Explaining deep convolutional models by measuring the influence of interpretable features in image classification

Data Mining and Knowledge Discovery (2024) 38:3169–3226
https://doi.org/10.1007/s10618-023-00915-x

Explaining deep convolutional models by measuring the influence of interpretable features in image classification

Francesco Ventura[1] · **Salvatore Greco**[1] · **Daniele Apiletti**[1] · **Tania Cerquitelli**[1]

Received: 15 June 2021 / Accepted: 2 January 2023 / Published online: 10 February 2023

Abstract

The accuracy and flexibility of Deep Convolutional Neural Networks (DCNNs) have been highly validated over the past years. However, their intrinsic opaqueness is still affecting their reliability and limiting their application in critical production systems, where the black-box behavior is difficult to be accepted. This work proposes EBAnO, an innovative explanation framework able to analyze the decision-making process of DCNNs in image classification by providing prediction-local and class-based model-wise explanations through the unsupervised mining of knowledge contained in multiple convolutional layers. EBAnO provides detailed visual and numerical explanations thanks to two specific indexes that measure the features' *influence* and their *influence precision* in the decision-making process. The framework has been experimentally evaluated, both quantitatively and qualitatively, by (i) analyzing its explanations with four state-of-the-art DCNN architectures, (ii) comparing its results with three state-of-the-art explanation strategies and (iii) assessing its effectiveness and easiness of understanding through human judgment, by means of an online survey. EBAnO has been released as open-source code and it is freely available online.

Responsible editor: Martin Atzmueller, Johannes Fürnkranz, Tomáš Kliegr, Ute Schmid

Francesco Ventura and Salvatore Greco have contributed equally to this work.

✉ Salvatore Greco
salvatore_greco@polito.it

Francesco Ventura
francesco.ventura@polito.it

Daniele Apiletti
daniele.apiletti@polito.it

Tania Cerquitelli
tania.cerquitelli@polito.it

[1] Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy

Keywords Explainable artificial intelligence · Explainability · Deep convolutional neural network · Image classification · Black-box classifier

1 Introduction

Modern decision-making processes have been highly improved with the advent of Artificial Intelligence's deep learning models (e.g., deep neural networks). Natural image understanding is one of the fields that benefited the most from these research efforts, with the introduction of even more accurate and complex Deep Convolutional Neural Networks (DCNN) (Simonyan and Zisserman 2015; Szegedy et al. 2016, 2017), establishing new standards in many machine-learning tasks. However, the decision-making process of DCNN models is still far from being understood by its users since it is a black-box, as widely highlighted by several researchers (Guidotti et al. 2018; Lepri et al. 2017; Ribeiro et al. 2016). Moreover, it is commonly known that deep black-box models require a huge amount of data to be trained, usually generated and evaluated by people, increasing the risk of inheriting various forms of human prejudice and bias (Lepri et al. 2017; Bolukbasi et al. 2016; Ribeiro et al. 2016). The critical social impacts that deep AI models are causing on our modern society stimulate the advancements in the *eXplainable Artificial Intelligence* (XAI) field of research (Confalonieri et al. 2021). Many researchers are devoting efforts to explainable and interpretable approaches, from the classification of structured and unstructured data (Datta et al. 2016; Ribeiro et al. 2016; Lundberg and Lee 2017), to recommendation systems (Lin et al. 2020), and knowledge discovery (Yeo et al. 2020).

In the literature of explanation frameworks, applicable to the image processing domain, both model-agnostic and domain-specific approaches are available. Model-agnostic solutions (Ribeiro et al. 2016; Lundberg and Lee 2017) are general-purpose and sometimes they propose visual and numerical explanations. However, their generality often limits the quality and the reliability of the provided explanations, especially when complex neural networks are used since they are not able to mine the knowledge contained in the model under analysis. Domain-specific solutions (Selvaraju et al. 2019; Petsiuk et al. 2018; Simonyan et al. 2014), on the other hand, typically exploit shallow model information (i.e., few network layers) and often provide only prediction-local visual explanations. Furthermore, very few works validate their solutions considering humans' feedback (Ribeiro et al. 2016) to assess the quality and the ease of understanding of the provided explanations. Nevertheless, explanation frameworks are built for humans, so their validation is crucial.

To bridge such gaps, we propose EBAnO (Explaining BlAck-box mOdels), an innovative explanation framework able to analyze the decision-making process of convolutional models providing prediction-local and class-based model-wise explanations through unsupervised mining of the inner knowledge contained in multiple layers of the DCNN. EBAnO provides both visual and numerical explanations, enabling both expert and non-expert users to better understand the reasons behind the predictions by projecting them on the input image. The main contributions of this work can be summarized as follows.

- The design of a novel explanation process exploiting the inner knowledge of multiple convolutional layers simultaneously.
- The introduction of a new index (nPIRP) to efficiently quantify both the *influence* and the *precision* of the input features w.r.t a given prediction.
- The unsupervised extraction of *interpretable features* easily understandable by humans and projected on the input image.
- The computation of informative class-based model-wise explanations.
- Qualitative human validation of the effectiveness and easiness of understanding of EBANO's explanations through an online survey.
- Quantitative validation of the proposed approach, producing almost 10,000 explanations for four state-of-the-art DCNNs on 250 input images.
- Qualitative, quantitative, and human-based comparison of EBANO with state-of-the-art explanation tools, i.e., *LIME* (Ribeiro et al. 2016), *Grad-CAM* (Selvaraju et al. 2019), and *Shapley values* (Štrumbelj and Kononenko 2014).

EBANO's open-source code repository, an interactive library of explanations produced by EBANO, and the online survey proposed to the users are available online.[1]

The rest of the paper is organized as follows. Section 2 presents the XAI state-of-the-art. Sections 3, 4, 5, 6, and 7 describe the local-explanation, while Sect. 8 the model-global explanations processes implemented in EBANO. Then, Sect. 10 reports the experiments, the qualitative and quantitative comparisons, and the human validation results. Sect. 11 discusses the current limitations of the framework. Finally, Sect. 12 provides conclusions and proposes future works.

2 Related work

More and more decisions, often critical and socially impacting, are taken by complex, intrinsically black-box, machine-learning models (Lepri et al. 2017). Thus, new explainable and interpretable approaches are starting to spread, highlighting the importance of the *eXplainable Artificial Intelligence* (*XAI*) in several domains. Many effective solutions have already been proposed, especially to explain in the domain of structured data (Proença and van Leeuwen 2020; Yeo et al. 2020), and recommendation systems (Díez et al. 2020; Zheng et al. 2019; Lonjarret et al. 2020).

This work focuses on the explanation of visual information processing performed by Deep Convolutional Neural Networks (DCNNs), a family of neural networks widely spread in computer vision (Simonyan and Zisserman 2015; Szegedy et al. 2016, 2017). In this section, we discuss the state-of-the-art explainability techniques, with a specific focus on those suitable for DCNN and computer vision.

2.1 Model-agnostic approaches

Some of the most promising efforts in the XAI field have already been collected in Guidotti et al. (2018); Adadi and Berrada (2018) where the authors try to explore and

[1] https://ebano-ecosystem.github.io.

analyze all the possible requirements that an explanation process should be able to fulfill.

Several techniques exploited by data scientists when dealing with black-box models are model-agnostic explainability approaches, like *LIME* (Ribeiro et al. 2016) and *SHAP* (Lundberg and Lee 2017). In particular, *LIME* (Ribeiro et al. 2016) allows to produce a prediction-local explanation of any predictive model, and it applies to both structured and unstructured data (e.g., image, text); to compute the local explanation, *LIME* performs a local approximation of the prediction, training a simpler and interpretable local model around small variations of the input data. *SHAP* (Lundberg and Lee 2017) proposes a unified approach to interpret local predictions produced by any machine learning model. *SHAP* is based on the idea of collaborative contribution coming from the *Game Theory*, measured by exploiting the concept of *Shapley Values* (Shapley 1953). Thus, in the prediction process, the model outcomes are considered as a collaborative contribution of the elements that compose the input data and the local explanation is given by the measure of the contribution of each feature in the prediction task. A further contribution in model agnostic prediction-local explanations for structured data (i.e., tabular data) is proposed in Rajapaksha et al. (2020) by exploiting an association rule mining approach to extract not only the rules that are supporting the current prediction but also the ones that are contradicting it and the associations that the model would require to change its outcome. However, while association rule approaches are suitable for structured problems, they have limited applications on the explanation process of machine learning tasks on unstructured data (e.g. image classification). Moreover, Kliegr et al. (2021) shows, from a psychological perspective, how interpretable machine learning models, and in particular logical rules, can be affected by cognitive biases. Consequently, also rule-based explanations could be affected by them.

Model agnostic techniques are really powerful and simple to use in many domains, but often they provide very approximate explanations, limiting their reliability in critical contexts. They are not able to analyze the prediction process taking advantage of the information contained in the model under analysis and to give specific outcomes taking into account the domain of interest. For these reasons, EBAnO leverages domain knowledge (i.e., DCNN for image classification) to produce more effective and reliable explanations.

2.2 Domain-specific approaches

Image understanding requires more domain-specific approaches, enabling the production of even more accurate and reliable explanations. To study the behavior of a DCNN during the prediction process taking advantage of the knowledge contained in the model itself, two types of approaches are the most common: (i) studying the model inner behavior, layer by layer, visualizing their output and trying to infer the details of the process that brought the model to a specific decision or (ii) exploiting the information produced by the model during the prediction phase to understand which are the portions of the input that mostly affect the decision process.

Several interesting approaches based on a graphical analysis of the network's neurons, inspecting the architecture of different convolutional layers through visualization techniques, have been proposed and summarized in Seifert et al. (2017). The effectiveness of this family of techniques is out of doubt when dealing with small architectures but they became nearly not applicable as the complexity of the network grows. Even more important, the prediction process analysis through these graphical techniques is strictly oriented to technical and domain expert users, limiting their applicability in many areas of interest.

On the contrary, many approaches proposed in literature aim to understand which portions of the input mostly affect the decision process (Binder et al. 2016; Simonyan et al. 2014; Fong and Vedaldi 2017; Zhang et al. 2018; Petsiuk et al. 2018; Selvaraju et al. 2019; Sundararajan et al. 2017; Selvaraju et al. 2019; Bach et al. 2015; Smilkov et al. 2017; Shrikumar et al. 2017). (Simonyan et al. 2014) explores strategies to produce (i) prediction-local explanations exploiting the inner information of the model, visualizing the portions of the input mostly characterizing the prediction through saliency maps and (ii) class-local explanations exploiting the CNN model under analysis to generate class-related images to maximize the probability of the class-of-interest. The authors in Binder et al. (2016) propose an extension of LRP (Bach et al. 2015), a technique which allows decomposing the prediction of a DNN into feature relevance scores, suitable for local normalization layers' non-linearity in convolutional neural networks exploiting Deep Taylor Decomposition (Montavon et al. 2015). SPRAY (Lapuschkin et al. 2019), instead, exploits Spectral clustering on LRP (Bach et al. 2015) explanations to globally explain models over large-scale datasets identifying typical and atypical patterns in the heatmaps. Zhang et al. (2018) studies how to modify traditional CNNs to make them self explainable, leveraging the idea that each convolutional layer should be activated only by a certain object part belonging to a specific category and highlighting the object parts with feature maps. Fong and Vedaldi (2017) proposes a paradigm that learns the minimally salient part of an image, finding the smallest perturbation mask that brings down the classification score. RISE (Petsiuk et al. 2018) analyses the effect of perturbing randomized input samples to produce prediction-local explanation in a general fashion, without taking into account the internals of the model under analysis, and measuring the effects that the *deletion* and *insertion* of input pixels have on the outcomes of the prediction process.

Many approaches available in literature exploit the concept of input *perturbation* to analyze the model reactions, like Alvarez-Melis and Jaakkola (2017); Ventura et al. (2018); Lundberg and Lee (2017); Ribeiro et al. (2016); Selvaraju et al. (2019); Fong and Vedaldi (2017). This idea, however, requires that the input features to be perturbed contain meaningful information for the model, otherwise the perturbation will not be able to highlight the importance of the perturbed portion of the image in the prediction process. Different from most of the *perturbation-based* approaches that create and evaluate a very large number of small perturbations, EBAnO implements a feature extraction process itself that extracts more effective features directly from the latent information hidden into the layers of the model. Therefore, EBAnO overcomes one of the major limitations of most of the perturbation-based techniques, i.e., the quality of their explanations depends on the number of perturbations tested, being inefficient.

Indeed, EBAnO perturbs the right portions of the input directly as a result of the unsupervised analysis of the input layers.

In contrast, other approaches compute features importance by *back-propagating* the predictions through each layer of the network until input pixels (Selvaraju et al. 2019; Sundararajan et al. 2017; Bach et al. 2015; Smilkov et al. 2017; Shrikumar et al. 2017; Kapishnikov et al. 2019). For instance, *Grad-CAM* (Selvaraju et al. 2019) proposes a *gradient-based* saliency approach to produce prediction-local explanations. It is based on the study of the gradient output of the last convolutional layer in a DCNN, generalizing the approach proposed in Zhou et al. (2016): it produces a saliency map that highlights the specific regions of the input that are mostly characterizing the prediction. Grad-CAM (Selvaraju et al. 2019) has been tested in a wide range of use cases showing its generality and it has been human-validated to assert the clearness of the produced explanations. However, this family of explanations, despite being more efficient in terms of complexity and runtime than perturbation-based methods, are often affected by noisy gradients (e.g., importance value assigned to neighboring individual pixels is affected by high-frequency variations) or issues with some typical layers frequent in CNN such as *max pooling* (Ancona et al. 2019).

Unlike most of the techniques discussed above, defined as *feature-based* explanations, the *concept-based* explanations attempt to provide explanations in the form of high-level human-readable concepts (Ghorbani et al. 2019; Yeh et al. 2020; Kim et al. 2018). TCAV (Kim et al. 2018) is a perturbation-based global explanation method that generates explanations by measuring the importance of human-defined concepts. Specifically, it extracts the class activation vector (CAV) by training a binary linear classifier using some positive and negative examples of the concept and extracting its weights. Then, it computes the directional derivates of the model's predictions with respect to the class activation vector to quantify the per-concept feature importance. One weakness of the approach is that the choice of examples is subjective and strictly affects the explanations produced (and could be influenced by human biases). ACE (Ghorbani et al. 2019) is a similar approach that, instead of the human choice of concept samples, relies on unsupervised clustering analysis of different resolution segments extracted from the set of images exploiting an ImageNet-trained CNN. However, it requires a further black-box model that could not reflect the feature learned by the model to be explained, especially in domain-specific tasks (i.e., with images really different from ImageNet). ConceptSHAP (Yeh et al. 2020) adapts Shapley values (Shapley 1953) to assign importance to each concept, and defines a completeness score to measure how sufficient are the concepts in explaining the model. However, being only global techniques, these methodologies are not able to explain in an effective and simple way the specific reasons behind single predictions, but only to globally explain the model.

Finally, other new emerging approaches attempt to produce expressive and verbal explanations. For instance, Rabold et al. (2020) extracts relational information from the inner layers of a DCNN to build an expressive global explanation by combining concept analysis and inductive logic programming, supporting the idea that explanations looking directly in the inner latent space of the model to extracts semantic concepts (features) are more reliable.

In conclusion, EBAnO is a *domain-specific* and *perturbation-based* methodology to *locally* and *globally* explain deep convolutional neural networks for image classification. It exploits the hidden internal knowledge of the model by mining the embedding representation (i.e., Hypercolumns) to produce more faithful, reliable and human-readable explanations without relying on or training any additional classifier. Moreover, it is able to *visualize* and *measure* the impact of both positively and negatively influential features, exploiting two quantitative indices to measure the *influence* and the *precision*. It is a big step forward from a preliminary idea described in Ventura et al. (2018), and partially applied, as a completely different solution, in the Natural Language Processing context (Ventura et al. 2022).

EBAnO deeply reshapes the authors' previous work by (i) combining the concepts of influence relation and influence precision,[2] (ii) extending the applicability of the approach to a new domain and considering a larger variety of DCNNs, (iii) defining a new strategy to compute the most informative explanation, (iv) revisiting the production of the visual explanations, (v) introducing class-based model global explanations, (vi) providing a deep qualitative, quantitative and human-subjective comparison with the state-of-the-art approaches, and (vii) human-validating the produced prediction-local explanations. (viii) experimentally evaluate the influence metrics on a larger multi-class problem (e.g., up to 1000 classes).

Despite the number of works that are exploring the explainability in the context of image understanding, further improvements are needed to fill some gaps. EBAnO improves the state-of-the-art by introducing a new unsupervised model-aware strategy that is able to (i) extract the information contained in multiple convolutional layers, exploiting the *Hypercolumns* (Hariharan et al. 2015) representation, (ii) identify relevant and *interpretable* input features by studying the contribution of each of them through an iterative perturbation process, (iii) quantify the positive or negative combination of *influence relation* and *influence relation precision* for each *interpretable feature*, (iv) produce both visual and numerical explanations (v) provide both detailed prediction-local and class-based model-global explanations. Although some of EBAnO's features are present in other techniques, it encapsulates them in a single framework to provide *human-readable*, *reliable* and *effective* explanations, suitable also for both technical expert and non-expert users.

3 Explanation process overview

EBAnO provides a detailed *prediction-local explanation* of a black-box outcome, given an input image and a DCNN predictive model. The prediction-local explanation aims to explain the reasons for the specific predicted class label of the black-box model given a single instance of an input image, and its main steps are shown in Fig. 1.

Firstly, an image is given as input to the black-box DCNN model in ①, producing the original predicted label and probabilities in ②. Then, the hypercolumns of the input image are extracted from the black-box model in ③ and are processed,

[2] A preliminary version of the influence relation index (nPIR) in the context of Natural Language Processing has been proposed in Ventura et al. (2022).

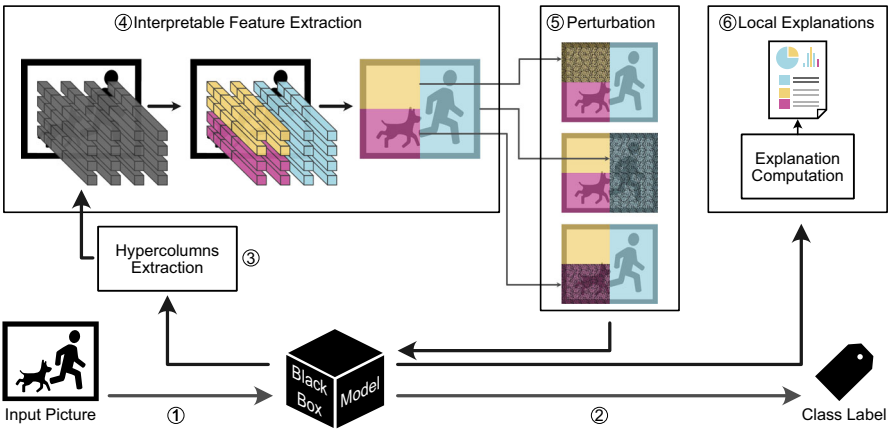


Fig. 1 EBANO's local explanation process

through an unsupervised analysis, to obtain a set of interpretable features in ④. The interpretable features are composed of semantically related groups of pixels (with similar inner representation extracted from the hypercolumns) representing human-understandable concepts that similarly influenced the original labels predicted by the model (as detailed in Sect. 4). Then, a new version of the input image is produced for each extracted feature by introducing some noise over their pixels (one new image for each feature) in ⑤. The perturbation step is required to understand the importance of each interpretable feature with respect to the prediction probabilities of the original class labels by the DCNN model (details in Sect. 5.1). To this aim, the DCNN model is presented with the perturbed images, and the different classification probabilities are analyzed. Comparing the probabilities before and after the perturbation, we can meet one of the following three cases.

1. Suppose the probability of belonging to a class label *decreases*. In that case, the noise over pixels of the interpretable feature (that caused an absence of the associated concept) is the main responsible for this decrease. Therefore, the feature was *positively impacting* (or influential) for the prediction of the original class label for the DCNN model.
2. If, instead, the probability *remains the same*, then the feature was *neutral* in the prediction of the class label.
3. Finally, if the probability *increases*, then the feature was *negatively impacting* the prediction of the original class label.

The *amplitude* and *precision* of the influence of each feature are measured with two quantitative indices ranging from -1 to +1, namely *nPIR* and *nPIRP* (as discussed in Sects. 5.2 and 5.3). Finally, EBANO produces the *local explanation* in ⑥, which consists of a *numerical explanation* (with the numerical indices for all the extracted interpretable features) and a *visual explanation* that shows the interpretable features over the original pixels of the image, with the corresponding influence highlighted by a color heatmap (as detailed in Sect. 7).

Fig. 2 *Pizza* input image

Table 1 Predictions for Fig. 2 with a pre-trained VGG16

Class	$P(c)$
Bottlecap	0.42
Pizza	0.28
Bakery	0.08
Trifle	0.06
Dining table	0.03

The ground-truth label is in bold (Pizza)

Explaining the classifier's prediction can be useful for understanding the reasons for a possible misleading prediction and understanding if the model focused on the correct portion of the image, even if the predicted class label is correct. Table 1 shows an example of wrong prediction by a pre-trained VGG16 model given the input image in Fig. 2. The black-box model predicted as most probable class label *Bottlecap* followed by *Pizza* with respectively probabilities 0.42 and 0.28 (even if the ground truth label was pizza). Still, the black-box nature of the model hides the reasons. In this case, an end-user can be interested in understanding the features that influenced the model's predictions for both the class labels. We will use this as a running example to explain the whole methodology in the following sections.

In Sect. 4, we provide a detailed discussion on how EBAnO is able to extract the interpretable features from the unsupervised analysis of the hypercolumns. In Sect. 5, we discuss the measurement of the feature importance through the process of perturbation and, in particular, we formally define the two quantitative indices that measure the influence and precision. In Sect. 6, we show how EBAnO automatically discovers the best possible explanation. Finally, in Sect. 7, we discuss how the final local explanation is produced.

4 Interpretable feature extraction

The objective of EBAnO is to identify the input pixels that mostly contribute to the DCNN prediction. However, the single-pixel contribution does not provide interpretable results and is computationally demanding. Thus, we identify the sets of correlated pixels mostly influencing the outcome of the black-box model, yielding more understandable results for humans since it is simpler for a human to evaluate macro portions of an image than single pixels. Indeed, the higher is the level of the features (i.e., pixels belonging to full objects), the higher their interpretability is by humans. Moreover, the higher is the fidelity of the explanation with respect to what the model has effectively learned, the higher is the reliability of the explanation itself. The proposed feature extraction strategy aims to identify meaningful and interpretable portions of the input image exploiting the unsupervised clustering analysis of *hypercolumns* (Hariharan et al. 2015), which are a vectorial representation of the model across all its inner levels. The segments extracted from the input image with this strategy are called *interpretable features*.

Existing state-of-the-art techniques are not suitable to obtain interpretable features for our purposes. On the one hand, deep models for image segmentation (Minaee et al. 2020) are able to extract full segments of pixels from input images corresponding to entire concepts (i.e., people, animals, background, etc.) that, in theory, can be used as interpretable features for EBAnO. However, even if these high-level segments are straightforward to understand by humans, it is impossible to assume that the whole segment impacted more the predicted class label by the deep learning model (and not only a part of it), increasing the risk of losing fidelity. For example, if a DCNN model predicts the *Dog* class label by focusing on the dog's eyes, by using a segmentation model that extracts the dog's pixels, this aspect would not be captured. Moreover, another limitation of the segmentation models for our purposes is that it would be required another model performing the segmentation that would have a number of fixed classes depending on the classification task. Consequently, the explanation process would not be portable. For example, a model that extracts segments in images for self-autonomous driving cars cannot be used to extract interpretable features for the explanation of a deep model used in the medical field and vice versa.

On the other hand, other existing techniques like adversarial attacks (Akhtar and Mian 2018) are able to create new perturbed variations of the input image that are not perceptible to the human eyes but that cause a wrong prediction of the model. These perturbations are composed of adversarial pixels that, added to the input image, create a new perturbed variation that looks identical to the human eyes but that will be misclassified by the model. However, these techniques can be used to explain some global weaknesses of the model but not to locally explain a single prediction for a given input image (for how the pixels are changed). Furthermore, the pixels changed by this technique are not easily interpretable by humans and, therefore, are not suitable for a good explanation.

For these reasons, EBAnO, by implementing itself the interpretable feature extraction strategy, is able to exploit, in an unsupervised way, the inner knowledge hidden in the learned weights of the model in order to extract correlated portions of the image that influenced the final output of the model similarly. Moreover, these features rep-

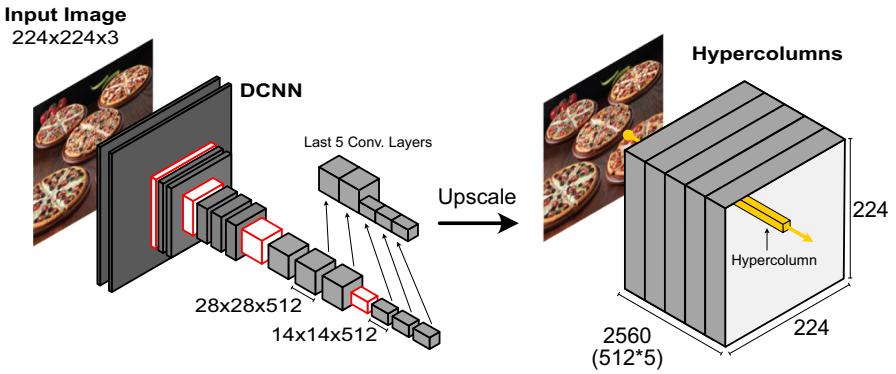


Fig. 3 Hypercolumns exctraction example

resent macro-concepts that are easily interpretable by humans and reflect what the model has effectively learned. Finally, the feature extraction strategy of EBAnO does not require training any additional model, but it can be integrated directly into the model to explain itself.

4.1 Hypercolumns extraction

Hypercolumns have been defined by Hariharan et al. (2015) as a vectorial representation of every input pixel. The main idea is that, if bias or knowledge have been learned by the black-box model, they can be extracted by mining the latent information under the form of *hypercolumn*, thanks to their ability to collect the information of the outputs related to a specific location across all the layers of the DCNN. The first layers of the DCNN are able to generalize over the shape of objects, identifying corners and edges, whereas the final layers are more sensitive to the semantic meaning of an image (Bengio et al. 2013; Mahendran and Vedaldi 2016; Hariharan et al. 2015). The hypercolumns of a specific input can be extracted feeding into the black-box model the target image: each convolutional layer of the network outputs a tensor that is the results of the application of the weights learned by the model. These tensors contain all the latent information learned by the model during the training phase. In the case of a very deep network, using all the convolutional layers to extract the hypercolumns can produce a very deep tensor which is difficult to manage. However, EBAnO focuses on the most characterizing information of the model, which is usually included in the deepest layers of the network, i.e., the deeper the layer, the more specialized it is, and the information that can be extracted from it are very task-specific. For this reason, the number of layers that should be considered is usually much lower than the total number of available layers in the network. The number of layers exploited to extract hypercolumns is a parameter that remains empirically configured, being related to target the network's architecture.

Figure 3 shows an example of hypercolumns extraction from a DCNN given an input image. In this example, the last 5 convolutional layers of the DCNN model are extracted. Then, an upscaling step is performed by exploiting bilinear interpolation to

lead back to the original size of the input image. After the upscale, the hypercolumn representation of the input image is composed of a tensor with the same width and height of the original image and a number of channels equal to the sum of the channels in the extracted layers. Notice that each pixel is represented by a vector representation of the same dimension. In this example, after the upscaling, the final representation is a tensor of shape (224, 224, 2560), and therefore, each pixel is represented by a vector of dimension 2560.

4.2 Feature extraction

Through an unsupervised clustering analysis of hypercolumns, EBANO can identify correlated beams of vectors and subsequently detect the input areas to which they correspond. EBANO projects the grouped beams of hypercolumns on the input image by labeling each pixel with its cluster. A peculiar feature of EBANO is that the clustering of the hypercolumns is not led by the pixels' locality in the image but only by the *weights* learned by the DCNN model, hence driving the explanation with the inner information of the model itself. Therefore, differently from other algorithms, the image segmentation strategy implemented in EBANO does not consider input colors or pixel positions, but it is strictly related to what the model has learned: the segments highlighted with this strategy reflect the inner knowledge of the model. Thus we expect to provide more relevant local explanations. This is because pixels with similar representation possibly represent similar and related semantic aspects that probably affected the original prediction of the model in a similar way. Consequently, the features are obtained by performing a clustering analysis in a high-dimensional embedding space defined by the hypercolumns, where each pixel is represented with a high-dimensional dense embedding vector.

To this aim, it exploits the Faiss (Johnson et al. 2021)[3] implementation of K-Means (Lloyd 1982), an efficient similarity search and clustering of dense vectors library that also supports GPUs. It was chosen because, from the experiments, it results in the best in terms of effectiveness and efficiency in extracting relevant features. However, An experimental evaluation of different clustering algorithms' performance in the feature extraction process of EBANO is provided in "Appendix B". The *K-Means* algorithm requires the specification of the number of clusters k, which determines the number of resulting image portions (i.e., our *interpretable features*) extracted from the input image. However, it is impossible to know the best number of features to extract from the input in advance. For this reason, EBANO iteratively produces the explanations for different possible divisions (i.e., different values of k) and chooses the best one. Nevertheless, the process is the same for each value of k. Thus, we first discuss the process of perturbation and influence measurement of each feature in Sect. 5. Then, in Sect. 6, we show how the best explanation is selected among the different values of k evaluated.

Figure 4 shows an example of *interpretable features* extracted from the running example introduced in Fig. 2 with a number of clusters $k = 3$. The features extracted are driven by the inner knowledge hidden in the model (hypercolumns) and reflect

[3] https://github.com/facebookresearch/faiss.

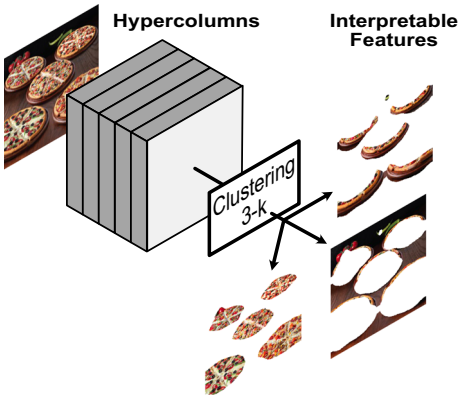


Fig. 4 Interpretable features extraction example

what the model has effectively learned, but also are easy to interpret and understand by humans.

5 Measuring the features' influence

After extracting the interpretable features, a perturbation phase is required to measure the importance and the impact that each feature had in the originally predicted output of the model, given the input image. Firstly, we explain the perturbation process implemented in EBANO in Sect. 5.1. Then, we formally define the two quantitative indices, introduced in EBANO, to measure the *influence* (*nPIR*) and the *precision* (*nPIRP*) in Sects. 5.2 and 5.3.

5.1 Perturbation

Ideally, each *interpretable feature* could represent a relevant concept of the input image. Our challenge is to identify the most relevant portions of the input for both the model and the user. The perturbation of the model's input is a well-known state-of-the-art technique (Alvarez-Melis and Jaakkola 2017; Ventura et al. 2018; Lundberg and Lee 2017; Ribeiro et al. 2016) to study the impact of input data on the prediction outcome. Our intuition is to drive the perturbation on the specific pixels of the *interpretable features* and measure the prediction difference on those concepts.

EBANO implements an iterative perturbation process based on Gaussian blur.[4] Specifically, for a given feature, the relative pixels are occluded with a sequence of extended box filters, which approximates a Gaussian kernel (Gwosdek et al. 2012). It requires the Gaussian radius parameter specification that we empirically set to 10. A new perturbed image is produced for each blurred *interpretable feature*, and we expect the model to miss the recognition of the corresponding concept; three results are possible: (i) no change in prediction (the concepts represented by the feature

[4] Exploiting the GaussianBlur implementation of the Pillow (Clark 2015) library. https://pillow.readthedocs.io/en/stable/reference/ImageFilter.html(January, 2022).

(a) Gaussian blur. (b) Black occlusion. (c) White occlusion. (d) Mean occlusion.

Fig. 5 Perturbed images examples with different occlusion techniques. Gaussian blur **(a)** is the one adopted by EBANO, and it blurs the feature pixels with a sequence of extended box filters, which approximates a Gaussian kernel. Other possible occlusion techniques consist of replacing the feature pixels with black **(b)**, white **(c)**, or the mean value of pixels **(d)**.

were not relevant for the predicted class); (ii) stronger prediction (the probability of belonging to the predicted class increases after the perturbation, hence, removing the feature, the predicted class is better modeled); (iii) weaker prediction (the probability of belonging to the predicted class decreases after the perturbation, hence the feature was important to model the class).

Even if several possible perturbation solutions exist, EBANO exploits Gaussian blur because it is less likely to create artifacts that would cheat the model. Figure 5 shows an example of a perturbed image, created for one of the features extracted from Fig. 4 with different perturbation techniques. The perturbation introduced by the Gaussian blur Fig. 5a is less invasive than setting the pixels values to 0 (black occlusion in Fig. 5b), to 255 (white occlusion in Fig. 5c), or to the mean value over the entire image (mean occlusion in Fig. 5d).

An experimental evaluation of the Gaussian blur radius parameter's impact is provided in "Appendix A".

5.2 Normalized perturbation influence relation—*nPIR*

Once applied the perturbation over each extracted feature, we want to measure the impact that pixel occlusion has caused. Specifically, the first aspect that we want to measure is the *sign*, the *amplitude* of the impact, and the *relative* influence of an interpretable feature for the prediction of the class label with the *normalized Perturbation Influence Relation* (nPIR) index.

Formally, let's consider a black box model able to distinguish between a set of classes $c \in C$ and let be $ci \in C$ the *class-of-interest* for which an explanation has to be computed. Given an input image I, EBANO extracts the set of *interpretable features* $f \in F$ and then for each f it performs the perturbation. Let's consider $p_{o,ci}$ as the probability of the original input image I (the unperturbed image) to be labeled with the class-of-interest ci by the model, and $p_{f,ci}$ as the probability of the same image to be labeled with the same class-of-interest ci when the feature f is perturbed. If $p_{f,ci}$ is lower than $p_{o,ci}$, than f contains a positively influential concept for the model and vice versa. To measure this effect, we exploit the nPIR index, which includes different

components, in particular (i) the amplitude of the impact and (ii) its relative influence on the perturbation process.

The amplitude of the perturbation influence, ΔI, for feature f can be measured by: $\Delta I_f = p_{o,ci} - p_{f,ci}$. It ranges from -1 to 1 since the domain for probability values falls in [0, 1]. If $\Delta I_f > 0$, the feature f has a positive influence on ci, since its perturbation causes a decrease of the probability to belong to ci, and vice versa.

The relative influence of the perturbation as a simple ratio between the probabilities was proposed in Ventura et al. (2018). However, it is asymmetric, as $\frac{p_{o,ci}}{p_{f,ci}}$ ranges from 0 to 1 in case of negative influence, but from 1 to ∞ in the other case, leading to hard comparisons between positive and negative effects. We instead introduce the *Symmetric Relative Influence* index to harmonize the measurement of each feature f relative influence, regardless of its positiveness or negativeness: $SRI_f = \frac{p_{o,ci}}{p_{f,ci}} + \frac{p_{f,ci}}{p_{o,ci}}$.

By combining the previously described contributions, the *Perturbation Influence Relation* can be defined as:

$$\begin{aligned} PIR_f &= \Delta\, I_f * SRI_f \\ &= p_{f,ci} * \beta - p_{o,ci} * \alpha \\ \text{with } \alpha &= 1 - \frac{p_{o,ci}}{p_{f,ci}}\,,\ \beta = 1 - \frac{p_{f,ci}}{p_{o,ci}} \end{aligned} \tag{1}$$

The coefficient α represents the contribution of the original input w.r.t. the perturbed one and, similarly, β represents the contribution of the perturbation of feature f w.r.t. the original input. The PIR, which ranges in the $(-\infty, +\infty)$ interval, is finally normalized in the $[-1; 1]$ range exploiting the common *Softsign* function, leading to the definition of the normalized Perturbation Influence Relation ($nPIR_f$):

$$nPIR_f = softsign(PIR_f) \tag{2}$$

Where:

$$softsign(x) = \frac{1}{1 + |x|} \tag{3}$$

Combining the previous equations, the formal definition of nPIR for a given class of interest ci is provided in Eq. 4, which avoids the Eq. 1 problem of being undefined for $p_f = 0$ and $p_o = 0$.

$$nPIR_f = \begin{cases} \frac{p_o^3 - p_o^2 p_f + p_o p_f^2 - p_f^3}{p_o^3 - p_o^2 p_f + p_o p_f + p_o p_f^2 - p_f^3}, & \text{if } p_{o,ci} > p_{f,ci} \\ \frac{p_o^3 - p_o^2 p_f + p_o p_f^2 - p_f^3}{p_f^3 - p_o p_f^2 + p_o p_f + p_o^2 p_f - p_o^3}, & \text{if } p_{o,ci} < p_{f,ci} \\ 0, & \text{if } p_{o,ci} = p_{f,ci} \end{cases} \tag{4}$$

The normalized Perturbation Influence Relation captures both the amplitude and the relative impact. Experimental results, as reported in Sect. 10.8, show that human-

appreciated features with a strong positive influence are characterized by nPIR greater than 0.75, whereas the negative-influence threshold is around -0.2.

5.3 Normalized Perturbation Influence Relation Precision–*nPIRP*

The second aspect that we want to measure is if an interpretable feature influenced the prediction of only one or several class labels. The wider the range of classes impacted by an interpretable feature, the less that feature can be considered focused on the class of interest: the model has not learned the concept/pattern associated with that feature as precisely relevant only to the class of interest. This behavior can bring to light possibly misleading knowledge, such as training bias or bad network design.

To this aim, we introduce the *normalized Perturbation Influence Relation Precision* (nPIRP) index to evaluate the precision of the absolute impact of f over ci (component ξ_{ci}) w.r.t. the sum of the positive impacts over classes $C \setminus ci$ (component $\xi_{C\setminus ci}$), which are defined as:

$$\xi_{ci} = p_{o,ci} * |nPIR_{ci}| \tag{5}$$

$$\xi_{C\setminus ci} = \sum_{c}^{C\setminus ci} p_{o,c} * max(0, nPIR_c) \tag{6}$$

The two measurements of influence are weighted by the probability of the original image to belong to each class so that the influences of the most probable classes are taken into greater consideration w.r.t. the influences obtained on less probable outcomes.
Then, the Perturbation Influence Relation Precision of a feature f is defined as:

$$\begin{aligned} PIRP_f &= \Delta I_f(\xi_{ci}, \xi_{C\setminus ci}) * SRI_f(\xi_{ci}, \xi_{C\setminus ci}) \\ &= \xi_{C\setminus ci} * b - \xi_{ci} * a \\ \text{with } a = 1 - \frac{\xi_{ci}}{\xi_{C\setminus ci}}&, b = 1 - \frac{\xi_{C\setminus ci}}{\xi_{ci}} \end{aligned} \tag{7}$$

By following similar reasoning to that of nPIR, we can normalize Eq. 7 exploiting the *softsign* function to obtain the Normalized Perturbation Influence Relation Precision index:

$$nPIRP = softsign(PIRP) \tag{8}$$

As nPIRP is computed for each feature and ranges in $[-1; 1]$. When f is very precise on describing ci, the nPIRP has values close to 1. When f is impacting more other classes $C \setminus ci$ than the class of interest ci, the index value is close to -1. When nPIRP is close to 0, f impacts similarly the class-of-interest and other classes as well.

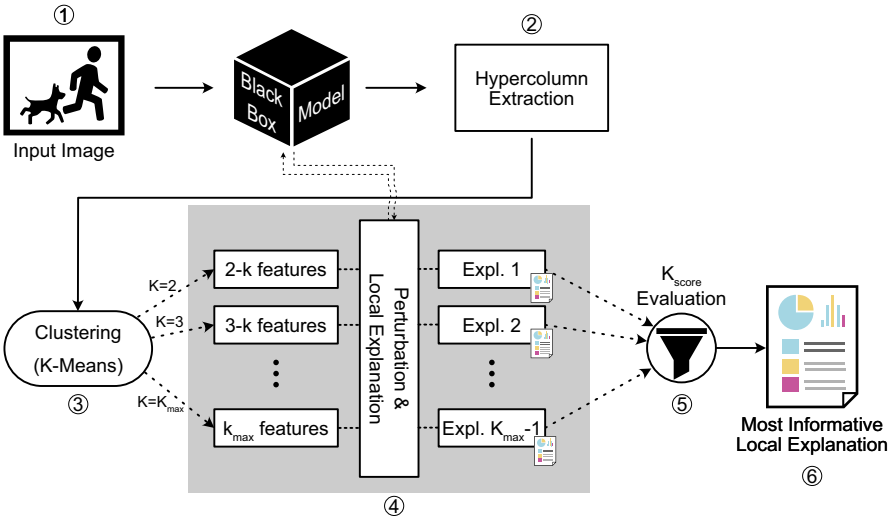


Fig. 6 Unsupervised interpretable features extraction details

6 Most informative local explanation

As introduced in Sect. 4, EBANO implements an iterative process where different k divisions are analyzed and evaluated to find the best cluster partitioning. We define the local explanation produced by the best k partitioning as the *most informative local explanation*.

Figure 6 shows in detail all the steps performed by EBANO to find the *most informative local explanation*. Firstly, as discussed in Sect. 4.1, given an input image ①, the hypercolumns are extracted from the DCNN model ②. Then, the unsupervised analysis of the hypercolumns is performed to extract a set of interpretable features with the *K-Means* algorithm ③. However, it is impossible to know the best number of features to extract from the input in advance. For this reason, EBANO produces the explanations for different possible divisions in range $K = [2, k_{max}]$, where k_{max} is a user-defined parameter of the local explanation. An important trade-off is to set k_{max} so that the number of extracted features f is small enough to be manually inspected by end-users but large enough to avoid missing details and diversity. More precisely, for each value of $k \in K$, EBANO extracts $f = k$ interpretable features by clustering the hypercolumns extracted before. Then, for each feature, it performs the perturbation (as discussed in Sect. 5.1) and extracts from the model the new probabilities (of each new perturbed image). Finally, it produces, for each feature of each $k \in K$, the *influence* and the *precision* indices, respectively *nPIR* and *nPIRP* (as discussed in Sects. 5.2 and 5.3). Even if the end-user can query all possible k divisions of the images and their relative explanations, EBANO is able to automatically suggest the *most informative local explanation* (i.e., the best k division among all possible k analyzed). The *most informative local explanation* is defined as the one maximizing the *contrast* of

the perturbation-influence-relation values (nPIR index defined in Sect. 7) between all their features.

Formally, for each $k \in [2, k_{max}]$, it is provided an explanation e_k composed by a set of interpretable features F_k (with k features). Then, for each feature $f \in F_k$, the corresponding influence e precision indices (nPIR and nPIRP) are computed. Finally, for each explanation e_k produced by each k partitioning, it is assigned a score by analyzing the different influence (nPIR) scores obtained by their features, as follows:

$$K_{score}(e_k) = \max_{f \in F_k} \left(nPIR_f(e_k)\right) - \min_{f \in F_k} \left(nPIR_f(e_k)\right) \tag{9}$$

In other words, the score assigned to each k possible division is equal to the difference between its most influential and its least influential feature. Once computed the score for each possible k, let E be the set of all e_k explanations produced with the different k values evaluated. The *most informative explanation* $\hat{e}$ is the one maximizing the K_{score}, as follows:

$$\hat{e} = \max_{e_k \in E} \left(K_{score}(e_k)\right) \tag{10}$$

The *most informative explanation* $\hat{e}$ is the one proposed to the end-user as the best explanation (even if it is possible to query all the others k divisions produced). The score function proposed in EBANO, defined by Eqs. 9 and 10, could be changed by the final user according to specific needs.

7 Local explanation

Comparing the probabilities before and after each perturbation, EBANO is able to study the influence of each *interpretable feature* on a specific predicted class, producing a *local explanation* with a *numerical* contribution and a *visual* part. We firstly discuss the numerical and *visual* parts of the *local explanation* produced by EBANO (Sects. 7.1 and 7.2). Then, we discuss in detail the local explanations produced for the running example (Sect. 7.3).

7.1 Numerical explanation

As discussed in Sect. 5, we introduce two indices (nPIR and nPIRP), allowing the user to objectively inspect the details of the prediction process. Differently from other works (Lundberg and Lee 2017; Ribeiro et al. 2016), our indices (i) efficiently measure the influence relation that exists between the input feature and the model outcomes in terms of neutral, positive or negative impact, and (ii) they also consider the influence precision of the features for the class-of-interest in a multi-class problem. Precise features are very focused, affecting only a specific predicted class. Low-precision features, instead, can affect many classes at the same time.

We recall that the influence sign and amplitude of a feature over the class-of-interest, quantified by the nPIR index, can be considered:

– *positive* if the predicted probability decreases after the perturbation, meaning that the feature was positively relevant for the model;
– *neutral* if the predicted probability remains the same after the perturbation, meaning that the feature was irrelevant for the model;
– *negative* if the predicted probability increases, meaning that the feature was negatively relevant for the model.

Also, the distribution of the probabilities for the other classes can change accordingly to the perturbed feature. The precision of influence of a feature, quantified by the nPIRP index, can be considered:

– *precise* if the class-of-interest is the only one affected by the perturbation process of the feature;
– *not precise* if the perturbation of the feature is equally affecting the class-of-interest and at least another class;
– *negatively precise* if the perturbation of the feature is affecting more any of the other classes other than the class-of-interest.

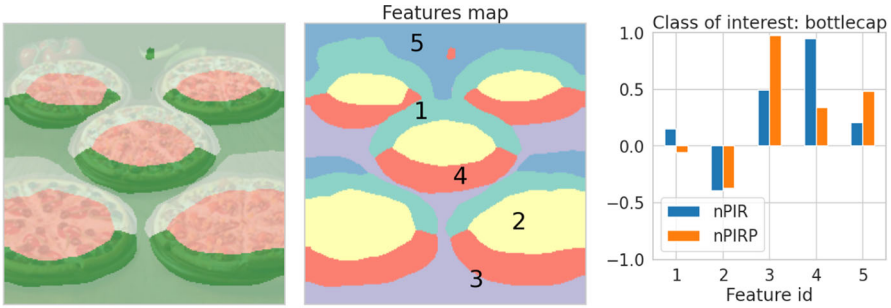
The two indices computed for each feature over a class of interest compose the *numerical explanation* part of the local explanation.

7.2 Visual explanation

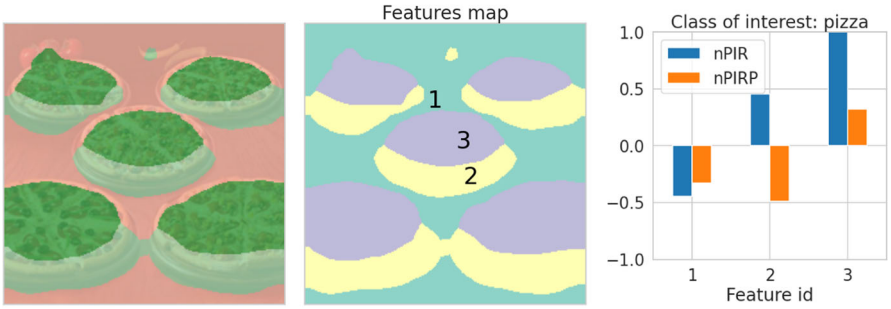
Along with the detailed quantitative explanation, our explanation framework provides an easy-to-understand prediction-local visual explanation, where, each interpretable feature is colored with a red-green gradient according to the value of nPIR. The more a green area is intense, the more the corresponding feature is positively influential for the class-of-interest. On the contrary, the more a red area is intense, the more the feature is negatively impacting the class-of-interest. White areas instead, which results almost transparent, show input portions that have a neutral impact on the prediction process (i.e., the model is completely independent of the presence of these features). Differently from other works (Selvaraju et al. 2019; Simonyan et al. 2014; Fong and Vedaldi 2017; Zhang et al. 2018; Petsiuk et al. 2018), the proposed visualization is not based on saliency maps. A saliency map is a simple and clear visualization strategy that smoothly shows the relevance of contiguous areas of pixels. However, it does not allow to differentiate the influence of multiple input areas at the same time. Instead, EBAnO can highlight the impacts of more input regions simultaneously, with their positive and negative contributions, including more information in a single visual representation (as shown in the example in Sect. 7.3).

7.3 Example of local explanation

As discussed before, for the input image of the running example in Fig. 2, are produced the output probabilities by a pre-trained VGG16 DCNN model shown in Table 1. The model predicts the wrong *Bottlecap* class as most probable, followed by *Pizza* with probabilities respectively 0.42 and 0.28 (even if the ground truth label was pizza). In this case, an end-user that wants to analyze the reasons for the wrong behavior of the



(a) Explanation of prediction *Bottlecap* (VGG16 in Table 1).



(b) Explanation of prediction *Pizza* (VGG16 in Table 1).

Fig. 7 Example of local explanations for two classes of interest and the image shown in Fig. 2. Each explanation is organized with the Visual explanation (left), the map of features (center), the quantitative explanation (right). (**a**) is the explanation for the *Bottlecap* class label, (**b**) for the Pizza class label

DCNN model in the predictions of the class label of the input image can inspect the *local explanation* produced by EBANO. For instance, in this case, it can be useful to produce the local explanation for both *Bottlecap* and *Pizza* classes of interest (as shown in Fig. 7). For both explanations, are provided the features map (center), the *visual explanation* (left), and the *numerical explanation* (right).

For the explanation of the *Bottlecap* class label (the most probable class predicted by the model) 7a, EBANO finds the division with 5 interpretable features as the *most informative* local explanation. From this local explanation, it emerges that the parts of the image the most responsible for the wrong prediction of the model are the pixels corresponding to the pizza borders (feature 4) with an nPIR (influence) index close to 1 and a positive nPIRP (precision). Moreover, the features corresponding to the table's pixels positively impacted the prediction of the *Bottlecap* class label, even if with a lower amplitude. Specifically, the table's pixels are divided into two interpretable features based on the amplitude of their influence on the prediction. Therefore, the corresponding pixels are colored in light green in the visual explanation (with higher intensity on the pixels of feature 3 because it obtained a higher influence score). Finally, the feature composed by the upper borders of pizza (feature 1) has a very small positive impact on the prediction, obtaining an nPIR score close to 0.1.

Instead, for the explanation of the *Pizza* class label, EBANO finds the partitioning with 3 interpretable features as the *most informative local explanation* 7b. The interpretable feature composed of the pizza topping's pixels highly positively impacted the prediction of the *Pizza* class label. Indeed, the corresponding pixels are highlighted in dark green in the visual explanation, and the influence score (nPIR) in the numerical explanation is close to 1. Moreover, the second feature, composed of the pizza crust's pixels, positively impacted the prediction (even if less than the first one). Therefore, it is highlighted in light green with a corresponding influence index close to 0.5. However, this feature is not precise because the perturbation of their pixels impacted not only the *Pizza* label but also other labels, obtaining a negative precision index (nPIRP). Finally, the third interpretable feature shows that the pixels of the table negatively impacted the original prediction of *Pizza* and is highlighted in light red with an influence index close to -0.5.

The conclusion that an end-user can draw, thanks to the detailed explanations provided by EBANO, is that the model's prediction is unreliable because it mostly looks from the context (table's pixels) to predict the class label. Moreover, the pizza borders are uncertainly captured by the model that assigns its pattern to different labels (*Bottlecap* and *Pizza*).

8 Class-based model explanation

A model-global explanation is usually exploited to study the influence of a specific concept on the whole prediction set provided by the model to detect possible bias, for instance. In data domains like tabular data or textual data, the explanation process takes advantage of well-defined features, i.e., columns and word tokens, that are simple to aggregate in model-global explanations. Works like Lundberg and Lee (2017); Ribeiro et al. (2016) study the behavior of the model aggregating the explanations produced for singular predictions by feature meaning.

In the case of image inputs, instead, the DCNN model processes each pixel. As previously discussed, single-pixel explanations are useless for humans. Hence, EBANO groups prediction-local explanations according to *interpretable features*. Analyzed together, nPIR and nPIRP describe the influence, in terms of both the contribution and the precision, of each *interpretable feature* of an input image on the prediction process. This information enables EBANO to identify behavioral patterns of the model w.r.t the prediction of each class.

The model-wise challenge is to aggregate the interpretable features belonging to different images by their semantic meaning without using another supervised model. To this aim, EBANO provides an unsupervised class-based model explanation by aggregating the prediction-local features according to their class-of-interest. Then, each class-of-interest is described by all the features extracted during the local-explanation process, exploiting their nPIR and nPIRP values. The features are projected on the $nPIR \times nPIRP$ space, and studying their distribution allows the user to inspect the class-wise behavior of the model during the decision-making process.

Figure 8 shows an example of a class-based model explanation for the class-of-interest *Pizza* computed for a VGG16 model, aggregating three local explanations of

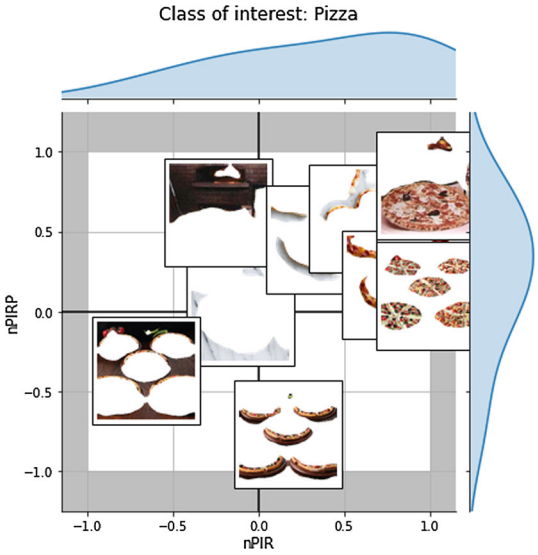


Fig. 8 Class-based model explanation example for class-of-interest *Pizza*

three different input images. The figure shows the interpretable features distributed in the $nPIR \times nPIRP$ space and their KDE distributions (Kernel Density Estimation) on the nPIR and nPIRP axis. The plot groups the features in the four quadrants. For instance, features being both positively influential and precise for the class-of-interest are in the quadrant with $nPIR \geq 0$ and $nPIRP \geq 0$. On the top and right axis, the KDE distributions of the features w.r.t. nPIR and nPIRP are reported.

The *optimal distribution of features* for a model is when all the features that are representative for the class-of-interest are positioned on the top-right corner with $nPIR = 1, nPIRP = 1$, and all the other features are close to the center with $nPIR = 0, nPIRP = 0$ so that the contextual features are not influencing the decision-making process. The presence of features spread around the plot means that the model can be considered uncertain about their role in the prediction process. The plot easily enables human experts to quickly drive their evaluation towards specific features for a semantic assessment of the model behavior.

9 EBANO- BATCH

In some specific scenarios, the time required to produce an explanation could be a bottleneck. Specifically, the execution time could become an issue when multiple explanations of several images are needed. An example of this situation is the global explanations, where multiple local explanations for different images predicted with the same class-of-interest should be produced. An iterative approach that produces the explanation of single image instances, one at a time, is not efficient. For this reason, we also propose a batch version of the framework called EBANO- BATCH.

EBANO- BATCH takes as input a set of images and classes of interest (instead of a single image and a class of interest) and outputs the local explanations for the entire

set, saving them persistently on disk. In EBANO- BATCH several steps of the pipeline are vectorized and optimized for an entire set of images to speed up the computation, increasing the overall efficiency of the methodology. Specifically, the hypercolumns of the entire batch are extracted in one forward pass of the DCNN, the probabilities predicted after the perturbation of each interpretable feature of all images are produced in batch, and the dimensionality reduction with the PCA is optimized for the entire batch.

Exploiting EBANO- BATCH, we reduced on average by a factor of 10 the time required to produce an explanation (by testing several batch sizes in [8, 16, 32, 64, 128] and several models).

10 Experimental evaluation

This section is structured as follows. Sect. 10.1 describes the experimental settings. Sections 10.2 and 10.3 present the prediction-local and the class-based global explanations, respectively. Sections 10.4 and 10.5 discuss the performance of the nPIR index in identifying the contribution of each feature in the prediction process, first across all the tested images and models, and then compared to the state-of-the-art *Shapley values*, respectively. Then, Sects. 10.6 and 10.7 qualitatively and quantitatively compare EBANO with two state-of-the-art explanation frameworks. Section 10.8 reports the human-validation process carried out to assess the interpretability of EBANO's explanations. Finally, Sect.10.9 performs a brief execution time comparison.

10.1 Experimental settings

To show the effectiveness and the reliability of the framework, EBANO has been tested on 4 different pre-trained DCNN models available in the Keras deep learning library (Chollet et al. 2015)[5]: (M1) VGG16 (Simonyan and Zisserman 2015), (M2) VGG19 (Simonyan and Zisserman 2015), (M3) InceptionV3 (Szegedy et al. 2016), and (M4) InceptionResNetV2 (Szegedy et al. 2017). All the models are pre-trained on the well-known ImageNet (Russakovsky et al. 2015) dataset with 1000 classes. EBANO has been applied to produce prediction-local explanations for the 4 different models using 250 input images, belonging to 54 different classes. The top-10 predicted classes of each image have been analyzed, for a total of 10,000 prediction-local explanations. The input images have been taken from different datasets (Coco (Lin et al. 2014), ImageNet (Russakovsky et al. 2015), Caltech (Li Fei-Fei et al. 2004), and web scraping).

The number of convolutional layers analyzed for each model has been experimentally set as follows. Models M1 and M2 are relatively small DCNNs and the last 5 and 8 convolutional layers, respectively, have been considered. Instead, models M3 and M4 have a more complex structure and the last 34 and 24 convolutional layers have been included in the analysis, respectively. We found these settings to be a fair trade-off between feature interpretability and affordable execution complexity.

[5] Keras version 2.2.4.

Fig. 9 *Mouse* input image (I1)

The number of extracted features has been set to range between 2 and 10. The upper limit prevents too small features with poor semantic meaning and low relevance for a human user to be extracted. Thus, each explanation will be described at most by 10 features, among which the *most informative explanation* (Sect. 6) is automatically proposed to the user, with the others available for further manual insights.

10.2 Prediction-local explanations

In this section we discuss in detail the insights provided by the local explanations of EBANO. We exploit models M1 and M4 for the discussion of the experimental results since they are representative architectures of the two remaining models as well. However, in "Appendix C", a further selection of prediction-local explanations is discussed in detail, on the results of models M2 and M3. Moreover, in "Appendix D", some other examples of local explanations for correctly classified images are reported. Finally, a larger number of prediction-local explanations of all four models is publicly available through an interactive web-based tool.[6]

Figure 9 shows an example input image (named I1) showing a mouse over a tailed surface.

The predictions of M1 and M4 are shown in Table 2 and 3, respectively. By applying EBANO to such predictions we aim to unwrap the black-box models M1 and M4, providing detailed explanations to answer the following questions:

- *Q1.* "Why is Fig. 9 representing a *Toilette seat* for model M1?"
- *Q2.* "Why is Fig. 9 not a *Mouse* for model M1?"
- *Q3.* "Why is Fig. 9 a *Mouse* for model M4?"

Answering Q1. Model M1 (VGG16) fails to predict the correct class for input I1, providing the label *Toilet seat* with the highest probability, and the correct class *Mouse* follows with lower probability. Figure 10a shows the explanation provided by EBANO

[6] https://ebano-ecosystem.github.io/#explanation-library.

Table 2 VGG16 (M1) predictions for Fig. 9

Class	$P(c)$
Toilet seat	0.23
Mouse	0.15
Soap dispenser	0.11
Washbasin	0.11
Can opener	0.06

The ground-truth label is in bold (Mouse)

Table 3 InceptionResNetV2 (M4) predictions for Fig. 9

Class	$P(c)$
Mouse	0.99
Mousetrap	0.00
Toucan	0.00
Joystick	0.00
Computer keyboard	0.00

The ground-truth label is in bold (Mouse)

for model M1 and the class of interest *Toilet seat*. It identifies the most informative explanation to consist of 9 features (Fig. 10a center). Figure 10a-left shows the visual explanation and Fig. 10a-right shows the numerical explanation.

We recall that the visual explanation highlights in green the interpretable features positively influencing the class-of-interest, while in red the negatively ones. We notice that the decision of assigning the class *Toilet seat* is mainly due to the presence of the horizontal lines of the background tiles. Hence, the prediction *Toilet seat* has been taken because of contextual information and not because of the subject itself. Based on the numerical explanation, where we recall that the bar chart reports the values of nPIR (influence) and nPIRP (precision) for each feature, we confirm that feature 6, corresponding to the lines of the tiles, is the most positively influencing. However, its nPIRPvalue is close to 0, meaning that it is not precise at all: it is a contextual feature similarly influencing also many other classes.

Answering Q2. Figure 10b shows the explanation produced by EBAnO for model M1 and for the class of interest *Mouse*, with 4 interpretable features. From the visual explanation, we notice that the *Mouse* is correctly associated with feature 1. However, (i) feature 4 is strongly affecting the *Mouse* prediction (very negative nPIR), and (ii) the nPIRPof feature 1 is negative, as highlighted by the numerical explanation in Fig. 10b-right. The case of positive nPIRand negative nPIRP(feature 1) describes the behavior of the DCNN model: even if feature 1 is positively influencing the (correct) class, other classes are more influenced by this feature w.r.t. the one under analysis. we can blame feature 4 (portions of the background tiles), with its negative nPIRand negative nPIRP, for the incorrect prediction. The decision process was mainly affected by the context of the subject: the model correctly distinguishes among the different sections of the input, but it evaluates the context more than the subject. Furthermore, the influence of different features on more than one class means that possibly the training set contained

some bias for this subject in the given context (e.g., toilet objects often represented with background tiles).

Answering Q3. Model M4 (InceptionResNetV2) correctly classifies the image as a *Mouse*. The EBANO explanation is reported in Fig. 10c with 5 features. Feature 1, the mouse body, is the only feature with high influence and precision, whereas all other features are neutral. Differently from model M1, M4 is well focused on the subject, ignoring the context around it. Its predictions can be considered generally reliable and it can be trusted with higher confidence than M1: it is not by chance that the prediction is correct.

10.3 Class-based explanations

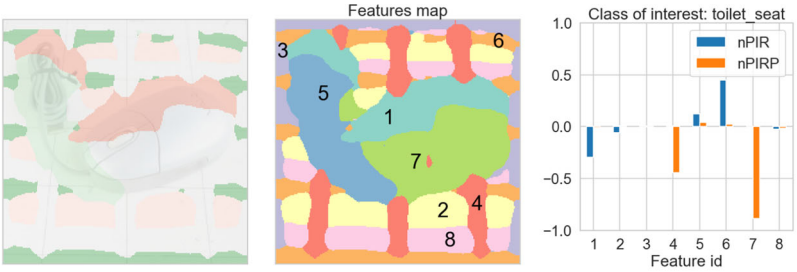
Figure 11 shows the results of a class-based model explanation computed on 50 images classified as *Dalmatian* by models M1 and M4 (further considerations about models M2 and M3 in "Appendix E"). While the features of M1 are scattered across the whole $nPIR \times nPIRP$ area (Fig. 11a), M4 presents tidier patterns (Fig. 11b). Recalling the *optimal distribution of features* introduced in Sect. 8, the class-based explanation of M1 shows a significant uncertainty in the prediction process of the class *Dalmatian*. Instead, M4 features are concentrated mostly in $0 \leq nPIR \leq 1$ and $nPIRP \approx 1$. In details, we notice that M4 contextual features are in general in the $0 \leq nPIRP \leq 1$, $nPIR \approx 0$ area. Such explanation describes a much more reliable prediction process of M4 for the class of interest *Dalmatian*, with the model assigning a much clearer role to each feature. This result is also coherent with the state-of-the-art knowledge: M4, i.e. InceptionResNetV2 is known in the literature as a much more accurate and reliable model w.r.t. M1, i.e. VGG16.

We finally note that EBANO highlights the model uncertainty in a totally unsupervised approach: it does not know the ground truth labels, and often they are not available. Hence, the class-based model explanations are not only widely applicable, but they also empower the end-user to better choose the model to trust based on easily readable and visual information. Moreover, exploiting EBANO- BATCH, they are produced more efficiently and quickly, explaining entire batches of images at a time.

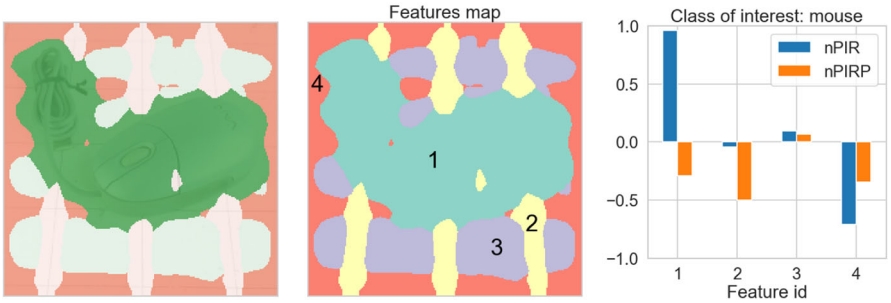
10.4 Feature-relevance index assessment

To provide a wide analysis of the behavior of the proposed nPIR index w.r.t. the different DCNN models across all the 250 images of the experimental set, the distributions of the nPIR minimum and maximum values have been computed and reported in Fig. 12. Such values represent the most informative explanations computed by EBANO when the class of interest is equal to the top-1 prediction. If the difference between the minimum and the maximum nPIR distributions is large, then we can support the wide applicability of the proposed approach, besides the limited number of examples reported in the experimental results due to the space constraints.

The top influential features, with maximum nPIR, are mostly included in the [0.8, 1.0] bin. Only models M3 and M4 have some features of the top influential ones falling in the [0.0, 0.2] bin. However, the value of maximum nPIR never goes



(a) Explanation of prediction *Toilette seat* (VGG16 in Table 2).



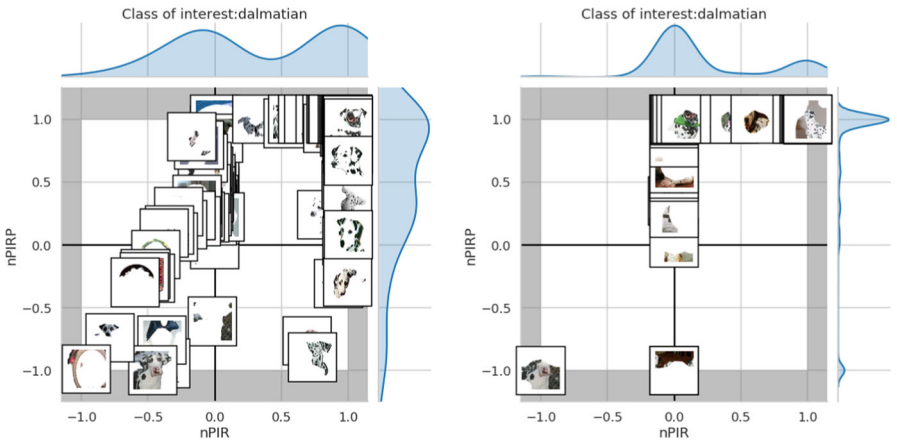
(b) Explanation of prediction *Mouse* (VGG16 in Table 2).

(c) Explanation of prediction *Mouse* (InceptionResNetV2 in Table 3).

Fig. 10 EBANO local explanations. The input image is shown in Fig. 9. Visual explanation (left), features (center), numerical explanation (right)

below 0 for any model, hence EBANO is always able to identify at least one positively influential feature.

The features with minimum nPIR are predominantly located in the $[-0.2, 0.2]$ range, meaning that most of the less influential features are from slightly negative to almost neutral for the prediction process. Minimum values higher than 0.2 are very rare, confirming the large distance from the minimum and the maximum nPIR values, which drives the right choice of the most informative explanation by EBANO.



(a) Model M1. (b) Model M4.

Fig. 11 Class-based model explanation for the *Dalmatian* class on 50 input images

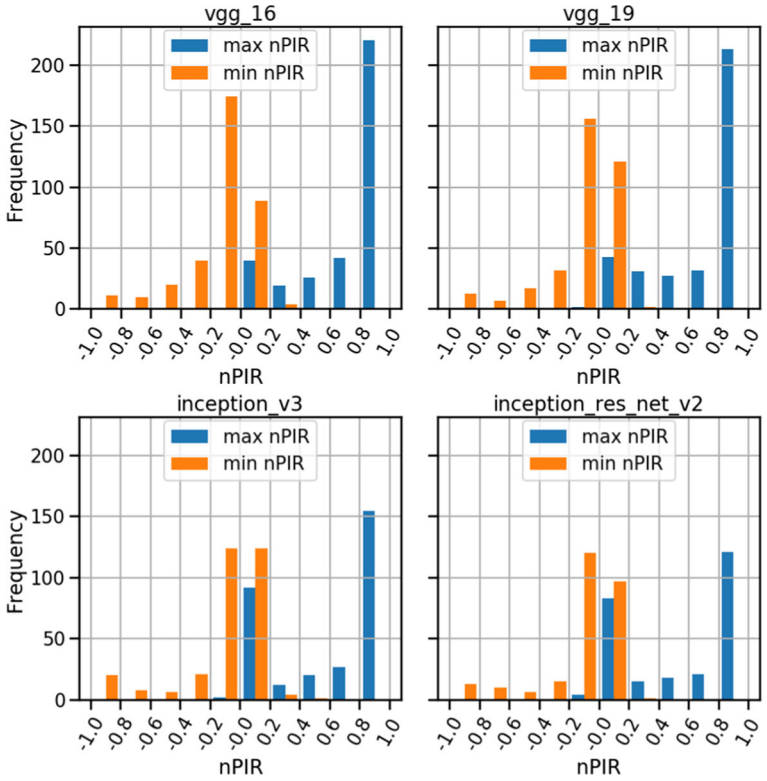


Fig. 12 For each tested model the distributions of min and max nPIR values obtained by the prediction-local explanations with class-of-interest equal to the top-1 predicted class for each input image

(a) Model M1 with class-of-interest *Toilet seat*.

(b) Model M1 with class-of-interest *Mouse*.

(c) Model M4 with class-of-interest *Mouse*.

Fig. 13 Comparison between nPIR index and *Shapley values*

10.5 Feature-relevance index comparison

In the state-of-the-art (see Sect. 2), the *Shapley values* (Štrumbelj and Kononenko 2014; Lundberg and Lee 2017) are widely used to explain the relevance of each feature in the decision-making process. *Shapley values* compute the effect of all the permutations of the possible *societies* of features, i.e., all the possible combinations of portions (features) of the input images, being computationally expensive.

Figure 13 shows the comparison between nPIR and *Shapley values* for the input image in Fig. 9. we analyze (a) model M1 for the class *Toilet seat*, (b) model M1 for the class *Mouse*, and (c) model M4 for the class *Mouse*. In Fig. 13a *Shapely values* have an almost flat trend with only feature 1 showing a slightly negative value, whereas nPIR greatly amplifies the different contributions of each feature, it better highlights the negative impact of feature 1 and it shows more clearly the positive impacts of features 5 and 6.

In Fig. 13b the nPIR is more effective than *Shapley values* for the explanation task. Feature 1 is identified by both indices as positively influencing, but nPIR marks it more prominently. Feature 4 is negatively influential, as correctly highlighted by nPIR, whereas *Shapely values* miss this contribution, being slightly positive.

In Fig. 13c, both indicators show the same behavior, with a positive contribution of feature 1 and a neutral contribution of all the other features.

To sum up, the proposed nPIR index is more computationally efficient and better emphasizes the contribution of the different input features, being always equal to or better than the state-of-the-art *Shapely values* in all the experiments performed.

10.6 Local-explanation qualitative comparison

In this section, the local explanations of EBAnO are qualitatively compared with those provided by state-of-the-art techniques: *LIME* (Ribeiro et al. 2016) and *Grad-CAM* (Selvaraju et al. 2019), as representative of *perturbation-based* and *gradient-based* explainability families, respectively. Discussions on model M1 (i.e. VGG16) explanations are reported in this section, and further results on model M4 are reported in "Appendix F".

Fig. 14 Input image (I2)

Table 4 VGG16 (M1) predictions for Fig. 14

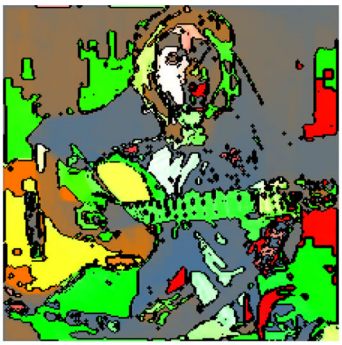
Class	$P(c)$
Acoustic guitar	0.22
Electric guitar	0.09
Golden retriever	0.06
Stage	0.04
Sussex spaniel	0.03

Figure 14 shows the input image I2 taken from Ribeiro et al. (2016) as a comparison example. M1 predicts *Acoustic Guitar* with a probability of 0.22 (see Table 4). The visual explanation of *LIME* is reported in Fig. 15a, and the *Grad* explanation is in Fig. 15b. *LIME* highlights in green the areas that are important for the class of interest, and in red those negatively impacting the prediction. Instead, *Grad-CAM* uses warm colors (e.g., red) for the most important areas and cold colors (e.g., blue) for the least important portions.

The explanation provided by *LIME* is quite confusing, due to the presence of many small green portions that are difficult to be interpreted or associated with a concept of the image. we notice that *LIME* performs the segmentation of the image without exploiting the knowledge contained in the network.

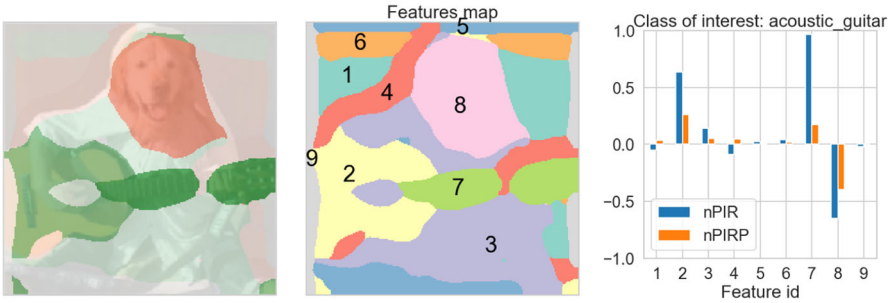
Grad-CAM is more precise in identifying the area of interest around the neck of the guitar, ignoring the background areas that were identified by *LIME* as important. However, *Grad-CAM* loses the information about the portions of the input that are negatively impacting the prediction. We notice that *Grad-CAM* extracts the information provided by the last convolutional layer of the network.

In general, none of the two state-of-the-art methods propose a human-readable numerical explanation of the prediction. In details, the explanation provided by EBAnO (Fig. 15c) (i) accurately identify concept-wise portions of the input that are responsible for the model's outcome (feature map in Fig. 15c-center), (ii) highlight both the positively-influential portions and the negatively-influential ones for each

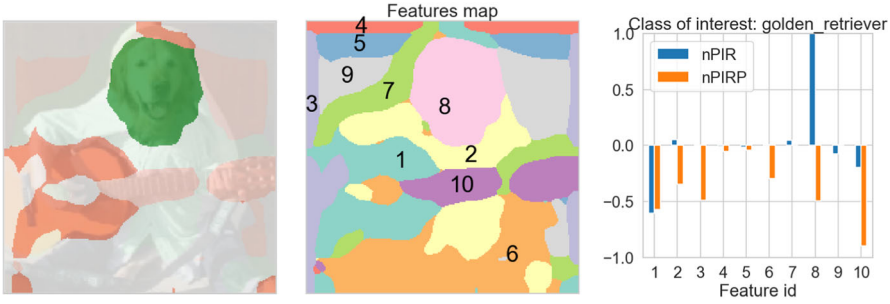


(a) Explanation of the prediction *Acoustic guitar* with *LIME* (in green).

(b) Explanation of prediction *Acoustic guitar* with *Grad-CAM*.



(c) Explanation of prediction *Acoustic guitar* with EBANO.



(d) Explanation of prediction *Golden retriever* with EBANO.

Fig. 15 EBANO local explanations. The input image is shown in Fig. 14. Visual explanation (left), features (center), numerical explanation (right)

class of interest (visual explanation in Fig. 15c-left) and (iii) quantify not only the influence but also the precision of each image portion with numerical explanations (Fig. 15c-right).

Regarding the example under analysis, EBANO identifies the image portions corresponding to the guitar as very influential with high nPIR values and positive nPIRP values, while, the face of the dog has been correctly identified as negatively impacting the class *Acoustic Guitar*.

For completeness, the EBANO's explanations for the class *Golden Retriever* have been provided in Fig. 15d, showing that the guitar has now a very negative impact and the dog face is identified as playing the main role in the decision-making process.

10.7 Local-explanation quantitative comparison

In this section, we quantitatively evaluate and compare the explanations produced by EBANO, *LIME* (Ribeiro et al. 2016), and *Grad-CAM* (Selvaraju et al. 2019) exploiting two quality measures: *Pointing Game* and *Pixels Flipping* (Samek and Müller 2019). For these evaluation tasks, we created a new dataset, consisting of 150 randomly selected images from the validation set of ImageNet Russakovsky et al. (2015) with the relative ground truth labels and bounding boxes.[7]

To evaluate EBANO, we selected the most positively influential feature for the predicted label (i.e., the one with max $nPIR$). For LIME, since its features have smaller dimensions, to make a fair comparison, we selected the combination of the *top-n* positive features for different $n \in [5, 10, 15, 20, 25, 50, 75, 100]$ as the most important feature. For instance, for *LIME-10* the most important feature is the combination of the top-10 features. Finally, GradCAM assigns an importance score (only positive) to each pixel based on the values of the activation map. Thus, we selected the combination of the *top-n percentiles* of pixels with higher activation values as the most important feature. We experimented with several percentiles of the non-zero activation values $top_{perc} \in [5, 10, 25, 50, 75]$. For instance, the feature for *GradCAM-5* is computed, for each image, by selecting the $perc_{95}$ as the *95-percentile* of the non-zero activation values, and taking the pixels with activation $\geq perc_{95}$ (i.e., top-5 percentile of activation values). For each model and task, we report in the paper and discuss the features with sizes similar to those selected by EBANO. However, full results for all combinations are reported in "Appendix G".

10.7.1 Pointing game

Firstly, we compared the most relevant regions found by the explanation methods, by measuring how much they lie on the object of the predicted label. This evaluation task measures how much the explanations are able of localizing target objects, exploiting the ground truth bounding boxes. This measure assumes that a well-trained classifier will mostly focus on the object to make the prediction, at least for correct predictions. Thus, reliable explanations for these examples should highlight important features in the target object bounding boxes. However, this is not always true because a model

[7] ILSVRC2012 https://image-net.org/challenges/LSVRC/2012/2012-downloads.php.

could predict the right label but for the wrong reasons. Nevertheless, using well-trained state-of-the-art models (such as VGG16, VGG19, Inception V2, and Inception V3), this can be considered a valid evaluation metric. For this task, we kept only the correctly classified images for each model. Then, we selected, for each image, all and only the bounding boxes corresponding to the target object of the ground truth label. Finally, we compared the most important regions found by each method with the ground truth bounding boxes, measuring the percentage of pixels that lie in the target-object bounding boxes. Table 5 shows the mean number of pixels $\overline{N}_{px}$, the percentage of the pixels with respect to the total image size (target image size as input of the model) $\overline{\%}_{px}$, the mean number of feature pixels inside the bounding boxes of the target class $\overline{N}_{px}^{bbox}$, and the percentage of feature pixels inside the bounding box $\overline{\%}_{px}^{bbox}$. Experimental results show that, in general, smaller features tend to fall more easily into bounding boxes. However, with the same feature size, EBANO gets comparable or better results than *LIME* and *GradCAM* in terms of the percentage of feature pixels that lie in the bounding box for all the experimental models. As a result, the features identified by EBANO, at least for correctly classified examples, tend to be more precise in identifying regions within the target object as important features, under the assumption that, if the model is reliable, the correct prediction should mostly focus on such features to make the prediction (the second experimental task ensures that this is true).

10.7.2 Pixel flipping

The second evaluation task consists of measuring the model performance decline by occluding the most relevant regions. The idea is that the more precise and faithful the pixels selected by the explanation, the larger the performance decline should be. Precisely, we adapted the original pixel flipping experiment to our case, since the original version requires that individual pixels can be sorted by importance score to measure the gradual decline of the classifier by removing pixels from most important to least important. However, EBANO assigns the importance score to large features, making it difficult to sort feature pixels by importance (i.e., they are considered equally important if inside the same feature). Therefore, to make a fairer comparison, we selected, as much as possible, similar-sized features by creating a single most important feature as a combination of the most important features, as previously described at the beginning of the current Section.

Table 6 shows the results by using the Gaussian blur as pixels occlusion. We report and discuss the results with Gaussian blur since it is the least likely to introduce artifacts possibly cheating the network. However, full results with many occlusion types (i.e., introducing 0, 255, or the mean value) and for several feature sizes are available in "Appendix G". $\overline{N}_{px}$ is the mean number of pixels of the selected features, $\overline{\%}_{px}$ is the mean percentage of feature pixels with respect to the total number of pixels (of the target size of the model), Acc_o is the original accuracy of the model for the 150 images, Acc_p is the accuracy of the model on the perturbed images (i.e., by applying the Gaussian blur over the feature pixels), $\Delta Acc_{o,p}$ is the drop of accuracy computed as $Acc_o - Acc_p$, and $\overline{\Delta P}_{o,p}(\hat{c})$ is the mean probability decrease for the most probable

Table 5 Pointing Game

		EBAnO	*LIME-75*	*LIME-100*	*GradCAM-25*	*GradCAM-50*
M1	$\overline{N}_{px}$	16,284	14,138	16,164	8,826	17,495
	$\overline{\%}_{px}$	32.5%	28.2%	32.2%	17.6%	34.9%
	$\overline{N}_{px}^{bbox}$	12,106	7833	8617	7000	10,912
	$\overline{\%}_{px}^{bbox}$	78.3%	57.8%	55.9%	79.0%	62.7%
		EBAnO	*LIME-75*	*LIME-100*	*GradCAM-25*	*GradCAM-50*
M2	$\overline{N}_{px}$	15,600	13,821	15,845	8,604	17,062
	$\overline{\%}_{px}$	31.1%	27.5%	31.6%	17.1%	34.0%
	$\overline{N}_{px}^{bbox}$	11,695	7,850	8,624	6,889	10,715
	$\overline{\%}_{px}^{bbox}$	78.4%	59.8%	57.1%	78.8%	62.7%
		EBAnO	*LIME-10*	*LIME-15*	*GradCAM-25*	*GradCAM-50*
M3	$\overline{N}_{px}$	25,333	19,849	27,195	15,066	30,158
	$\overline{\%}_{px}$	28.3%	22.2%	30.4%	16.9%	33.7%
	$\overline{N}_{px}^{bbox}$	20,183	14,754	18,304	12,570	21,308
	$\overline{\%}_{px}^{bbox}$	81.2%	75.3%	67.7%	82.7%	70.4%
		EBAnO	*LIME-15*	LIME-20	*GradCAM-50*	*GradCAM-75*
M4	$\overline{N}_{px}$	30,402	27,735	34,516	26,261	39,370
	$\overline{\%}_{px}$	34.0%	31.0%	38.6%	29.4%	44.0%
	$\overline{N}_{px}^{bbox}$	23,195	18,592	21,313	21,720	28,267
	$\overline{\%}_{px}^{bbox}$	76.9%	67.4%	61.8%	81.9%	70.8%

Experimental comparison in *Pointing Game* experiments with M1 (VGG16), M2 (VGG19), M3 (Inception v3), and M4 (Inception ResNet v2). *LIME-n* means that the combination of the *top-n* important features is considered, while *GradCAM-n* means that the combination of the *top-n* percentile of the most important pixels is considered. $\overline{N}_{px}$ is the number of feature pixels, $\overline{\%}_{px}$ is the percentage of feature pixels w.r.t. the image size, $\overline{N}_{px}^{bbox}$ is the number of feature pixels inside the bounding box, $\overline{\%}_{px}^{bbox}$ is the percentage of feature pixels inside the bounding box w.r.t. the feature size

predicted class $\hat{c}$. Even when considering fewer pixels, the occlusion of the most important features of EBAnO causes a larger decrease of accuracy $\Delta Acc_{o,p}$ and a larger mean decrease of probability for the most probable predicted class $\overline{\Delta P}_{o,p}(\hat{c})$ than the occlusion of the features found by *LIME* and *GradCAM*, for all the models considered. The results obtained using the other types of occlusion are also similar, as shown in "Appendix G". This highlights that, probably, by mining the internal layers of the DCNN directly, EBAnO is more precise and faithful in extracting the effective features exploited by the model to make the prediction.

10.8 Human validation

Human beings are the main beneficiary of explanation frameworks like EBAnO and, for this reason, their comprehensiveness and effectiveness should always be validated

Table 6 Pixel Flipping

		EBAnO	*LIME-75*	*LIME-100*	*GradCAM-25*	*GradCAM-50*
M1	$\overline{N}_{px}$	15,427	14,222	16,481	8,444	16,747
	$\overline{\%}_{px}$	30.7%	28.3%	32.8%	16.8	33.4%
	Acc_o	0.77	0.77	0.77	0.77	0.77
	Acc_p	0.06	0.38	0.33	0.27	0.17
	$\Delta Acc_{o,p}$	**−0.71**	−0.39	−0.44	−0.50	−0.60
	$\overline{\Delta P}_{o,p}(\hat{c})$	**−0.76**	−0.47	−0.52	−0.56	−0.66
		EBAnO	*LIME-75*	*LIME-100*	*GradCAM-25*	*GradCAM-50*
M2	$\overline{N}_{px}$	14,805	14,410	16,617	8,374	16,633
	$\overline{\%}_{px}$	29.5%	28.7%	33.1%	16.7%	33.1%
	Acc_o	0.77	0.77	0.77	0.77	0.77
	Acc_p	0.03	0.45	0.40	0.30	0.16
	$\Delta Acc_{o,p}$	**−0.74**	−0.32	−0.37	−0.47	−0.61
	$\overline{\Delta P}_{o,p}(\hat{c})$	**−0.78**	-0.45	−0.49	−0.57	−0.67
		EBAnO	*LIME-10*	*LIME-15*	*GradCAM-25*	*GradCAM-50*
M3	$\overline{N}_{px}$	24,388	19,668	26,947	15,118	30,278
	$\overline{\%}_{px}$	27.3%	22.0%	30.1%	16.9%	33.9%
	Acc_o	0.85	0.85	0.85	0.85	0.85
	Acc_p	0.15	0.33	0.24	0.53	0.26
	$\Delta Acc_{o,p}$	**−0.70**	−0.52	−0.61	−0.32	−0.59
	$\overline{\Delta P}_{o,p}(\hat{c})$	**−0.78**	−0.58	−0.69	−0.45	−0.68
		EBAnO	*LIME-15*	*LIME-20*	*GradCAM-50*	*GradCAM-75*
M4	$\overline{N}_{px}$	29,280	27,156	34,117	26,436	39,603
	$\overline{\%}_{px}$	32.8%	30.4%	38.2%	29.6%	44.3%
	Acc_o	0.87	0.87	0.87	0.87	0.87
	Acc_p	0.09	0.26	0.21	0.25	0.13
	$\Delta Acc_{o,p}$	**−0.78**	−0.61	−0.66	−0.62	−0.74
	$\overline{\Delta P}_{o,p}(\hat{c})$	**−0.79**	−0.66	−0.70	−0.66	−0.76

Experimental comparison in *Pixels Flipping* experiment with M1 (VGG16), M2 (VGG19), M3 (Inception v3), and M4 (Inception ResNet v2). *LIME-n* means that the combination of the *top-n* important feature are considered, while *GradCAM-n* means that the combination of the *top-n* percentile of the most important pixels is considered. $\overline{N}_{px}$ is the number of feature pixels, $\overline{\%}_{px}$ is the percentage of feature pixels w.r.t. the image size, Acc_o is the original model accuracy for the original images and Acc_p for the perturbed images, $\Delta Acc_{o,p}$ is the accuracy drop caused by the feature occlusion, $\overline{\Delta P}_{o,p}(\hat{c})$ is the mean probability decrease of the predicted class

by including end-user feedbacks. To this aim, a publicly-available online survey has been conducted,[8] allowing people to assess the effectiveness and the easiness of understanding of the prediction-local explanations proposed by EBAnO. The main purpose of the survey is to validate how much EBAnO explanations are human understand-

[8] https://ebano-ecosystem.github.io/#ebano-survey.

able, and which explanation framework the user prefers among the following: EBANO, *LIME* (Ribeiro et al. 2016), and *Grad-CAM* (Selvaraju et al. 2019).

The explanations in the survey have been selected to be appreciated by a general audience, hence we included 12 images with correct class predictions, avoiding misleading behaviors of the models or complex concepts that would require detailed analysis or expert skills. At the time of writing, the survey has been completed by 60 people with different instruction levels: 25% bachelor, 38% master, 32% Ph.D., 5% other; and different age ranges: 18% between 19 and 24, 51% between 25 and 29, 30% with more than 30 years.

For each image, we asked the user two questions corresponding to two different evaluation tasks. An example is shown in "Appendix H".

The first task evaluates the reliability and understandability of the relevant portions of the visual explanation provided by EBANO in terms of positive (green) and negative (red) influence. The idea is that, at least for the correctly classified examples, if the explanations proposed are human-readable and reliable, they should be coherent with the human-judgment in which are the positively or negatively important features for the prediction, given the predicted label. Results of this task report that that green areas of the visual explanations provided by EBANO have been chosen to be *Important* for the class of interest 55% of the times, *Partially Important* 40% of the times, and only 5% of the times they have been considered *Not Important*. So, in 95% of the cases (considering *important* and *partially important* as coherent with the positively influential features of EBANO), the influential areas of the input (green areas) highlighted by EBANO are coherently characterizing the class of interest with the human-judgment. Similarly, red areas, when present, have been coherently considered *Not Important* for the target class of interest in the 69% of the answers.

The second evaluation task asks the user to select (multiple selection allowed) which of the visual explanations among EBANO, *LIME*, and *Grad-CAM* are representative and understandable for the class of interest of the proposed image. Fig. 16 shows, for each of the 12 images, the percentage of times in which EBANO, *Grad-CAM*, and *LIME* have been selected as the best option in identifying the portions of the image responsible for the prediction of the class of interest. EBANO provided more appreciated explanations w.r.t. the other two methods in 67% of the cases. In particular, EBANO has been selected to be more interpretable than *Grad-CAM* in 75% of the cases and more interpretable than *LIME* on 75% of the cases as well. In absolute terms, over the whole survey (720 answers), EBANO has been selected 439 times as the most interpretable explanation for the class of interest, *Grad-CAM* has been selected 242 times, and *LIME* 156 times.

We noticed that users exploit the quality of the image segmentation as an implicit metric to evaluate the reliability of the model (i.e. the better the image portions look, the most the prediction is considered reliable). Differently from the other methods, the *interpretable features* of EBANO directly correlate the image portions to the knowledge of the model, as explained in Sect. 4.

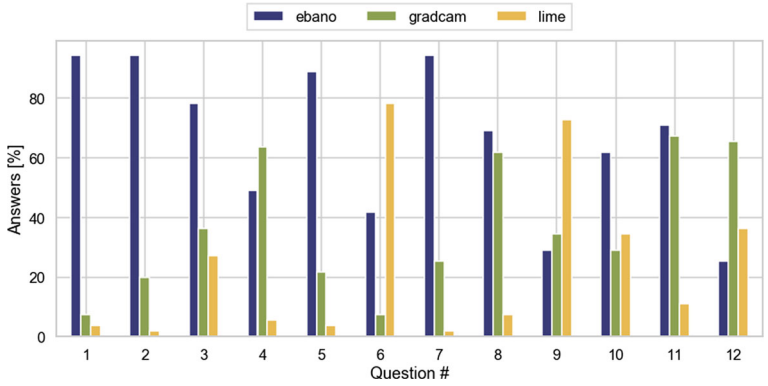


Fig. 16 Survey results. For each of the 12 images, we report the percentage of times that EBANO, *GradCAM*, and *LIME* have been selected as the best explanation result for the class of interest (multiple choice allowed)

Table 7 Execution time comparison

	M1	M2	M3	M4
EBANO	25.1 s	33.2 s	35.4 s	59.0 s
LIME	384.3 s	489.7 s	117.3 s	277.4 s
GradCAM	1.6 s	2.0 s	0.9 s	2.0 s

Mean seconds to produce a single explanation for the different explanations methods and models

10.9 Execution time comparison

Finally, we also briefly compared the time required to compute an explanation by EBANO, *LIME*, and *GradCAM*. The execution time for all the methodologies depends on the complexity of the models. Moreover, the execution time of EBANO depends also on the maximum number of different possible partitioning analyzed K_{max} and the number of convolutional layers used for the extraction of the hypercolumns. We recall that, in our experiments, we set $K_{max} = 10$, and the number of extracted convolutional layers are 5 for M1, 8 for M2, 34 for M3, and 24 for M4. Instead, for *LIME*, the execution time depends on the parameter which specifies the number of perturbed local examples used to train the linear model that we set to 1000.

Table 7 shows the mean execution time (in seconds) to produce a single explanation by the different techniques. The execution times were computed with an Apple MacBook PRO 13 with M1 Chip and 16GB of RAM (then without exploiting GPU). EBANO is slower compared with *GradCAM* (gradient-based), but faster than *LIME* (perturbation-based). Therefore, EBANO also partially bridges some efficiency and runtime gaps of *perturbation-based* techniques with respect to gradient-based ones.

11 Discussion

This work contributes to the XAI literature by proposing a new feature-based technique that primarily differs in that it performs unsupervised analysis of the internal layers of the model when generating explanations and differs in the type of perturbation performed. In this way, effective and interpretable explanations can be generated, as shown by the experimental results. However, this section also discusses limitations that pave the way for future developments of the proposed approach.

11.1 Influence index

EBANO adopted the nPIR as influence index (see Sect. 5.2). The strength of this index is that it takes into account not only the absolute difference in probability, but also the initial value of the class probability. The weakness of the index is its sensitivity to very small values of the original probability p_o and the probability after the perturbation of a feature p_f, for a given class of interest. Such cases are those with the least interest in producing an explanation, i.e., classes whose original predicted probability of the model tends to be 0 are rarely requested to be explained. This limitation could be mitigated by reducing the number of decimal digits of the probability (e.g., keeping only 2 decimal digits).

Furthermore, EBANO could be applied with other user-provided influence indexes at the user's discretion. One example of a possible influence metric is the simple difference between the original probability and the probability after the perturbation of a feature. Such an index would assign the same influence to a perturbation reducing the probability from 0.6 to 0.4, and to another perturbation reducing the probability from 0.2 to 0., whereas the second example is clearly stronger.

Similar considerations also apply to other measures, such as Eq. 10 in Sect. 6, used to select the most informative local explanation. The metric we propose selects the most informative feature division by maximizing the contrast between the most and least influential features. An alternative metric could be applied, for instance, by considering the most positively influential feature, having the highest influence index and the smallest size (i.e., the lowest number of pixels). This can be achieved by using a metric that considers a feature's influence and the number of pixels at the same time.

11.2 Explaining models for multi-label tasks

The experimental models presented in this paper used to validate the proposed methodology were trained on multi-class tasks. In case the models were trained on multi-label tasks (i.e., multiple class labels can be predicted by the model for the same input image), the proposed approach could be applied to each predicted label separately (i.e., explaining one class/label at a time). Similarly to the local explanations provided by EBANO for each class, an explanation could be provided for each label. Each explanation will show the interpretable features (i.e., pixel regions of the input image) and their positive or negative influence on the specified class/label.

It is currently not possible to show the most influential features (i.e., with the highest influence index) for all predicted classes/labels in a single heatmap, because the most influential features of different classes of interest may have overlapping regions of pixels. This limitation can be overcome in future developments by considering multiple classes/labels simultaneously when the features are extracted, the perturbations are applied, and the indices are computed. Hence, the explanation could be shown as a single heatmap containing only the most positively impacted regions overall.

12 Conclusions

This work introduced EBANO, a new explanation framework able to open the decision-making black-box process of Deep Convolutional Neural Networks, providing both prediction-local and class-based model-wise explanations through unsupervised mining. Thanks to the extraction of interpretable features and the definition of a new index that measures features' *influence* and their *influence precision*, EBANO provides detailed visual and numerical explanations. The quantitative and qualitative comparisons w.r.t. the state-of-the-art showed that EBANO is (i) more easily interpretable in visually presenting the features' influence and more detailed in their numerical quantification. (ii) more reliable thanks to the mining of the multiple convolutional layers of the black-box model, as demonstrated by quantitative experiments (iii) more effective and human-readable, as it has been selected by human users as providing the best explanations.

Future works include: i) the exploration of new influence indices that can overcome current limitations; ii) The improvement of the approach for models trained on multi-label tasks by considering multiple classes at the same time in producing the explanations; iii) the application of EBANO to even more complex tasks related to convolutional models (e.g., object detection); iv) the extension of the approach to domains where DCNN-based solutions exist, like audio classification and Natural Language Processing, by generalizing the proposed approach to different scenarios and use cases; v) the extension of the approach to multi-modal models.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement. The research activity is partially supported by the grant "National Centre for HPC, Big Data and Quantum Computing," CN000013 (approved under the M42C Call for Proposals - Investment 1.4 - Notice "National Centers" - D.D. No. 3138, Dec. 16.12.2021, admitted for funding by MUR Decree No. 1031, Dec. 17.06.2022)

Table 8 Mean influence ($\overline{nPIR}$), percentage of pixels ($\overline{\%}_{px}$), and size ($\overline{N}_{px}$) of the most influential features for different Gaussian blur radius values

Blur radius	1	5	10	15	20	50
$\overline{nPIR}$	0.376	0.913	0.941	0.938	0.947	0.964
$\overline{\%}_{px}$	38.6%	38.0%	37.2%	36.3%	36.3%	35.9%
$\overline{N}_{px}$	19,354	19,042	18,687	18,269	18,227	18,002

Appendix A: Gaussian blur perturbation assestment

The appendix sections are organized as follows. "Appendix A" evaluates the impact of the Gaussian blur perturbation radius in the explanations produced. "Appendix B" assesses the performance of several clustering algorithms in the feature extraction process of EBANO. "Appendix C and D" discuss and show some additional local explanations. "Appendix E" shows additional results on the class-based global explanations. "Appendix F" and "Appendix G" integrate the qualitative and the quantitative experimental comparison with state-of-the-art, respectively. "Appendix H" shows the survey structure.

To assess the impact of the Gaussian blur perturbation radius parameter on the explanation produced by EBANO, we evaluated its impact on the influence and size of the most positively influential features extracted. The experiment was performed by randomly sampling 50 images from the test set of ImageNet and producing the explanations of the predictions made by the VGG16 model (M1). For each explanation produced, we measured the mean influence $\overline{nPIR}$, the mean number of pixels $\overline{\%}_{px}$, and the mean percentage of pixels with respect to the total image size $\overline{\%}_{px}$ of the most influential feature for the best k division, using the predicted label as class-of-interest. Table 8 shows the results for different $radius \in [1, 5, 10, 15, 20, 50]$. As can be noticed, if the radius is large enough ($radius \geq 5$), the most positively influential features have very similar characteristics on the measured attributes. These results show that the choice of this parameter does not have an excessive impact on the output produced by EBANO. Therefore, we empirically set the default value of this parameter to 10, but the final user can change it according to its own needs.

Appendix B: Clustering algorithm assestment

We experimented with several possible clustering algorithms to extract the interpretable features from the hypercolumns. Specifically, the evaluated algorithms are: the *Scikit-learn* Pedregosa et al. (2011) implementations of *K-Means*, *DBScan*, *Gaussian Mixture*, *Spectral*, and the *Faiss* Johnson et al. (2021) implementation of *K-Means*. They all require the number of clusters as a parameter, except for *DBScan*. However, even if it is usually a not trivial parameter to set a priori, in our case, it has a specific interpretation (i.e., the number of interpretable features to extract). Therefore, EBANO requires the user specification of the maximum number of features to extract, and it evaluates all the possible divisions in range $[2, max\_features]$, selecting the best (as

Table 9 Mean influence ($\overline{nPIR}$), percentage of pixels ($\overline{\%}_{px}$), and size ($\overline{N}_{px}$) of the most influential features for different clustering algorithm on the hypercolumns

Algorithm	$K-Means_{Scikit}$	*DBScan*	*Gaussian Mixture*	$K-Means_{Faiss}$	*Spectral*
$\overline{nPIR}$	0.94	0.97	0.95	0.94	0.88
$\overline{\%}_{px}$	37.1%	72.4%	35.2%	38.2%	41.4%
$\overline{N}_{px}$	18,594	36,323	17,656	19,191	20,767
$\overline{T}(s)$	13 s	360 s	20 s	11 s	200 s

$\overline{T}$ is the execution time, in seconds, to produce a local explanation per clustering algorithm

discussed in Sects. 4.2 and 6). Concerning *DBScan*, instead, it requires the specification of two parameters: ϵ and $MinPts$. These two parameters do not have a specific interpretation in our domain and, as a result, are difficult to set correctly. As for the other algorithms, different ranges of parameter values could be tested. However, also the possible ranges could change based on the size of the images processed by each specific model.

Moreover, different clustering algorithms make different assumptions on the size and shapes of the clusters. However, we are performing clustering of the hypercolumns (i.e., high-dimensional embedding space) and not on the original input space. Even if the embedding space is composed of easier relationships among the attributes and, consequently, more regular shapes, it is not trivial to prove which algorithm better fits the hypercolumns data features for each possible model. Therefore, we empirically evaluated the possible clustering algorithms, for our purposes, by comparing the performance in the explanation produced. Specifically, we measured the size and the influence of the most influential features and the execution time required to produce the explanation. Table 9 shows the mean influence $\overline{nPIR}$, the mean percentage of pixels with respect to the image size $\overline{\%}_{px}$, and the mean number of pixels $\overline{N}_{px}$ of the most influential features found by EBANO. Moreover, it also reports the mean execution time, in seconds, to produce an explanation $\overline{T}$ using the different clustering algorithms. The results are the average from a dataset of 50 images sampled from the validation set of ImageNet, using a MacBook Pro with M1 chip and 16 GB of RAM. Concerning *DBscan*, for each image, we selected the best explanation produced for $\epsilon \in [0.05, 0.1, 0.5, 1.0]$ and $MinPts \in [1000, 2500, 5000, 10000]$, while for all the other algorithms for all possible $k \in [2-5]$ divisions. The results show that *DBScan*, despite the large number of different ϵ and $MinPts$ values tested, is ineffective because it identifies very influential features but of enormous size (mean percentage of 72.4%). Moreover, due to the difficulty in selecting the best parameter values a priori, it results inefficient in terms of execution time. The *Spectral* algorithm poorly performs in all metrics analyzed. Finally, $K-Means_{Scikit}$, *Gaussian Mixture*, and $K-Means_{Faiss}$ perform similarly in most of the evaluation metrics considered. In conclusion, $K-Means_{Faiss}$ has been chosen as the default hypercolumns clustering algorithm because it turns out to be the fastest, given the same performance on the other metrics, and also supports the use of the GPU to further reduce run times when available.

Appendix C: Additional local explanation results

Figure 17 shows a further example input image (denoted input I3) used to discuss the results obtained with EBANO, highlighting its power. Input I3 shows a pizza with an uncommon heart shape. Models M2 and M3 have been exploited to predict the label for this image. Apparently, in this case, the two models are not influenced by the uncommon shape of the subject: both M2 and M3 produce as the most probable result the class *Pizza* as reported in Tables 10 and 11, respectively. To assess the quality and the reliability of these predictions and, thus, to assert the reliability of models M2 and M3, further investigation is needed. EBANO enables these analyses allowing to wrap the prediction process carried out by the black-box models producing detailed explanations that answer questions like:

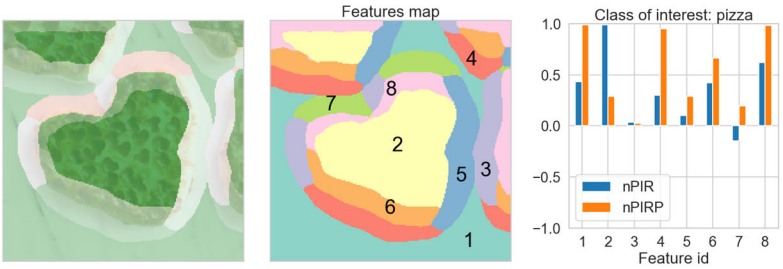
- *Q4.* "Why Fig. 17 is a *Pizza* for model M2?"
- *Q5.* "Why Fig. 17 is a *Pizza* for model M3?"

Answering Q4. Model M2 seems to have taken the correct decision, but with some uncertainty, while predicting the content of input I3, showing a probability of belonging to class *Pizza* of 0.48 and class *Bagel* of 0.16. Thus, to answer the question "Why is it a *Pizza*?" and consequently "Why should it be a *Bagel*?" we can exploit the EBANO engine.

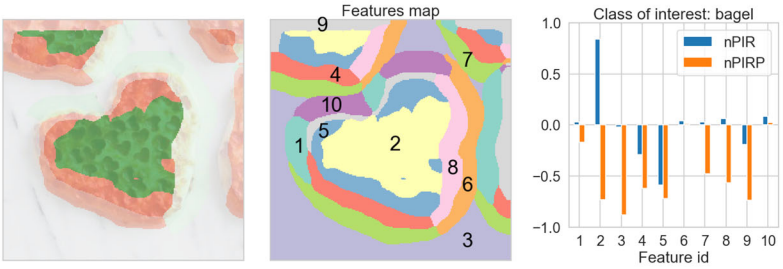
Figure 18a shows the explanation that answers to Q4. The visual explanation (Fig. 18a-left) shows that most of the features in the input are positively impacting the class *Pizza*: the core of the pizza (feature with id 2) is the most positively influential for the decision, but even the table (feature with id 1) is relevant for this class. Inspecting the numerical explanation, we can notice that feature 2 has a very positive nPIRbut a low value of nPIRP, meaning a low precision for the class-of-interest. Instead, feature 1 shows a nPIRvalue close to 0.5, but it is very precise with a nPIRPvalue close to 1. This means that the context of the image (the table under the pizza) is more significant for the class pizza than the pizza itself.

Instead, Fig. 18b shows the reasons why I3 can be confused with the class *Bagel*. The pizza's seasoning (feature with id 2 in Fig. 18b) is considered influential by model M2 for class *Bagel* since its nPIR is close to 0.9, but it is also very negatively precise

Fig. 17 *Pizza* input image (I3)



(a) Explanation of prediction *Pizza* (VGG19 Table 3). The number of interpretable features suggested by the engine equal to 8.



(b) Explanation of prediction *Bagel* (VGG19 Table 3). The number of interpretable features suggested by the engine equal to 10.

(c) Explanation of prediction *Pizza* (InceptionV3 Table 4). Number of interpretable features suggested by the engine equal to 3.

(d) Explanation of prediction *Honeycomb* (InceptionV3 Table 4). Number of interpretable features suggested by the engine equal to 3.

Fig. 18 EBANO Local explanations. The input image is shown in Fig. 17. Visual explanation (left), Interpretable features (center), nPIR and nPIRP (right).(Continue) EBANO Local explanations. The input image is shown in Fig. 17. Visual explanation (left), Interpretable features (center), nPIR and nPIRP (right)

Table 10 VGG19 (M2) predictions for Fig. 17

Class	$P(c)$
Pizza	0.48
Bagel	0.16
Corn	0.05
Pretzel	0.05
Meat loaf	0.04

The ground-truth label is in bold (Pizza)

Table 11 InceptionV3 (M3) predictions for Fig. 17

Class	$P(c)$
Pizza	0.76
Honeycomb	0.24
Dutch oven	0.00
Custard apple	0.00
Starfish	0.00

The ground-truth label is in bold (Pizza)

since nPIRP is strongly negative, meaning that the feature is more influential for other classes than for *Bagel*. In this case, feature 2 is more influential for class *Pizza*, so the prediction can be considered reliable, and the user can trust it.

Answering Q5. Model M3, while predicting the content of input I3, provides a high probability of 0.76 for class *Pizza* instead, and the second relevant predicted class is *Honeycomb* with a probability of 0.24, as reported in Table 11.

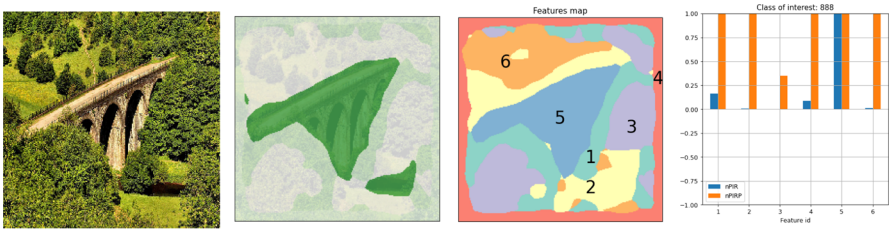
However, by inspecting the explanations produced by EBAnO we can understand that the decision about assigning the class *Pizza* as the most probable class has been taken into account completely wrong assumptions. Figure 18a-left shows the visual explanation: it is clear from the strongly green sections (features 2 and 3 in the feature map of Fig. 18a-center) that the most positively influential elements that contribute to this decision are related to the context of the image. Instead, the pizza itself (feature 1) is slightly negative for the prediction of the class *Pizza*. This behavior is confirmed by the nPIRand nPIRPindexes shown in Fig. 18a-right.

Since the reasons behind the prediction of class *Pizza* are completely erroneous, it is important to check even why model M3 is predicting as second best the class *Honeycomb* to if it is one of the causes of this misleading behavior. The explanation for the class *Honeycomb* is shown in Fig. 18b. From the explanation, it is noticeable that the class *Honeycomb* is predicted by model M3 considering mainly the pizza, while the context is negatively impacting the prediction. This behavior is particularly visible looking at the visual explanation in Fig. 18b-left and supported by the indexes in Fig.18b-right.

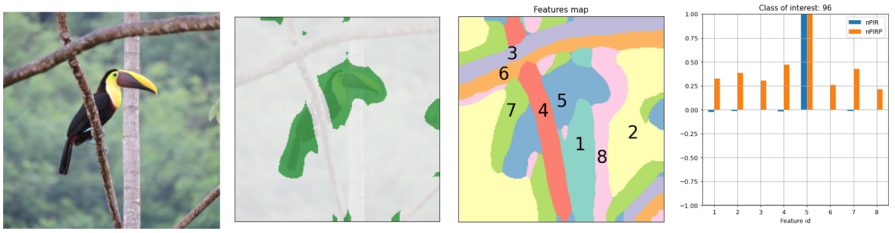
Thus, the decision to assign class *Pizza* is completely biased by the context of the image, even in this case, while the uncommon texture and pattern of the pizza are triggering the decision to assign the *Honeycomb* class.

Appendix D: Further local explanations of correct predictions

In this section, we show some examples of local explanations produced by EBANO with input images correctly classified by the models. By exploiting EBANO, an end-user can inspect the explanations to understand, even if the class label is correct, if the model is using the correct pattern of features or not. Figures 19, 20, 21, and 22 show some examples of local explanations for M1, M2, M3, and M4. For each example, the *most informative local explanation* is selected, and we report from left to right: i) the original image; ii) the visual explanation; iii) the feature map; iv) the numerical explanation. If the influential features (colored in dark green in the visual explanations) are correlated with human judgment, then the predictions can be considered correct and trusted.

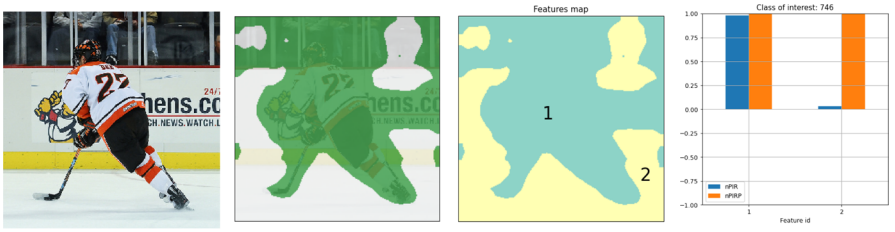


(a) Local Explanation for *Viaduct* class label predicted with probability 0.99 by M1.

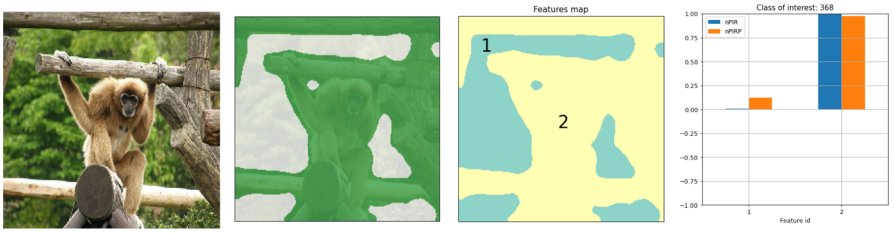


(b) Local Explanation for *Toucan* class label predicted with probability 0.99 by M1.

Fig. 19 Examples of *Local Explanation* with VGG16 model (M1)

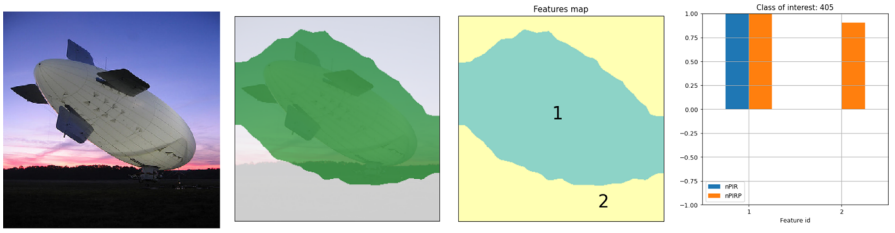


(a) Local Explanation for *Puck* class label predicted with probability 0.99 by M2.

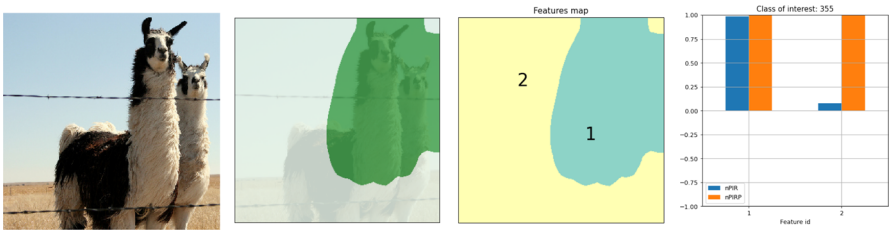


(b) Local Explanation for *Gibbon* class label predicted with probability 0.98 by M2.

Fig. 20 Examples of *Local Explanation* with VGG19 model (M2)

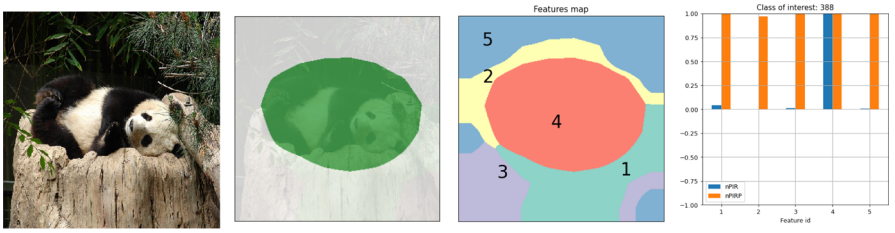


(a) Local Explanation for *Airship* class label predicted with probability 0.99 by M3.

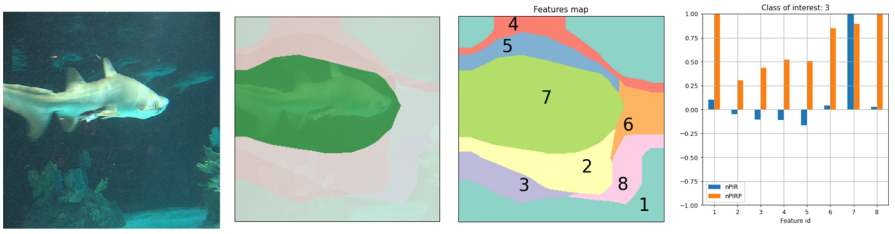


(b) Local Explanation for *Llama* class label predicted with probability 0.99 by M3.

Fig. 21 Examples of *Local Explanation* with InceptionV3 model (M3)



(a) Local Explanation for *Giant Panda* class label predicted with probability 0.92 by M4.



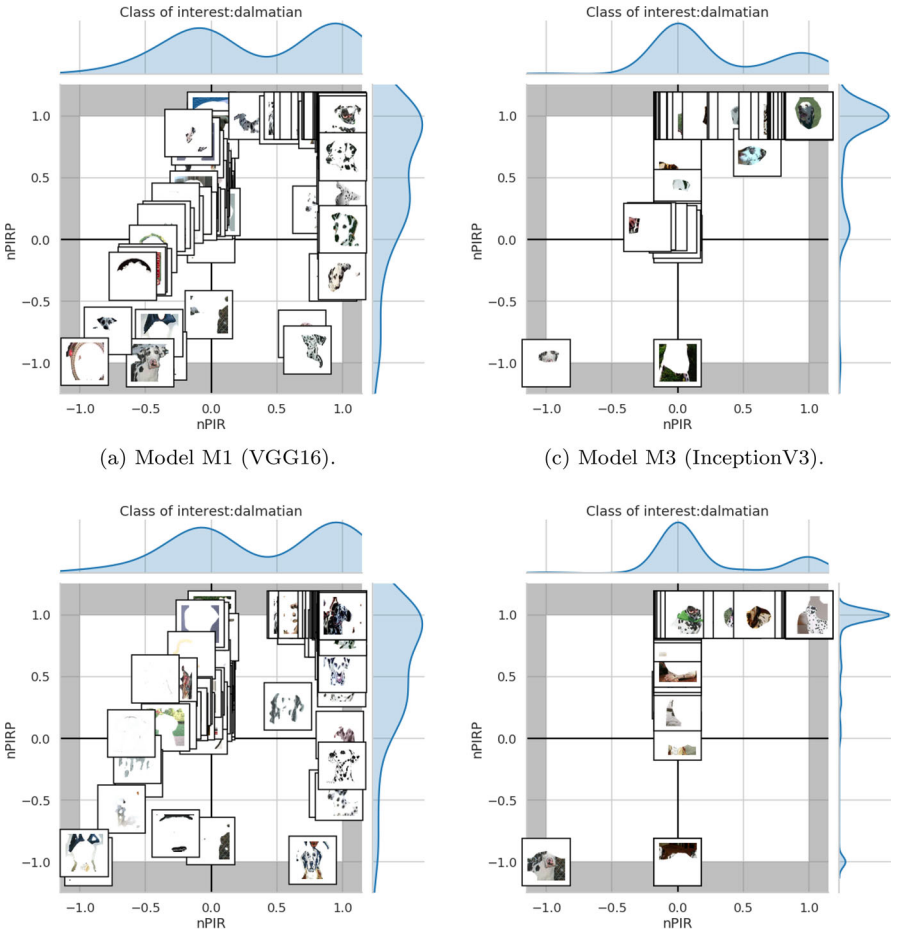
(b) Local Explanation for *Tiger Shark* class label predicted with probability 0.81 by M4.

Fig. 22 Examples of *Local Explanation* with InceptionResNetV2 (M4)

Appendix E: Additional considerations on class-based model explanation

Figure 23 shows a zoom on the class-based model explanations produced by EBAnO for class-of-interest *Dalmatian* with models M1, M2, M3, and M4. A detailed description of explanations for models M1 and M4 is provided in the main paper.

As a further comparison, it is interesting to notice how M1 and M2 are showing a very similar pattern having the *interpretable features* distributed in the whole $nPIR \times nPIRP$ space. Moreover, also M3 and M4 have comparable decision-making patterns but distributing the interpretable features on the edges of the first quadrant. This means that the features used to predict class-of-interest *Dalmatian* have a very high influence precision even if the influence itself is low in some cases. This leads models M3 and M4 to be more reliable w.r.t. M1 and M2.



(a) Model M1 (VGG16). (c) Model M3 (InceptionV3).

(b) Model M2 (VGG19). (d) Model M4. (InceptionResNetV2)

Fig. 23 Class-based model explanation with class-of-interest *Dalmatian*. (Continue) Class-based model explanation with class-of-interest *Dalmatian*

Appendix F: Additional local explanation qualitative comparison

Figure 24 shows a further input image (denoted input I4) taken from Selvaraju et al. (2019) paper to show a fair comparison between their local explanations and the explanations computed by EBAnO. The comparison takes into account the input in Fig. 24 for which model M4 predicts *Bull Mastiff* with a probability of 0.63 (model outcomes are shown in Table 12). Similar to the first comparison, taking into account the class-of-interest *Bull Mastiff*, LIME highlights in green the whole top area of the input that includes the head of the dog as well as much of the background (Fig. 25a). Instead, GRAD-CAM identifies a much more specific region corresponding only to the head of the dog. Both methods, however, still show the same problems highlighted

Fig. 24 Input image (I4) taken from Selvaraju et al. (2019) for comparison

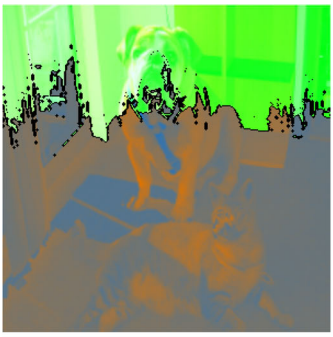
Table 12 InceptionResNetV2 (M4) predictions for Fig. 24

Class	$P(c)$
Bull mastiff	0.63
Tiger cat	0.11
Tiger	0.04
Tabby	0.02
Boxer	0.01

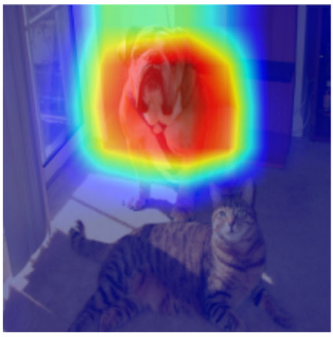
in the first comparison: LIME is not able to precisely identify the influential region, while GRAD-CAM does not evaluate the input areas around the influential ones.

The explanation produced by EBAnO, for class-of-interest *Bull Mastiff*, showed in Fig. 25c instead highlights in green the area containing the head of the dog (visual explanation in Fig. 25c-left), also showing that it is very positively influential and precise, with both nPIR and nPIRP equal to 1 (numerical explanation in Fig. 25c-right). Also, the visual and numerical explanations show that the other features are negatively impacting the prediction process, in particular for what concerns the feature containing the cat (feature 2 in Fig. 25c-center).

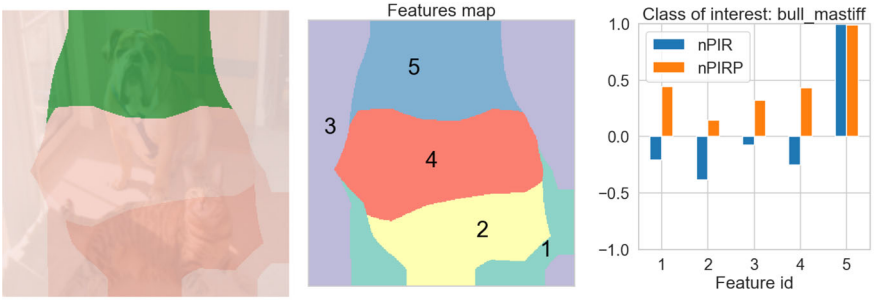
For completeness, also the explanation of class-of-interest *Tiger Cat* is reported in Fig. 25d, showing that the feature containing the cat (feature 5 in Fig. 25d-center) is very positively influential for the prediction process with a nPIR value of almost 1.0 while its nPIRP value is very low giving a possible reason why the probability of class *Tiger Cat* has been predicted to be only 0.11 by model M4 for this image. The dog head is instead marked by EBAnO as correctly negatively impacting the class *Tiger Cat* identifying the model behavior also in this case.



(a) Explanation of prediction *Bull mastiff* with LIME. Green segments are the explanation.



(b) Explanation of prediction *Bull mastiff* with GRAD-CAM.



(c) Explanation of prediction *Bull mastiff* with EBANO. Visual explanation (left), Interpretable features (center), nPIR and nPIRP (right).

(d) Explanation of prediction *Tiger cat* with EBANO. Visual explanation (left), Interpretable features (center), nPIR and nPIRP (right).

Fig. 25 EBANO local explanations. The input image is shown in Fig. 24. Visual explanation (left), Interpretable features (center), nPIR and nPIRP (right)

Appendix G: Additional local explanation quantitative comparison

This section, reports the full set of experiments carried out for the quantitative comparison explained in Sect. 10.7, for M1 (VGG16), M2 (VGG19), M3 (InceptionV3), and M4 (InceptionResNetV2). We recall that *LIME-n* means the combination of the *top-n* most important features of *LIME*. While, *GC-n* means the *top-n* percentile of important pixels (i.e., with highest activation values) found by *GradCAM*.
The *Pointing Game* table reports the results of the full set of feature combinations for *LIME* and *GradCAM*. For each of them, the mean number of pixels $\overline{N}_{px}$, the percentage of the pixels with respect to the total image size (target image size as input of the model) $\overline{\%}_{px}$, the mean number of feature pixels inside the bounding boxes of the target class $\overline{N}_{px}^{bbox}$, and the mean percentage of feature pixels inside the bounding box $\overline{\%}_{px}^{bbox}$ are reported. Smaller features tend to be, as a percentage, more inside the bounding boxes. Probably because centering features composed of fewer pixels inside bounding boxes is a simpler task. However, for the same feature size, EBANO achieves comparable or better performance with respect to *LIME* and *GradCAM*, for all the experimental models.

Instead, the *Pixels Flipping* table reports the full set of feature combinations and also the results for all the occlusion types tested. Specifically, the *Gauss* occlusion, replaces the feature pixels with the Gaussian blur, the *Mean* occlusion with the mean pixel value of the image, the *Black* occlusion with 0, and the *White* occlusion with 255. For each feature combination, we reported the mean number of pixels of the features selected $\overline{N}_{px}$, the mean percentage of feature pixels with respect to the total number of pixels $\overline{\%}_{px}$, the original accuracy of the model for the original set of images Acc_o, the drop of accuracy $\Delta Acc_{o,p}$, and the mean probability decrease for the most probable predicted class $\overline{\Delta P}_{o,p}(\hat{c})$. The best EBANO's performance on all models, with the same feature size, is also confirmed for most other occlusion types (as discussed in Sect. 10.7).

Pointing game		*EBAnO*	*LIME-5*	*LIME-10*	*LIME-15*	*LIME-20*	*LIME-25*	*LIME-50*
M1	$\overline{N}_{px}$	16,284	3379	5326	6621	7617	8437	11,531
	$\overline{\%}_{px}$	32.5%	6.7%	10.6%	13.2%	15.2%	16.8%	23.0%
	$\overline{N}_{px}^{bbox}$	12,106	2675	3834	4511	4923	5321	6721
	$\overline{\%}_{px}^{bbox}$	78.3%	80.1%	74.8%	70.8%	67.3%	66.1%	61.0%
M2	$\overline{N}_{px}$	15,600	3269	5061	6510	7492	8522	11,517
	$\overline{\%}_{px}$	31.1%	6.5%	10.1%	13.0%	14.9%	17.0%	23.0%
	$\overline{N}_{px}^{bbox}$	11,695	2726	3974	4715	5162	5589	7011
	$\overline{\%}_{px}^{bbox}$	78.4%	83.5%	78.5%	75.9%	72.1%	69.7%	63.9%
M3	$\overline{N}_{px}$	25,333	11,240	19,849	27,195	34,329	40,873	60,075
	$\overline{\%}_{px}$	28.3%	12.6%	22.2%	30.4%	38.4%	45.7%	67.2%
	$\overline{N}_{px}^{bbox}$	20,183	9355	14,754	18,304	20,936	23,069	28,884
	$\overline{\%}_{px}^{bbox}$	81.2%	84.4%	75.3%	67.7%	61.4%	56.7%	47.8%
M4	$\overline{N}_{px}$	30,402	11,475	20,057	27735	34,516	41,061	61,125
	$\overline{\%}_{px}$	34.0%	12.8%	22.4 %	31.0%	38.6%	45.9%	68.4%
	$\overline{N}_{px}^{bbox}$	23,195	9,437	14,798	18,592	21,313	23,525	29,16
	$\overline{\%}_{px}^{bbox}$	76.9%	83.1%	74.8%	67.4%	61.8%	57.4%	47.4%

Pointing game		*LIME-75*	*LIME-100*	*GC-90*	*GC-75*	*GC-50*	*GC-25*
M1	$\overline{N}_{px}$	14,138	16,164	3539	8826	17,495	26,214
	$\overline{\%}_{px}$	28.2%	32.2%	7.1%	17.6%	34.9%	52.2%
	$\overline{N}_{px}^{bbox}$	7833	8617	3176	7000	10,912	13,506
	$\overline{\%}_{px}^{bbox}$	57.8%	55.9%	88.7%	79.0%	62.7%	51.7%
M2	$\overline{N}_{px}$	13,821	15,845	3446	8604	17,062	25,547
	$\overline{\%}_{px}$	27.5%	31.6%	6.9%	17.1%	34.0%	50.9%
	$\overline{N}_{px}^{bbox}$	7,850	8624	3109	6889	10,715	13,126
	$\overline{\%}_{px}^{bbox}$	59.8%	57.1%	88.2%	78.8%	62.7%	51.8%
M3	$\overline{N}_{px}$	61,076	61,076	5995	15,066	30,158	45,292
	$\overline{\%}_{px}$	68.3%	68.3%	6.7%	16.9%	33.7%	50.7%
	$\overline{N}_{px}^{bbox}$	29,253	29,253	5,400	12,570	21,308	27,463
	$\overline{\%}_{px}^{bbox}$	47.5%	47.5%	89.0%	82.7%	70.4%	60.7%
M4	$\overline{N}_{px}$	61711	61,711	5,196	13,096	26,261	39,370
	$\overline{\%}_{px}$	69.0%	69.0%	5.8%	14.6%	29.4%	44.0%
	$\overline{N}_{px}^{bbox}$	29,429	29,429	4963	12,083	21,720	28,267
	$\overline{\%}_{px}^{bbox}$	47.2%	47.2%	95.5%	92.0%	81.9%	70.8%

		Pixel flipping	*EBANO*	*LIME-5*	*LIME-10*	*LIME-15*	*LIME-20*	*LIME-25*	*LIME-50*
M1		$\overline{N}_{px}$	15,427	3416	5446	6915	7846	8680	11,920
		$\overline{\%}_{px}$	30.7%	6.8%	10.9%	13.8%	15.6%	17.3%	23.8%
		Acc_o	0.77	0.77	0.77	0.77	0.77	0.77	0.77
	Gauss	Δ_{Acc}	−0.71	−0.08	−0.15	−0.20	−0.24	−0.28	−0.28
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.76	−0.16	−0.23	−0.29	−0.32	−0.34	−0.41
	Mean	Δ_{Acc}	−0.72	−0.17	−0.24	−0.31	−0.36	−0.38	−0.53
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.77	−0.26	−0.37	−0.44	−0.48	−0.51	−0.59
	Black	Δ_{Acc}	−0.76	-0.24	−0.34	−0.40	−0.46	−0.47	−0.54
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.77	−0.33	−0.43	−0.48	−0.52	−0.54	−0.60
	White	Δ_{Acc}	−0.72	−0.24	−0.29	−0.40	−0.42	−0.43	−0.55
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.76	−0.28	−0.38	−0.46	−0.49	−0.51	−0.58
M2		$\overline{N}_{px}$	14,805	3436	5320	6615	7726	8585	11,988
		$\overline{\%}_{px}$	29.5%	6.8%	10.6%	13.2%	15.4%	17.1%	23.9%
		Acc_o	0.77	0.77	0.77	0.77	0.77	0.77	0.77
	Gauss	Δ_{Acc}	−0.74	−0.10	−0.15	−0.16	−0.19	−0.20	−0.26
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.78	−0.15	−0.22	−0.23	−0.26	−0.29	−0.38
	Mean	Δ_{Acc}	−0.72	−0.18	−0.26	−0.28	−0.34	−0.36	−0.48
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.77	−0.27	−0.37	−0.41	−0.45	−0.48	−0.58
	Black	Δ_{Acc}	−0.74	−0.24	−0.29	−0.40	−0.44	−0.44	−0.49
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.78	−0.32	−0.41	−0.46	−0.50	−0.53	−0.58
	White	Δ_{Acc}	−0.73	−0.17	−0.26	−0.39	−0.42	−0.43	−0.52
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.77	−0.27	−0.37	−0.44	−0.48	−0.51	−0.61
M3		$\overline{N}_{px}$	24,388	11,117	19,668	26,947	34,134	40,606	59,230
		$\overline{\%}_{px}$	27.3%	12.4%	22.0%	30.1%	38.2%	45.4%	66.3%
		Acc_o	0.85	0.85	0.85	0.85	0.85	0.85	0.85
	Gauss	Δ_{Acc}	−0.70	−0.29	−0.52	−0.61	−0.68	−0.74	−0.78
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.78	−0.39	−0.58	−0.69	−0.73	−0.76	−0.81
	Mean	Δ_{Acc}	−0.66	−0.36	−0.60	−0.66	−0.78	−0.78	−0.82
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.74	−0.45	−0.65	−0.74	−0.80	−0.81	−0.83
	Black	Δ_{Acc}	−0.70	−0.49	−0.69	−0.76	−0.77	−0.78	−0.82
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.76	−0.58	−0.73	−0.78	−0.80	−0.81	−0.83
	White	Δ_{Acc}	−0.68	−0.41	−0.60	−0.72	−0.76	−0.77	−0.82
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.75	−0.51	−0.68	−0.77	−0.80	−0.81	−0.83
M4		$\overline{N}_{px}$	29,280	11,428	20,021	27,156	34,117	40,808	60,248
		$\overline{\%}_{px}$	32.8%	12.8%	22.4%	30.4%	38.2%	45.6%	67.4%
		Acc_o	0.87	0.87	0.87	0.87	0.87	0.87	0.87
	Gauss	Δ_{Acc}	−0.78	−0.23	−0.53	−0.61	−0.66	−0.73	−0.77
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.79	−0.31	−0.56	−0.66	−0.70	−0.74	−0.80
	Mean	Δ_{Acc}	−0.77	−0.29	−0.56	−0.71	−0.78	−0.81	−0.84
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.78	−0.38	−0.62	−0.73	−0.78	−0.81	−0.84
	Black	Δ_{Acc}	−0.77	−0.43	−0.64	−0.71	−0.76	−0.79	−0.84
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.79	−0.47	−0.67	−0.75	−0.78	−0.80	−0.83
	White	Δ_{Acc}	−0.76	−0.42	−0.62	−0.73	−0.76	−0.80	−0.84
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.78	−0.47	−0.67	−0.75	−0.78	−0.81	−0.84

		Pixel Flipping	*LIME-75*	*LIME-100*	*GC-5*	*GC-10*	GC-25	GC-50	*GC-75*
M1		$\overline{N}_{px}$	14,222	16,481	1690	3383	8444	16,747	25,081
		$\overline{\%}_{px}$	28.3%	32.8%	3.4%	6.7%	16.8	33.4%	50.0%
		Acc_o	0.77	0.77	0.77	0.77	0.77	0.77	0.77
	Gauss	Δ_{Acc}	−0.39	−0.44	−0.12	−0.25	−0.50	−0.60	−0.61
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.47	−0.52	−0.21	−0.34	−0.56	−0.66	−0.71
	Mean	Δ_{Acc}	−0.58	−0.62	−0.16	−0.28	−0.52	−0.64	−0.68
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.64	−0.67	−0.27	−0.40	−0.60	−0.70	−0.73
	Black	Δ_{Acc}	−0.59	−0.64	−0.22	−0.37	−0.56	−0.68	−0.72
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.64	−0.67	−0.32	−0.47	−0.66	−0.73	−0.75
	White	Δ_{Acc}	−0.61	−0.62	−0.18	−0.32	−0.54	−0.61	−0.68
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.63	−0.66	−0.27	−0.41	−0.61	−0.68	−0.73
M2		$\overline{N}_{px}$	14,410	16,617	1675	3355	8374	16,633	24,900
		$\overline{\%}_{px}$	28.7%	33.1%	3.3%	6.7%	16.7%	33.1%	49.6%
		Acc_o	0.77	0.77	0.77	0.77	0.77	0.77	0.77
	Gauss	Δ_{Acc}	−0.32	−0.37	−0.14	−0.23	−0.47	−0.61	−0.61
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.45	−0.49	−0.23	−0.36	−0.57	−0.67	−0.69
	Mean	Δ_{Acc}	−0.56	−0.62	−0.20	−0.29	−0.49	−0.65	−0.66
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.64	−0.66	−0.30	−0.41	−0.61	−0.69	−0.72
	Black	Δ_{Acc}	−0.52	−0.61	−0.23	−0.35	−0.61	−0.69	−0.70
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.62	−0.67	−0.35	−0.47	−0.66	−0.72	−0.74
	White	Δ_{Acc}	−0.60	−0.63	−0.16	−0.30	−0.50	−0.64	−0.67
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.66	−0.69	−0.30	−0.42	−0.62	−0.69	−0.72
M3		$\overline{N}_{px}$	60,221	60,291	2980	6017	15,118	30,278	45,462
		$\overline{\%}_{px}$	67.4%	67.4%	3.3%	6.7%	16.9%	33.9%	50.9%
		Acc_o	0.85	0.85	0.85	0.85	0.85	0.85	0.85
	Gauss	Δ_{Acc}	−0.78	−0.78	−0.08	−0.14	−0.32	−0.59	−0.74
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.81	−0.81	−0.14	−0.23	−0.45	−0.68	−0.77
	Mean	Δ_{Acc}	−0.82	−0.82	−0.07	−0.16	−0.34	−0.58	−0.77
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.83	−0.83	−0.17	−0.24	−0.47	−0.67	−0.78
	Black	Δ_{Acc}	−0.82	−0.82	−0.16	−0.24	−0.44	−0.68	−0.78
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.83	−0.83	−0.24	−0.35	−0.58	−0.73	−0.80
	White	Δ_{Acc}	−0.82	−0.82	−0.06	−0.16	−0.42	−0.62	−0.78
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.83	−0.83	−0.18	−0.29	−0.53	−0.71	−0.80
M4		$\overline{N}_{px}$	60,905	60,905	2587	5237	13,187	26,436	39,603
		$\overline{\%}_{px}$	68.1%	68.1%	2.9%	5.9%	14.8%	29.6%	44.3%
		Acc_o	0.87	0.87	0.87	0.87	0.87	0.87	0.87
	Gauss	Δ_{Acc}	−0.77	−0.77	−0.04	−0.14	−0.34	−0.62	−0.74
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.80	−0.80	−0.10	−0.19	−0.41	−0.66	−0.76
	Mean	Δ_{Acc}	−0.84	−0.84	−0.02	−0.16	−0.33	−0.61	−0.80
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.84	−0.84	−0.12	−0.22	−0.43	−0.68	−0.82
	Black	Δ_{Acc}	−0.84	−0.84	−0.08	−0.22	−0.45	−0.71	−0.80
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.83	−0.83	−0.16	−0.28	−0.50	−0.75	−0.82
	White	Δ_{Acc}	−0.84	−0.84	−0.05	−0.17	−0.42	−0.67	−0.83
		$\overline{\Delta P}_{o,p}(\hat{c})$	−0.84	−0.84	−0.13	−0.23	−0.47	−0.72	−0.83

Appendix H: Survey structure

Figure 26 shows an example question of the survey proposed to validate the human interpretability of the prediction-local explanations produced by EBAnO. The question is divided into three sections:

1. The user can inspect the input image with details about the model's prediction.

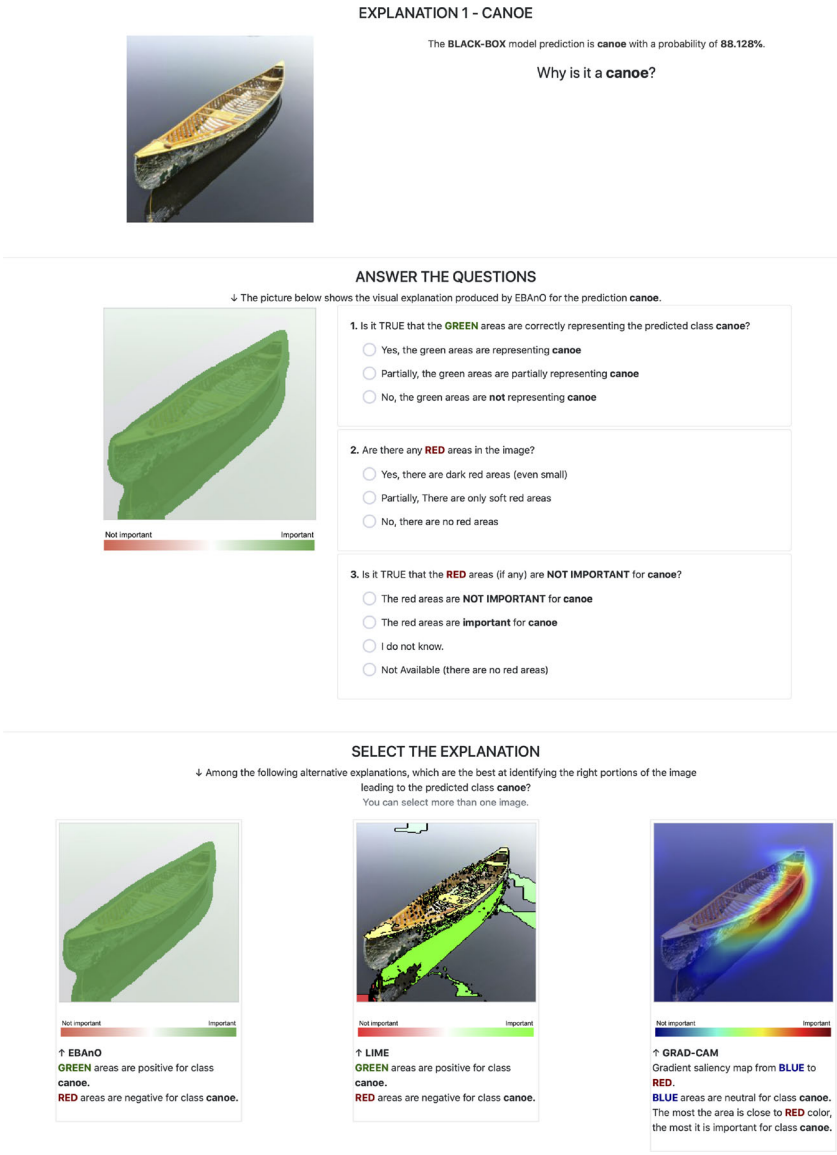


Fig. 26 Survey example question

2. Sub-questions regarding the relevance of green areas and the presence and the irrelevance of red areas are proposed to the user.
3. The visual explanations computed by EBAnO *LIME* and *GRAD-CAM* are proposed to the user who is asked to select the ones that better represent the prediction of the target class-of-interest.

References

Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (xai). IEEE Access 6:52138–52160

Akhtar N, Mian A (2018) Threat of adversarial attacks on deep learning in computer vision: A survey. *CoRR*, arXiv:1801.00553

Alvarez-Melis D, Jaakkola T (2017) A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp 412–421, Copenhagen, Denmark. Association for Computational Linguistics

Ancona M, Ceolini E, öztireli C, Gross MH (2019) Gradient-based attribution methods. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L., Müller, KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer International Publishing, Cham, 169–171. https://doi.org/10.1007/978-3-030-28954-6_9

Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE 10(7):1–46

Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828

Binder A, Montavon G, Lapuschkin S, Müller K-R, Samek W (2016) Layer-wise relevance propagation for neural networks with local renormalization layers. In: Villa AE, Masulli P, Pons Rivero AJ (eds) Artificial neural networks and machine learning - ICANN 2016. Springer International Publishing, Cham, pp 63–71

Bolukbasi T, Chang KW, Zou J, Saligrama V, Kalai AT (2016) Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: Advances in Neural Information Processing Systems, vol 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf

Chollet F et al. (2015) Keras. https://keras.io

Clark A (2015) Pillow (pil fork) documentation

Confalonieri R, Coba L, Wagner B, Besold TR (2021) A historical perspective of explainable artificial intelligence. WIREs Data Min Knowl Discov 11(1):e1391

Datta A, Sen S, Zick Y (2016) Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pp 598–617

Díez J, Pérez-Núñez P, Luaces (2020) Towards explainable personalized recommendations by learning from users' photos. Inf Sci 520:416–430

Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. *2017 IEEE international conference on computer vision (ICCV)*

Ghorbani A, Wexler J, Zou JY, Kim B (2019) Towards automatic concept-based explanations. In *Advances in neural information processing systems*, pp 9273–9282

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. ACM Comput Surv 51(5):93:1-93:42

Gwosdek P, Grewenig S, Bruhn A, Weickert J (2012) Theoretical foundations of gaussian convolution by extended box filtering. In: Bruckstein AM, ter Haar Romeny BM, Bronstein AM, Bronstein MM (eds) *Scale space and variational methods in computer vision*, pp 447–458. Springer Berlin Heidelberg, Berlin, Heidelberg

Hariharan B, Arbeláez P, Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 447–456

Johnson J, Douze M, Jegou H (2021) Billion-scale similarity search with gpus. IEEE Trans Big Data 7(03):535–547

Kapishnikov A, Bolukbasi T, Viégas F, Terry M (2019) Xrai: better attributions through regions. arXiv:1906.02825

Kim B, Wattenberg M, Gilmer J, Cai CJ, Wexler J, Viégas FB, Sayres R (2018) Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: Dy J, Krause A (eds) Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 80. PMLR, pp 2688–2677. https://proceedings.mlr.press/v80/kim18d.html

Kliegr T, Bahník Štěpán, Fürnkranz J (2021) A review of possible effects of cognitive biases on interpretation of rule-based machine learning models. Artif Intell 295:103458

Lapuschkin S, Wäldchen S, Binder A, Montavon G, Samek W, Müller K (2019) Unmasking clever hans predictors and assessing what machines really learn. *CoRR*, arXiv:1902.10178

Lepri B, Staiano J, Sangokoya D, Letouzé E, Oliver N (2017) The tyranny of data? The bright and dark sides of data-driven decision-making for social good. Springer International Publishing, Cham, pp 3–24

Li Fei-Fei Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: *2004 conference on computer vision and pattern recognition workshop*, pp 178–178

Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision - ECCV 2014. Springer International Publishing, Cham, pp 740–755

Lin Y, Ren P, Chen Z, Ren Z, Ma J, de Rijke M (2020) Explainable outfit recommendation with joint outfit matching and comment generation. IEEE Trans Knowl Data Eng 32(8):1502–1516

Lloyd S (1982) Least squares quantization in pcm. IEEE Trans Inf Theory 28(2):129–137

Lonjarret C, Robardet C, Plantevit M, Auburtin R, Atzmueller M (2020) Why should i trust this item? explaining the recommendations of any model. In: *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*, pp 526–535

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30. Curran Associates Inc, pp 4765–4774

Mahendran A, Vedaldi A (2016) Visualizing deep convolutional neural networks using natural pre-images. Int J Comput Vis 120(3):233–255

Minaee S, Boykov Y, Porikli F, Plaza A, Kehtarnavaz N, Terzopoulos D (2020) Image segmentation using deep learning: a survey. *CoRR*, arXiv:2001.05566

Montavon G, Bach S, Binder A, Samek W, Müller K (2015) Explaining nonlinear classification decisions with deep taylor decomposition. *CoRR*, arXiv:1512.02479

Petsiuk V, Das A, Saenko K (2018) RISE: randomized input sampling for explanation of black-box models. In: *British machine vision conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3–6, 2018*, p 151

Proença HM, van Leeuwen M (2020) Interpretable multiclass classification by mdl-based rule lists. Inf Sci 512:1372–1393

Rabold J, Schwalbe G, Schmid U (2020) Expressive explanations of dnns by combining concept analysis with ilp. In: Schmid U, Klügl F, Wolter D (eds) KI 2020: advances in artificial intelligence. Springer International Publishing, Cham, pp 148–162

Rajapaksha D, Bergmeir C, Buntine W (2020) Lormika: local rule-based model interpretability with k-optimal associations. Inf Sci 540:221–241

Ribeiro MT, Singh S, Guestrin C (2016) "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*, KDD'16, pp 1135–1144, New York, NY, USA. Association for Computing Machinery

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis (IJCV) 115(3):211–252

Samek W, Müller K (2019) Towards explainable artificial intelligence. *CoRR*, arXiv:1909.12072

Seifert C, Aamir A, Balagopalan A, Jain D, Sharma A, Grottel S, Gumhold S (2017) *Visualizations of deep neural networks in computer vision: a survey*, pp 123–144. Springer International Publishing, Cham

Selvaraju RR, Cogswell M, Das A et al (2020) Grad-CAM: visual explanations from deep networks via gradient-based localization. Int J Comput Vis 128:336–359. https://doi.org/10.1007/s11263-019-01228-7

Shapley LS (1953) A value for n-person games. Contrib Theory Games 2(28):307–317

Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. *CoRR*, arXiv:1704.02685

Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv:1312.6034

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556

Smilkov D, Thorat N, Kim B, Viégas FB, Wattenberg M (2017) Smoothgrad: removing noise by adding noise. *CoRR*, arXiv:1706.03825

Štrumbelj E, Kononenko I (2014) Explaining prediction models and individual predictions with feature contributions. Knowl Inf Syst 41(3):647–665

Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. *CoRR*, arXiv:1703.01365

Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the thirty-first AAAI conference on artificial intelligence*, AAAI'17, pp 4278–4284. AAAI Press

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2818–2826

Ventura F, Cerquitelli T, Giacalone F (2018) Black-box model explained through an assessment of its interpretable features. In *New trends in databases and information systems - ADBIS 2018 short papers and workshops, AI*QA, BIGPMED, CSACDB, M2U, BigDataMAPS, ISTREND, DC, Budapest, Hungary, September, 2–5, 2018, Proceedings*, pages 138–149

Ventura F, Greco S, Apiletti D, Cerquitelli T (2022) Trusting deep learning natural-language models via local and global explanations. Knowl Inf Syst 64:1863–1907. https://doi.org/10.1007/s10115-022-01690-9

Yeh CK, Kim B, Arik S, Li CL, Ravikumar P, Pfister T (2020) On completeness-aware concept-based explanations in deep neural networks. In; Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada. Curran Associates Inc., Red Hook, NY, 1726

Yeo J, Park H, Lee S, Lee EW, Hwang S (2020) Xina: explainable instance alignment using dominance relationship. IEEE Trans Knowl Data Eng 32(2):388–401

Zhang Q, Wu YN, Zhu S (2018) Interpretable convolutional neural networks. In: *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp 8827–8836

Zheng X, Wang M, Chen C, Wang Y, Cheng Z (2019) Explore: explainable item-tag co-recommendation. Inf Sci 474:170–186

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, pp 2921–2929

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830