Explain Any Concept : A review of the Concept-Based Explanation method

Nicolas Allègre *nicolas.allegre@telecom-paris.fr*
Mastère Specialisé IA
Télécom Paris

Louis Borreill *louis.borreill@telecom-paris.fr*
Mastère Specialisé IA
Télécom Paris

Merlin Poitou *merlin.poitou@telecom-paris.fr*
Mastère Specialisé IA
Télécom Paris

Abstract

Explain Any Concept (EAC) is an explainable AI (XAI) method proposed by Ao Sun & Wang (2023), which leverages the Segment Anything Model (SAM) to explain the outputs of image classification models. Producing explainable outputs from deep neural networks (DNNs) is a crucial challenge in machine learning. In computer vision, and particularly in image classification tasks, outputs are traditionally explained using pixel-based XAI methods that aim to highlight the most important pixels influencing the DNN's decision. EAC introduces a novel approach by using SAM to segment an image into concepts (i.e., high-level groups of pixels), and then training a model to predict which concept was most decisive in the classifier's prediction. The goal of this study is to reproduce the original results of EAC, compare them with the same baselines, and propose improvements to the method.

1 Introduction

Our work is inspired by challenges such as the ML Reproducibility Challenge (MLRC). Our goal is to understand, replicate, implement our own version, and compare it with the results of a machine learning research paper. Since our focus is on XAI, we chose to reproduce the work of Ao Sun & Wang (2023), namely the Explain Any Concept (EAC) method.

Explainable AI (XAI) is a vital branch of artificial intelligence focused on producing explanations for the outputs of deep learning models. By nature, such models are often considered "black boxes." Given the increasing use of AI in critical domains — such as medicine and justice — where decisions can have significant societal or human consequences, the need for transparent and trustworthy models is more pressing than ever.

EAC was developed in this context, addressing the explainability of image classification models. Image classification involves assigning an image to one of several predefined classes. EAC approaches this by using the Segment Anything Model (SAM) (Kirillov et al., 2023) to identify which part of the image contributed most to the classification decision. Semantic segmentation models like SAM assign a segmentation label to each pixel in an image, identifying meaningful substructures. While these models do not associate segmentation classes with specific labels, they are useful for isolating distinct regions such as objects, animals, or backgrounds within an image.

This paper will first present an overview of EAC's approach, along with our own implementation choices. We will then compare our results with those of the original implementation, as well as with another similar method (LIME) cited in the original paper.

2 Explain Any Concept : model explanation

EAC's approach can be divided into three main steps: generating segmentation masks, training the surrogate model, and using this model to predict the explanatory segmentation mask.

The target classification model, referred to as f_{target} in the following, is fixed in our implementation to a ResNet-50(He et al., 2015), like in the original paper implementation.

The surrogate model, denoted $f_{\text{surrogate}}$, is in our case a simple one-layer MLP followed by the same prediction layer as the target model f_{target}, which we denote as f_c. The purpose of this surrogate model is to replicate the behavior of f_{target} for a given input image. Further details will be discussed in the **Training** section.

2.1 Phase 1 : Segmentation

The first step of the EAC pipeline is to segment the image using SAM. We will not go into detail here about how SAM works. SAM generates a concept classification for all pixels in the image, which can be viewed as a form of pixel clustering. Based on our own experiments using the huge model, SAM typically produces around 30 distinct concept classes (sometimes a lot more), depending on the image's complexity and the parameters used. We refer to the group of pixels belonging to each concept class as a concept mask. This number of mask is important, as it motivates the use of Monte Carlo sampling in the subsequent steps.

From these concept masks, EAC aims to identify which one is most important for the prediction made by the target model f_{target}. One potential limitation of the method emerges here: it identifies only a single most relevant concept mask, even though multiple regions may contribute meaningfully to the prediction. We will revisit this point in the **Further Work and Conclusion** section.

2.2 Phase 2 : Training

The training of the surrogate model is the core component of EAC's pipeline. From the previous step, we now have a concept class assigned to each pixel. The surrogate model, $f_{\text{surrogate}}$, must now be defined and trained. As previously mentioned, it consists of a simple sequential layer that takes as input a one-hot encoding of the activated concept masks (a binary vector of size n_{concepts}), and maps it to the input size of f_c, the fixed prediction layer from the target model f_{target}.

Since $f_{\text{surrogate}}$ is trained individually for each image, we do not have a large dataset on which to train it. EAC addresses this by generating a synthetic training set composed of various combinations of concept masks. Although SAM generates around 30 concept masks per image as we said, the total number of possible combinations is approximately 10^{31}. This makes exhaustive enumeration infeasible. To approximate the behavior of f_{target}, we employ Monte Carlo sampling: we generate 2,500 random combinations of concept masks, apply them to the image, and use the resulting masked images and one-hot encoding of the combination to build the training dataset.

Before training $f_{\text{surrogate}}$, we must generate the corresponding ground truth. Since our goal is to approximate the behavior of f_{target}, the ground truth is the prediction output of f_{target} on the previously generated masked images. In the original EAC paper, the surrogate model is trained only to replicate the score for the predicted class (i.e., the argmax of f_{target}'s output). This is another design choice we will discuss further in the **Further Work and Conclusion** section.

Training $f_{\text{surrogate}}$ is then performed on the generated one-hot encoding of the combination of masks, keeping the output layer f_c fixed and using a binary cross-entropy loss between f_{target}'s prediction and $f_{\text{surrogate}}$'s output for the predicted class. The original paper used a learning rate of 0.008, which we adopted as well. The optimizer was not specified in the paper, so we used SGD with momentum, similar to the original implementation. On ours first implementation, we had used Adam but we faced convergence problem with difficulties to tune Adam hyperparameters.

2.3 Phase 3 : Prediction

To estimate the contribution of each concept mask to the prediction, EAC employs a Monte Carlo approximation of the Shapley Value, with 50,000 samples. Originating from cooperative game theory, the Shapley Value quantifies the contribution of an individual agent to a collective outcome. Formally, it is defined as:

$$\phi_i(v) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{\binom{N-1}{k-1}} \sum_{S \in S_k(i)} (u(S \cup \{i\}) - u(S))$$

In the context of EAC, each "player" corresponds to a concept mask, and $S_k(i)$ represents the set of all combinations of concept masks that do not include the i-th. The utility function u is defined by the surrogate model $f_{surrogate}$, which outputs the predicted score for the class of interest. The Shapley Value for a specific concept i is thus the average marginal contribution of that concept to the prediction across different coalitions of concept masks.

As previously mentioned, SAM typically outputs around 30 concept masks per image, making exact Shapley Value computation computationally infeasible due to the exponential number of possible combinations. Therefore, a Monte Carlo approximation is used. The original paper samples 50,000 random combinations per concept to approximate the value.

The final output of EAC is the concept mask with the highest Shapley Value, which is interpreted as the most influential region in the image for the target model's prediction.

3 Evaluation

In this section, we present a comparison of three approaches: our implementation of EAC, the original implementation from the paper, and an alternative method—LIME (Ribeiro et al., 2016). Although the original paper compares EAC to several explainability methods, we limit our comparison to LIME due to time and computational constraints.

For the same reasons, all evaluations were conducted exclusively on the ImageNet dataset (original paper also compared on CoCo). We used a subset of 2,500 images from the validation set for the initial evaluation phase, and a smaller subset of 100 images for the second phase.

3.1 AUC score

The score used as baseline for the evaluation is a re visitation of the Area Under Curve (AUC), adapted to the image classification context. The idea is to order our masks by Shapley Value after the last step of EAC, then to use two schemes to infer AUC scores:

- Deletion scheme: starting from the full image, we delete the masks by order of importance, and compute each time the prediction by the f_{target} model. AUC is then computed with y equal to the prediction and x equal to the percentage of coverage of mask to the full image.
- Insertion sceme: similar to the deletion one, but starting with a fully masked image, and inserting masks by order of importance.

With Deletion AUC, it is better to have a lower score, while for Insertion AUC higher is better.

3.2 Ours vs Original paper on 2500 ImageNet subset

First comparison is done between our understanding of the paper, that is, our own implementation (denoted 'ours') and the original paper's code (denoted 'original'). We are comparing here the aforementioned AUC score1. For all parameters, we used the same as the original paper's ones.

Table 1: Comparison between our implementation and the original paper's code on ImageNet, for Insertion and Deletion AUC computation schemes.

	OURS	ORIGINAL
ImageNet/Insertion AUC ↑	51,547	83,542
ImageNet/Deletion AUC ↓	37,709	23,852

The first thing we can say about those results is that we get the same results for the paper's code as the ones announced in the paper, even though we ran on a smaller sample (2500 images). As for our own implementation, even though it is not visually far and often produce the same output, we failed to reproduce similar AUC scores. Our thoughts on the matter are that even though our EAC process is good, we failed to properly understand and implement the evaluation part.

3.3 Qualitative evaluation

In order to evaluate the quality of the rendered explanatory masks, we coded a simple comparison application, where two images are opposed and the user needs to choose the one explaining the image class the best. We used this to compare our implementation to the original one 3, then the original one to an other approach based on superpixels, LIME4, and lastly our implementation to LIME5. For this qualitative evaluation, the subsample is only 100 images, as it was a bit costly to run LIME on 2500 images. We asked a sample of 5 different users unrelated to our project to give their opinion. We also offered the option to vote for 'DISCORD' when either none of the images was a good explanation, the classification was simply wrong, or both were equivalent.

A first point is that we saw lots of inaccuracies in the class predictions, which makes for a lot of the 'DISCORD' percentages and as nothing to do with the XAI models results.

Table 2: Results for the same process in the original paper, on a sample of 100 images from ImageNet and 100 images from COCO. Other methods were used there, denoted 'OTHERS'.

	ORIGINAL	LIME	OTHERS	DISCORD
Percentage of better explanation masks	70%	2%	21%	7%

Table 3: Qualitative comparison between our implementation and the original paper's code on ImageNet, on a 100 image sub-sample.

	ORIGINAL	OURS	DISCORD
Percentage of better explanation masks	12.6%	2.7%	84.7%

Table 4: Qualitative comparison between the original paper's and LIME approach on ImageNet, on a 100 image sub-sample.

	ORIGINAL	LIME	DISCORD
Percentage of better explanation masks	47.6%	27.7%	26.7%

Table 5: Qualitative comparison between ours and LIME approach on ImageNet, on a 100 image sub-sample.

	OURS	LIME	DISCORD
Percentage of better explanation masks	39.2%	32%	28.7%

From what we saw, whether it is with our results or the ones using the original paper's code, the qualitative evaluation is not as dominant as exposed in the original paper. We chose a random sample of 100 images,

and when comparing EAC and LIME we see EAC is indeed better, but not as much as a 70% to 2% gap between the two methodologies.

The other point is that our reproduction of the method is fairly close to the original EAC, with 84% of discord between the two resulting masks, showing that we produce in around 84% of cases the same output.

Globally, we saw that indeed the explanatory masks resulting from EAC are better than the ones from LIME, but the results are much more mitigated as they were presented originally.

4 Further and Conclusions

In this section, we reflect on the proposed EAC method, suggest possible improvements, and present our conclusions.

SAM One possible improvement lies in refining the segmentation masks produced by SAM. Currently, SAM can generate a very large number of masks—sometimes over 400 per image—and many of these are extremely small or insignificant. By tuning SAM's internal thresholds, we could filter out overly small masks and reduce the overall number of masks, improving both interpretability and computational efficiency.

Shapley value computation Another area for enhancement is the Monte Carlo sampling of concept mask combinations. In the current approach, combinations are sampled randomly. However, a more intelligent sampling strategy could improve the accuracy of Shapley Value approximation. For instance, concept masks could be pre-processed and ordered based on characteristics like size or location in the image, as not all masks contribute equally to the prediction. Additionally, some masks—such as those forming small, semantically meaningful combinations—should always be included in the sampled coalitions.

Surrogate model Regarding the overall process and design choices for the surrogate model, it's important to highlight some limitations and opportunities for improvement.

The surrogate model is trained to replicate the output of the target model only for the predicted class—that is, the class with the highest probability in the original prediction. While this simplifies the task, it also significantly reduces the amount of information the surrogate learns. In the case of models like ResNet trained on ImageNet, which outputs probabilities across 1,000 classes, focusing solely on a single output class seems like a limited approximation of the target model's behavior.

Additionally, the current version of EAC only returns one concept mask—the one with the highest Shapley Value—as the explanation. While this is useful in many cases, it may be insufficient in situations where:

- Multiple objects in the image contribute to the same class prediction.
- The prediction is better explained by a combination of masks rather than just one.

In such cases, it would be more informative to return several concept masks, for example, all those with Shapley Values above a certain threshold, or the top-k masks when their contributions are similar. This would provide a more nuanced and complete explanation, particularly for complex images.

The surrogate model needs to strike a balance between accuracy and computational efficiency. It must approximate the target model's behavior closely enough to be a reliable proxy, but also remain lightweight to ensure fast training and inference. In our implementation, we use a one-layer MLP. This raises the question: is a single layer sufficient to capture the complexity of the target model's decision function?

The choice of training algorithm and its hyperparameters can have a significant impact on performance. Our work relied on SGD with momentum, but further experiments with other optimizers—such as Adam or RMSProp—could lead to better results. A more systematic hyperparameter search would also help identify the most effective configuration, though the best choices may depend on the complexity of the surrogate model.

Issues faced We encountered several issues during the reproduction of the original EAC implementation, primarily related to performance. Initially, our code was not designed to be highly parallelized, which resulted in extremely slow execution times and made the process impractical for large-scale experiments.

Another unresolved issue concerns the computation of the AUC score. Despite obtaining visually similar results to those of the original implementation—particularly when examining the masked images—our AUC scores diverge significantly, especially in the insertion scheme. We thoroughly reviewed our code multiple times but were unable to identify the source of this discrepancy.

Conclusion Overall, we consider the EAC method to be a promising alternative to existing explainability techniques. Although our results are more mixed compared to those reported in the original paper, they still outperform the LIME approach, suggesting that EAC remains an effective method for generating meaningful explanations.

References

Yuanyuan Yuan Ao Sun, Pingchuan Ma and Shuai Wang. Explain any concept: Segment anything meets concept-based explanation. *NeurIPS 2023 Conference*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL `https://arxiv.org/abs/1512.03385`.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL `https://arxiv.org/abs/2304.02643`.

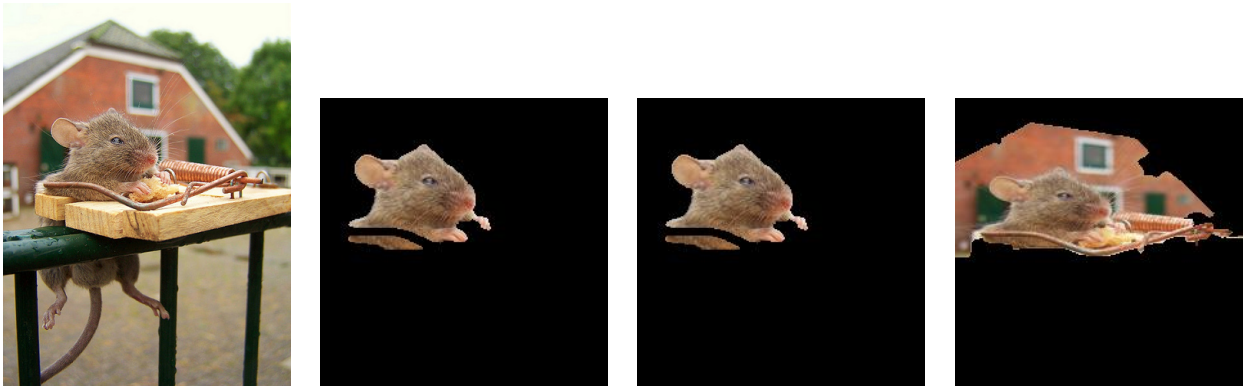
Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

A Appendix

Figure 1: Application interface

(a) IMAGE (b) ORIGINAL (c) OURS (d) LIME

Figure 2: Image ILSVRC2012_val_00000008.JPEG and its different masks for the class: bakery

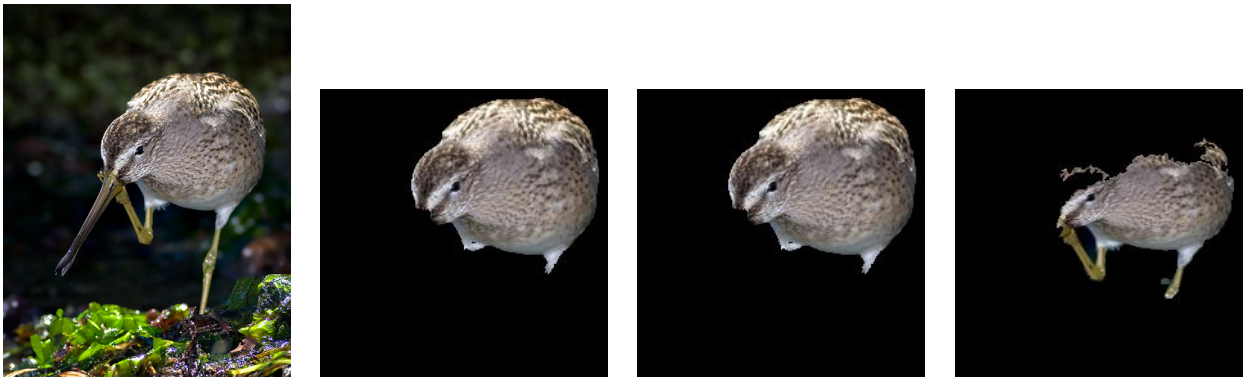


(a) IMAGE (b) ORIGINAL (c) OURS (d) LIME

Figure 3: Image ILSVRC2012_val_00000009.JPEG and its different masks for the class: mousetrap

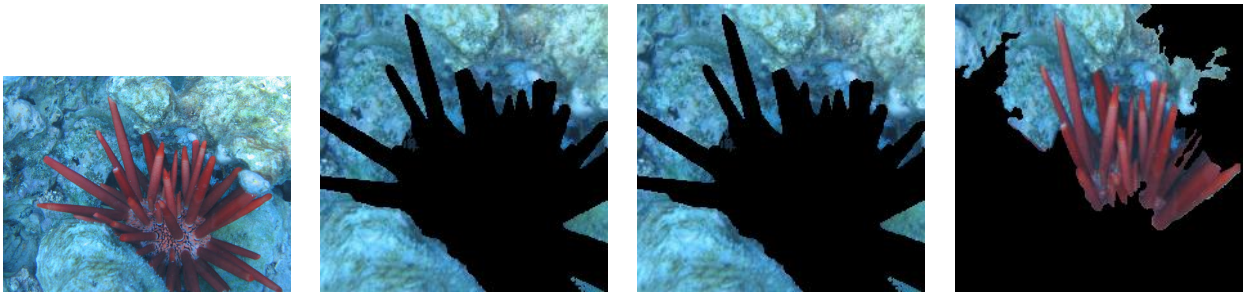
(a) IMAGE (b) ORIGINAL (c) OURS (d) LIME

Figure 4: Image ILSVRC2012_val_00000054.JPEG and its different masks for the class: handkerchief



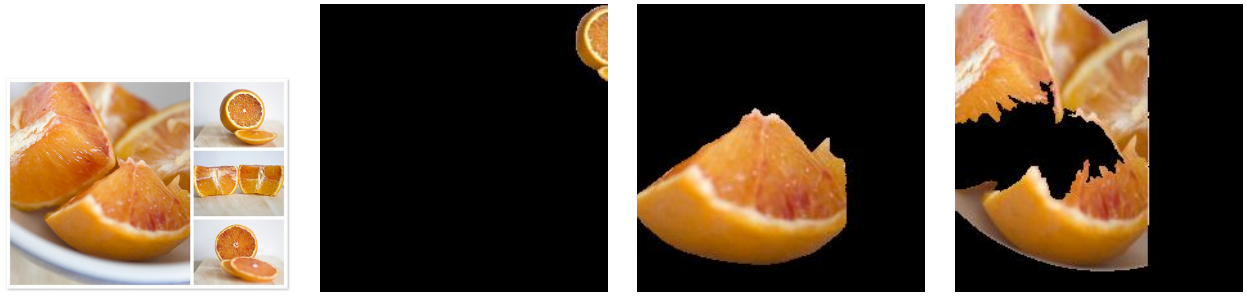
(a) IMAGE (b) ORIGINAL (c) OURS (d) LIME

Figure 5: Image ILSVRC2012_val_00000098.JPEG and its different masks for the class: dowitcher



(a) IMAGE (b) ORIGINAL (c) OURS (d) LIME

Figure 6: Image ILSVRC2012_val_00000142.JPEG and its different masks for the class: sea urchin



(a) IMAGE (b) ORIGINAL (c) OURS (d) LIME

Figure 7: Image ILSVRC2012_val_00001406.JPEG and its different masks for the class: orange