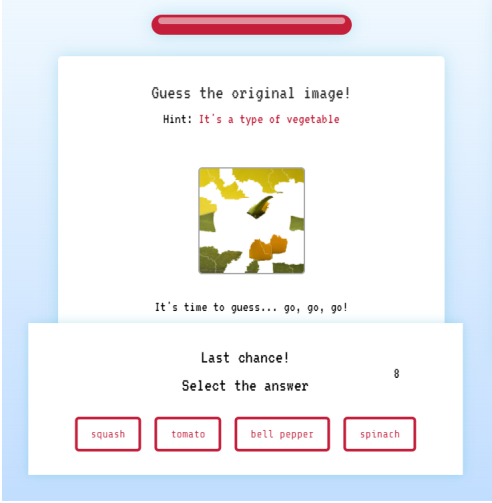


(b) Explainer selects the top explanations, in a ranked manner

Figure 3.1: Explainer Round



(a) Guesser can see explanations being revealed every 15 seconds. They type their guesses in the chatbox on the right.



(b) If the guesser is unable to guess the image after the 4 explanations are revealed, they receive a text hint.

Figure 3.2: Guessing Round

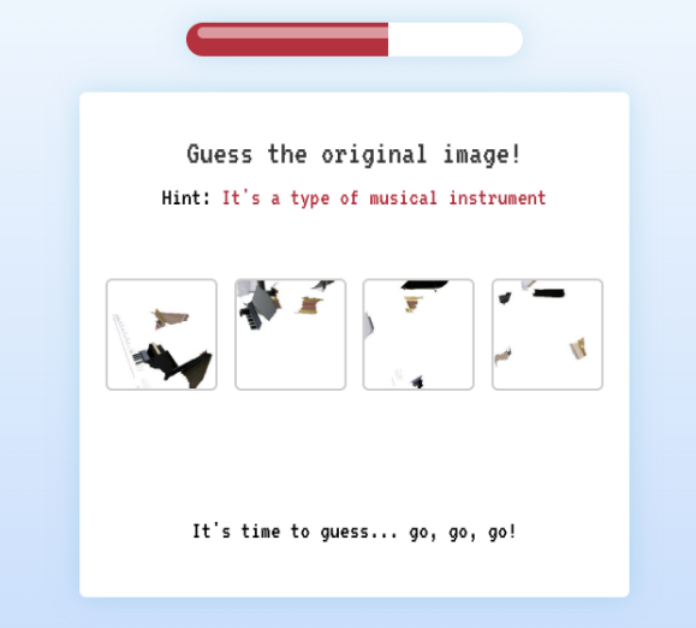


Figure 3.3: Old Guessing Round

as deemed by the XAI technique, i.e `top_0` . 15 seconds in, `top_1` is also revealed, 15 more seconds and `top_2` is revealed, and finally `top_3` is revealed. Thus for each XAI technique, the user has the opportunity to see the top 4 most important explanations.

Both these rounds happen for all the XAI techniques that we're testing, namely - LIME, GradCam, and FeatureViz. We also run the same rounds for a baseline that is outlined in more detail in 3.3. The order in which these techniques are shown to the User is randomized, to prevent any sort of biasing.

Some quick background: In the guessing round, we were initially revealing these images separately, and out of order, as opposed to in a ranked fashion (see figure 3.3). This was an artifact from the original version of this game [5]. However, after a few trials of that, we realized that there 2 major issues with that approach -

- We observed that showing explanations out of order didn't let us compare quality of the techniques themselves, since we would see people usually always getting it right once `top_0` was revealed, irrespective of what was shown earlier.

  In addition, the random order didn't let us compare how effective the top X% of one technique's output was, compared to another technique.

- In addition, showing explanation side-by-side, as opposed to superimposed just demanded extra cognitive effort from the participants for something that we were not testing.

  Participants reported having to just superimpose the different explanations in their head which requires effort and is not always perfect, meaning that we could arbitrarily lose information from explanations depending on the participants' ability to put the explanations together in their head.

3.1.2 Data Collection

The game generates data that can be used to evaluate the quality of the XAI techniques in 2 major ways -

- **Explainer Round:** In this round, we log the explanations chosen by the explainer, as well as their ranking according to the XAI technique being evaluated. This allows us to look at correlations between what parts of the image a human deems important vs what the XAI technique thought was important.
- **Guessing Round:** In this round, we log all the guesses made by the user, the time taken for each guess, as well the number of explanations available to the user at the time of the guess. In addition, we also log whether the user needed the final textual hint.

  Using this information, we're able to draw conclusions about the efficacy of the XAI technique, when it comes to isolating the most important parts of the image. One example analysis is to look at the number of guesses and the number of explanations a user needed to finally guess the original image.

We show how this data can be used to evaluate current state-of-the-art XAI techniques in Section 4 as a reference for anyone looking to crowd source quality metrics about other XAI techniques in the future.

3.2 Experimental Parameters

Dataset: We selected 50 images as our dataset. The main criteria for our selection was that ImageNet [4] had a clear unambiguous class for the image, and the image featured only one object. Our dataset can be viewed here.

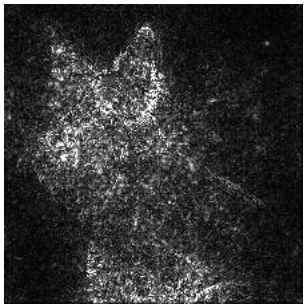
Model: We used GoogLeNet [27], pretrained on the ImageNet dataset [4], as our ML model for all the XAI techniques. This architecture was able to correctly classify all the images in our dataset. There is no reason to choose GoogleNet over any other architecture. Any architecture that would support both LIME and Grad-CAM could be used.

3.3 Explanation Generation Process

For the explanation generation process, we tried out a variety of different XAI techniques. In specific, the ones we ran on our data set were LIME, GradCam, SmoothGrad, and Guided Gradcam. In addition, we also ran the Marr Hildreth Edge Detection algorithm on our dataset. Sample outputs from all these techniques are shown in figure 3.4.
We envision all of these being pitted against each other through *Eye into AI* in the future. However, for the purposes of our research, we decided to use a subset of these, namely - LIME, and Gradcam, in our actual trials. We did this due to two main reasons -

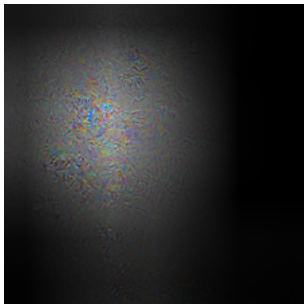
(a) Original Image

(b) Gradcam



(c) SmoothGrad

(d) LIME



(e) Guided Gradcam

(f) Edge Detection - Marr Hildreth

Figure 3.4: Various XAI Techniques tried out by us
All the techniques were run with the classification target as the ImageNet label 199 (Human Readable Label - 'Scotch terrier, Scottish terrier, Scottie')

- Game length was a concern. Gradcam and LIME, in addition to our baseline as well as a feature vizualization round (a legacy remnant we decided to keep) already made our GWAP quite long. We wanted to make sure our users were engaged to the very end.
- Ease of ranked image segmentation based on the XAI technique. While possible with every XAI technique, by definition, LIME and Gradcam were the easiest to work with for our purposes.

Our approach for explanation generation was to create explanations that reveal 10% of the image per explanation. The desired output was 4 ranked explanations (revealing, in total, 40% of the image), each revealing 10% in order of importance, as deemed by the XAI technique.

Outlined below are our explanation generation processes, in detail, for LIME, GradCAM, and our Baseline. In addition, we also mention our feature visualization explanations even though they were not created by us since they have been carried over from the original version of *Eye into AI* [5].

3.3.1 LIME

As mentioned in Section 2.2, LIME is model-agnostic, meaning that it can be applied to any machine learning model. The technique attempts to understand the model by perturbing the input of data samples and understanding how the predictions change.For the case of images, LIME generates perturbations by turning on and off some of the super-pixels in the image.

LIME supports a variety of different segmentation algorithms to create these superpixels. We used Quickshift - a relatively recent 2D image segmentation algorithm [28], based on an approximation of the kernelized mean-shift. [3]

LIME then uses weighted local surrogate model to fit these perturbations, ultimately giving us a list of superpixels, ranked in order of importance for the final prediction. (See figure 3.5)

For our dataset, we found that with LIME (using QuickShift), the average no. of generated superpixels was 50 with a standard deviation of 8.22. Thus, in order to reveal around 10% of the image in each explanation, we revealed 5 superpixels per explanation, starting from the top 5 (`top_0`) , then the next 5 (`top_1`), and so on. See figure 3.6 for a visual example.

3.3.2 Grad-CAM

As talked about in more detail in Section 2.2, Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept (say 'dog' in a classification network or a sequence of words in captioning network) flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. This localization map is presented as a heatmap marking the regions deemed important

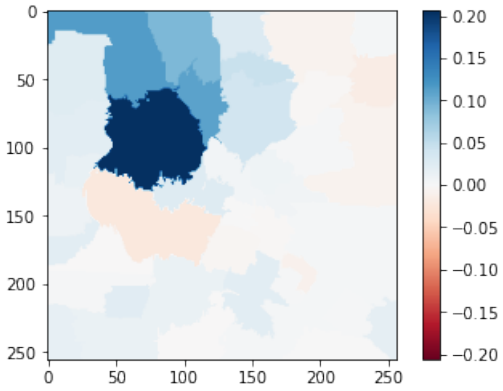


Figure 3.5: Superpixels for a Scotty Terrier (see figure 3.4a for the original image)
Blue is positive, and red is negative - one can observe that the head of the Scotty Terrier is the most blue, meaning it was the highest ranked superpixel by LIME

(a) `top_0`
($x < 10\%$)

(b) `top_1`
($10\% < x < 20\%$)

(c) `top_2`
($20\% < x < 30\%$)

(d) `top_3`
($30\% < x < 40\%$)

Figure 3.6: LIME Explanations for a Scotty Terrier
(x is percentage ranking from the top as determined by LIME)

(a) `top_0`
($x < 10\%$)

(b) `top_1`
($10\% < x < 20\%$)

(c) `top_2`
($20\% < x < 30\%$)

(d) `top_3`
($30\% < x < 40\%$)

Figure 3.7: Grad-CAM Explanations for a Scotty Terrier
(x is percentage ranking from the top as determined by Grad-CAM)

by the model (see figure 3.4b) [22].

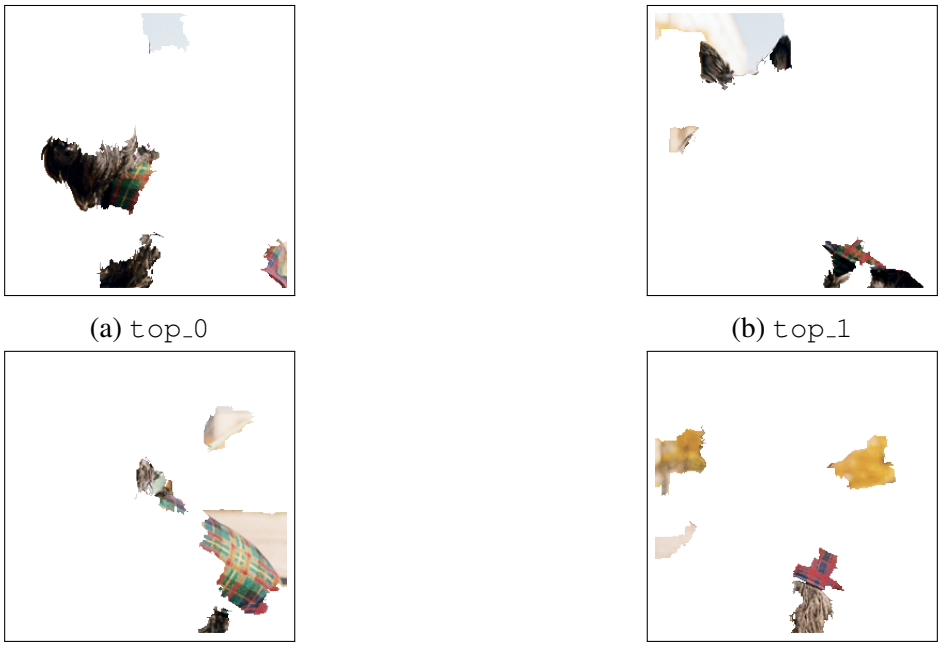
Note that in Grad-CAM, there is no segmentation into superpixels. So in order to create our explanations, we worked solely with the heatmap. The heatmap assigns a value to each pixel of the image, so in order to create the 4 explanations, we simply sorted the heatmap values and took the top 10%, then 10-20%, 20-30% and so on for each explanation. (see figure 3.7)

3.3.3 Baseline

As mentioned in subsection 3.1.1, we also generate baseline explanations for each image in our dataset that are also shown to a user in every game run. The goal of the baseline here is to get random pixels with no assigned importance to them, i.e to understand how much simply revealing random parts of the image helps in guessing the image.

For these baseline explanations, we use QuickShift, exactly like LIME, to first segment the image. As mentioned in 3.3.1, the number of superpixels for our dataset were approximately 50.

Thus, to get 10% of the image revealed per explanation, just like in the LIME setting, we used 5 superpixels per explanation. The difference, however, was that we chose those superpixels randomly, as opposed to using LIME's rankings (see figure 3.8 for a visual example).



(a) `top_0` (b) `top_1`

(c) `top_2` (d) `top_3`

Figure 3.8: Baseline Explanations for a Scotty Terrier
(For the baseline, there is no ranking, each explanation is just randomly chosen superpixels such that 10% of the image is revealed)

3.3.4 Feature Visualization

We didn't implement feature visualizations and generate them ourselves for this work, but we carried over the explanations used in the original version of the game, created using the Lucid library [1]. More details about the explanation generation process for feature visualizations can be found in the original paper on *Eye into AI* [5].

The goal was to be able to see how feature visualizations, a completely different style of explanations, stacked up against Grad-CAM, LIME, and our baseline. See figure 3.9 for a visual example of Feature Visualisations.

[1] https://github.com/tensorflow/lucid

(a) Original Image

(b) `top_0`

(c) `top_1`

(d) `top_2`

(e) `top_3`

Figure 3.9: Feature Visualisations
(`top_0` is the most positive channel, `top_1` is the next most positive channel, and so on)

Chapter 4

Evaluation

4.1 Evaluation approach + process

For our evaluation, we focused our trials and data collection on trying to understand if we could use *Eye into AI* as a viable tool for crowd-sourcing quality metrics about XAI techniques and pitting techniques against each other to meaningfully compare completely different techniques.

In total, we got 71 participants to play our game as a part of their classes at CMU.

Note: For these trials, we also had an additional round in our game, along with the LIME, Grad-CAM, and Baseline rounds that were mentioned in section 3.1. This is a remnant of the last iteration of the game [5] and was left in there for legacy purposes, and as an additional analysis vector.

4.2 Insights - Crowd Sourcing Viability

We've laid out our insights from our trials below. Our current analysis scripts can be found here. To understand if *Eye into AI* is a viable tool for crowd-sourcing quality metrics about XAI techniques, we wanted to see if we could get some meaningful information about how effective LIME and Grad-CAM were, in their ability to identify which parts of the image were the most important for the model.

We derived these insights from the two major parts of the game - the guessing round, and the explainer round.

4.2.1 Guessing Round

From the guessing round, we're able to isolate the efficacy of the top 40% of the image (in 10% increments), as ranked by an XAI technique, in being able to identify the subject of the entire image.

4.2.1.1 Comparison Insights:

By looking at the performance of users across the entire dataset for different techniques, in comparison to the baseline, we can meaningfully derive insights about whether they add any value at all, and also about which ones potentially add more value.

Looking at figure 4.1, we can observe some interesting trends.

Feature Visualizations (featureViz) : It's clear that feature visualizations are unable to compete with Grad-CAM and LIME at all. Even our baseline, by virtue of simply revealing random parts of the image, performs much better than feautureViz.

We concede that the argument can be made that comparing featureViz, an abstract visualization, to XAI techniques that actually reveal part of the original image is not a fair comparison. However, even looking at featureViz in isolation, we see poor performance, with a correct guess percentage of just 9%. This is quite interesting as it highlights the fact that a well known and well regarded XAI technique might actually not be useful to humans at all.

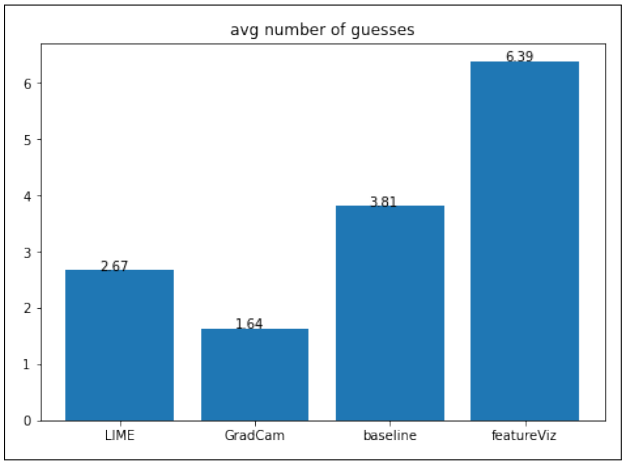
Baseline: As one would expect, our random baseline performs worse than both LIME and Grad-CAM across the board. Avg time for the right guess, as well as the avg number of guesses taken by users are substantially higher for the baseline. In addition, the correct guess percentage is also lower than both LIME, and Grad-CAM. This tells us that both LIME and Grad-CAM are indeed helpful to some degree.

Grad-CAM and LIME: Looking at the performances of Grad-CAM and LIME. We can see that Grad-CAM has markedly better performance than LIME. Users, on average, take $\tilde{3}$ more seconds to arrive at the right guess, take $\tilde{1}$ more guess (i.e, need to see 10% more of the image), and only get the right answer 80% of the times, as opposed to a 100% in the case of Grad-CAM.

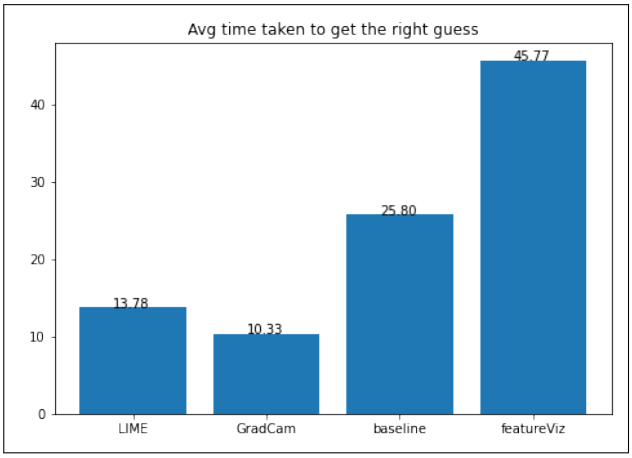
This last statistic is arguably the most important, as it signifies that 20% of the times, the 40% of the image isolated by LIME is just not enough for the user to identify the image. In mission critical scenarios that require explainibility, a few extra seconds to correctly identify the image might not matter too much but failure to identify the image (or in other words, explain the model correctly) can be catastrophic.

Finally, as a sanity check, we find that our result here is consistent with previous work done in a similar vein. Though we're not aware of any work that tried comparing a saliency map technique with a surrogate model approach like LIME, previous work done on solely saliency based approaches has seen Grad-CAM outperform other methods [13] [2].

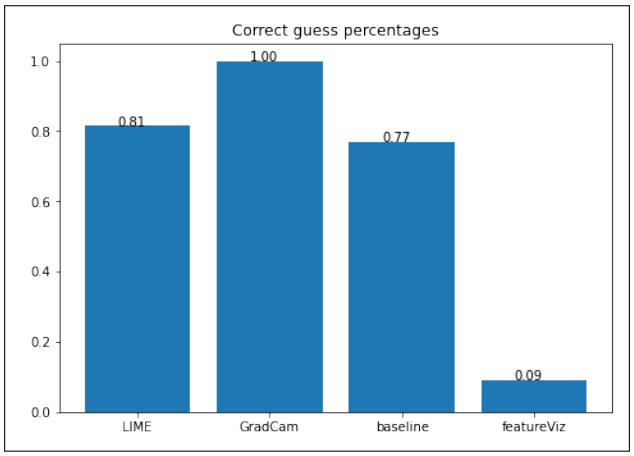
One reason why Grad-CAM performs so well with humans might be the fact that Grad-CAM produces low-resolution feature maps that are then linearly interpolated to fit the resolution of the original image, resulting in mostly connected regions as opposed to more distributed regions.



(a) Average Number of guesses to get the right answer



(b) Average time taken for the right guess



(c) Percentage of trials in which the user was able to guess correctly

Figure 4.1: Guessing Round - Comparision Insights

4.2.1.2 Isolated Image Insights:

In addition, data from the guessing round also allows us to isolate images that a particular technique might have been particularly bad or good at explaining, thereby giving us insight into wrong and right use-cases for the technique.

For instance, for an image of a "cucumber" in our dataset, using *Eye into AI*, we were able to realize that LIME did much worse than Grad-CAM when it came to isolating the best parts of the image. Only 25% of the users got cucumber right when shown the LIME explanations, but a 100% of them got it right when shown the grad-CAM explanations.

Looking at the actual explanations themselves, we can see that for an image style like this one, Grad-CAM's explanation style (of a heat-map centered at the most important part of the image) seems better suited. We believe this might be because the more connected nature of Grad-CAM explanations does a better job at allowing humans to contextualize the image and arrive at the right answer.

4.2.2 Explainer Round

Looking at the data from the explainer round, our insights from the guessing round are backed up. As we've talked about in 3.1.1, in the explainer round, users pick the top 4 out of the 8 shown explanations as their top 4 (see figure 3.2a). We compared the 4 images picked by humans with the rankings attributed to them by the XAI techniques.

Our hypothesis was that a high correlation between what the humans thought was important and what the XAI technique thought was important would imply high interpretability for the technique.

To start off, looking at figure 4.5, we can see that for featureViz, the rank of the explanation according to the XAI technique had almost no bearing on the amount of times the explanation was picked, giving us an *almost* uniform distribution. This hints towards poor human interpretability of feature visualizations.

For Grad-CAM, we can see that the users' top 4 ranking and the technique's top 4 ranking are quite similar. For LIME, however, while a large percentage of people did pick from `top_0` to `top_3` as their top 4, in 32% of the trials users also picked an image from `bottom_0` to `bottom_3`. In comparison, this only happened in 8% of the trials in Grad-CAM.

We believe that this is again due to the difference in the style of explanations between LIME and Grad-CAM. Since LIME employs intelligent segmentation, even the explanations deemed the least important by LIME can have some level of coherency making it more likely for it to be found important by a cognitive human. Grad-CAM explanations, on the other hand, have a ring like structure due to the heatmap which makes the less important explanations almost entirely useless for a human.

(a) Top 40% according to LIME

(b) Top 40% according to Grad-CAM

Figure 4.2: Isolated Image Insights

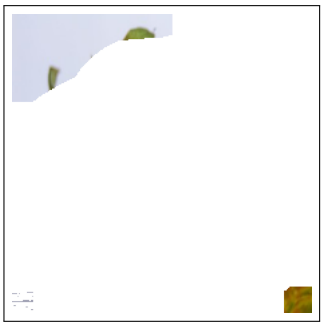
(a) Original Image

Figure 4.3: Isolated Image Insights

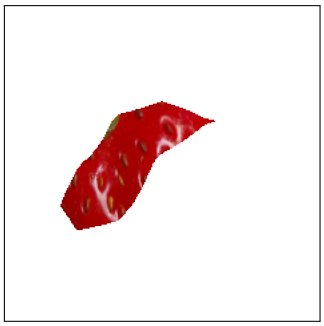
Taking a specific example from our dataset, we looked at the explanations for an image of a strawberry. In the case of LIME, users chose one of the bottom 4 explanations as their top four 16 times. On the other hand, in the case of Grad-CAM, users chose one of the bottom 4 explanations only once. Looking at the top most explanation and the bottom most explanation in figure 4.4, we can see why this 16x difference might exist.

In the case of LIME, one could argue that the bottom most explanation gives just as much, if not more, useful information for classification than the top most image. In the case of grad-cam however, the top most explanation is unequivocally more useful than the bottom most explanation because the bottom explanation only contains the background part of the image.

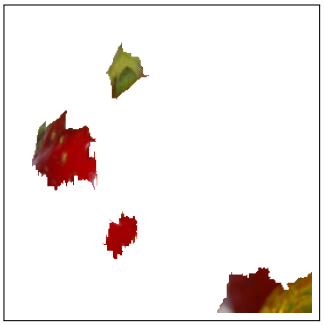
(a) Original Image



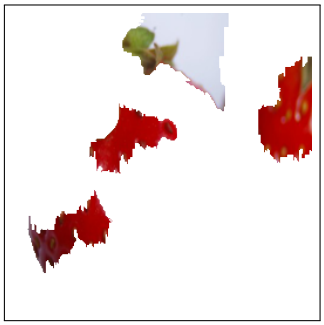
(b) Gradcam - `bottom_0`



(c) Gradcam - `top_0`



(d) LIME - `bottom_0`



(e) LIME - `top_0`

Figure 4.4: LIME explanations VS Grad-CAM explanations - Strawberry

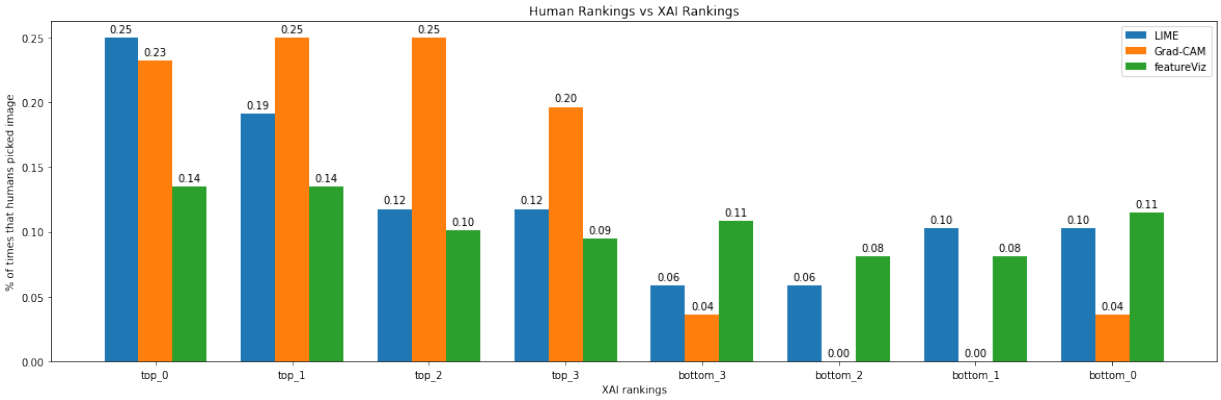


Figure 4.5: Human Rankings vs XAI Rankings

Chapter 5

Conclusion

In this work, we investigated a novel approach to conducting human evaluation of XAI techniques with a GWAP. In addition, we used our proposed approach to conduct an analysis and comparative evaluation of LIME, Grad-CAM, and Feature Visualization. We conducted a total of 71 trials across 2 different CMU classes.

Our results showed a clear ranking of the XAI techniques, with Grad-CAM performing the best, LIME coming second, and Feature Visualizations performing the worst. In addition, our analysis also shed some light on how the explanation styles of Grad-CAM, and LIME affect the use cases where they peform the best.

Our results also pointed at Grad-CAM's higher interpretability than LIME and Feature Visualizations for non-experts. Our users were able to pick out the explanations deemed by Grad-CAM to be the most important much more easily than other XAI techniques.

Going forward, it would be very interesting to see *Eye into AI* applied to more XAI techniques. In addition, a lot of future work can also be done in evaluating re-playability, and ease of use of this game, and in improving those aspects of the game so that users truly enjoying playing it and providing data. During our trials, we received some very good suggestions, such as introducing better scoring, and a way for users to see how well they're doing in the game, as a way to increase user engagement and re-playability. It would be very interesting to see these features implemented and analyse improvements in engagement.

Bibliography

[1] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052. 1

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps, 2020. 4.2.1.1

[3] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 3.3.1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848. 2.1, 2.2.3, 3.2, 3.2

[5] Laura Beth Fulton, Ja Young Lee, Qian Wang, Zhendong Yuan, Jessica Hammer, and Adam Perer. Getting playful with explainable ai: Games with a purpose to improve human understanding of ai. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2020. 1, 3.1.1, 3.1.1, 3.3, 3.3.4, 4.1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1, 2.1

[7] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015. 1

[8] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019. 1

[9] Amanda Hutton, Alexander Liu, and Cheryl Martin. Crowdsourcing evaluations of classifier interpretability. In *2012 AAAI Spring Symposium Series*, 2012. 2.3

[10] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 2020. 2.3

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105, 2012. 2.1

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. 2.1

[13] Xiaotian Lu, Arseny Tolmachev, Tatsuya Yamamoto, Koh Takeuchi, Seiji Okajima, Tomoyoshi Takebayashi, Koji Maruhashi, and Hisashi Kashima. Crowdsourcing evaluation of saliency-based xai methods, 2021. 2.3, 4.2.1.1

[14] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation, 2018. 1

[15] Thu Trang Nguyen, Thach Le Nguyen, and Georgiana Ifrim. A model-agnostic approach to quantifying the informativeness of explanation methods for time series classification. In *International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 77–94. Springer, 2020. 1

[16] Mahsan Nourani, Samia Kabir, Sina Mohseni, and Eric D Ragan. The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 97–105, 2019. 1

[17] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017. 2.2.3, 2.2

[18] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26, 2020. 1

[19] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. (document), 1, 2.2.1, 2.1

[20] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016. 1

[21] Lars Schmarje, Monty Santarossa, Simon-Martin Schröder, and Reinhard Koch. A survey on semi-, self- and unsupervised techniques in image classification. *CoRR*, abs/2002.08721, 2020. URL `https://arxiv.org/abs/2002.08721`. 1

[22] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi: 10.1109/ICCV.2017.74. 1, 2.2.2, 3.3.2

[23] Subarna Shakya. Analysis of artificial intelligence based image classification techniques. *Journal of Innovative Image Processing (JIIP)*, 2(01):44–54, 2020. 1

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 2.1

[25] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional net-

works: Visualising image classification models and saliency maps, 2014. 2.2.2

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 2.1

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL `http://arxiv.org/abs/1409.4842`. 3.2

[28] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *European conference on computer vision*, pages 705–718. Springer, 2008. 3.3.1

[29] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Commun. ACM*, 51 (8):58–67, aug 2008. ISSN 0001-0782. doi: 10.1145/1378704.1378719. URL `https://doi.org/10.1145/1378704.1378719`. 1

[30] Jichen Zhu, Antonios Liapis, Sebastian Risi, Rafael Bidarra, and G Michael Youngblood. Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2018. 1