



# Towards a Joint Approach to Produce Decisions and Explanations Using CNNs

Isabel Rio-Torto<sup>1</sup>(✉), Kelwin Fernandes<sup>3</sup>, and Luís F. Teixeira<sup>1,2</sup>

<sup>1</sup> Faculdade de Engenharia da Universidade do Porto, Porto, Portugal  
icrto@gmail.com, luisft@fe.up.pt

<sup>2</sup> INESC TEC, Porto, Portugal

<sup>3</sup> NILG.AI, Porto, Portugal  
kelwin@nilg.ai

**Abstract.** Convolutional Neural Networks, as well as other deep learning methods, have shown remarkable performance on tasks like classification and detection. However, these models largely remain black-boxes. With the widespread use of such networks in real-world scenarios and with the growing demand of the right to explanation, especially in highly-regulated areas like medicine and criminal justice, generating accurate predictions is no longer enough. Machine learning models have to be explainable, i.e., understandable to humans, which entails being able to present the reasons behind their decisions. While most of the literature focuses on post-model methods, we propose an in-model CNN architecture, composed by an explainer and a classifier. The model is trained end-to-end, with the classifier taking as input not only images from the dataset but also the explainer's resulting explanation, thus allowing for the classifier to focus on the relevant areas of such explanation. We also developed a synthetic dataset generation framework, that allows for automatic annotation and creation of easy-to-understand images that do not require the knowledge of an expert to be explained. Promising results were obtained, especially when using L1 regularisation, validating the potential of the proposed architecture and further encouraging research to improve the proposed architecture's explainability and performance.

**Keywords:** Explainable AI · Explainability · Interpretability · Deep learning · Convolutional Neural Networks

## 1 Introduction

Deep learning changed the machine learning paradigm in recent years, significantly improving performance on tasks like classification, sometimes even outperforming humans. Due to the achieved outstanding predictive capability, deep learning methods have since been employed in tackling other problems, such

---

This work was partially funded by [NILG.AI](#).

© Springer Nature Switzerland AG 2019  
A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11867, pp. 3–15, 2019.  
[https://doi.org/10.1007/978-3-030-31332-6\\_1](https://doi.org/10.1007/978-3-030-31332-6_1)

as detection and segmentation, surpassing the performance of classical machine learning models also on these tasks.

Despite this overwhelming dominance, deep learning models, and in particular convolutional neural networks (CNNs), are still considered black-box models, i.e., models whose reasons for the outputted decisions cannot be understood at a human-level. However, with the growing ubiquitousness of deep learning systems, especially in highly regulated areas such as medicine, criminal justice or financial markets [6], an increasing need for these models to output explanations in addition to decisions is arising. Moreover, the new General Data Protection Regulation (GDPR) includes policies on the right to explanation [5], thus increasing this need for explainable deep learning systems that can operate within this legal framework.

The research community has rapidly taken an interest in this topic, proposing several methods that try to meet this explainability requirement. Nevertheless, the field is still lacking a unified formal definition or possible evaluation metrics. The terms explainability and interpretability are often used interchangeably in the literature. In this work, we adopt the definition proposed by Gilpin et al. [4]. The authors loosely define interpretability as the process of understanding the model’s internals and describing them in a way that is understandable to humans, while explainability goes beyond that. Briefly, an explainable model is one that can summarise the reasons for its behaviour or the causes of its decisions. In fact, for a model to be explainable, it needs to be interpretable, but also complete, i.e., to describe the system’s internals accurately. As such, explainable models are interpretable by default, while the reverse may not be true. Therefore, a good explanation should be able to balance the interpretability-completeness trade-off, because the more accurate an explanation, the less interpretable it is to humans; for example, an entirely complete explanation of a neural network would consist of all the operations, parameters and hyperparameters of such network, rendering it uninterpretable. Conversely, the most interpretable description is often incomplete.

The majority of the literature focuses only on interpretability, and more specifically on post-model or post-hoc interpretability methods, i.e. methods that are applied after the model is trained. Examples range from proxy methods that approximate the original network model [9] to methods that output visual cues representing what the network is focusing on to make its decisions, such as Sensitivity Analysis and Saliency Maps [8], SmoothGrad [10], DeConvNet [17] or Layer-Wise Relevance Propagation [2].

While a few works focus on in-model approaches, in which interpretability is taken into account while building the model, these are for the most part application oriented. Some work has been developed trying to make predictions based only on patches of the input images, which limits the interpretability of the classifier. Although such in-model methods exist for models such as CNNs [11, 12], these are still not considered intrinsically interpretable, making this category still dominated by classic methods like decision trees.

In this work, we propose a preliminary end-to-end in-model approach, based on an explainer+classifier architecture. This architecture outputs not only a class label, but also a visual explanation of such decision. The classifier takes as input an image, as well as the output of the explainer, i.e. it is trained using the explanation. Therefore, the classifier focuses on the regions the explainer deemed relevant and does not take into account regions where the explanation for that class is not present. This approach aligns with the intuition that, when explaining a decision, for example, whether or not image X is an image of a car, humans tend to distinguish what is an object and what is not, and then proceed to look for patterns in the region where the object is in order to classify it correctly. Conversely, sometimes humans cannot tell if an object belongs to some class, but can tell which regions of the image do not contain said class.

We also propose a synthetic dataset generation framework, allowing for automatic image generation and annotation. The generated images consist of simple polygons, therefore easily explainable by humans, which allows for a qualitative and quantitative evaluation of the produced explanations without the need of expert knowledge, necessary in most real-world datasets.

## 2 Methodology

### 2.1 Proposed Architecture

We propose a model consisting of an explainer and a classifier, as depicted in Fig. 1. Figure 2 depicts a detailed diagram of the proposed architecture, which is a concretisation of the aforementioned model. The explainer (top row) outputs an image, which we call the explanation, with the same spatial dimensions as the input image. It is composed of a downsampling and an upsampling path. The downsampling path is a simple convolution-convolution-pooling scheme. The upsampling path follows a convolution-convolution-deconvolution scheme, where the first convolution operation is applied to the sum of the previous layer’s output with the corresponding convolutional layer in the downsampling path. These connections allow for the successive layer to learn a more precise output. Also, batch normalisation is applied to the last layer of the explainer.

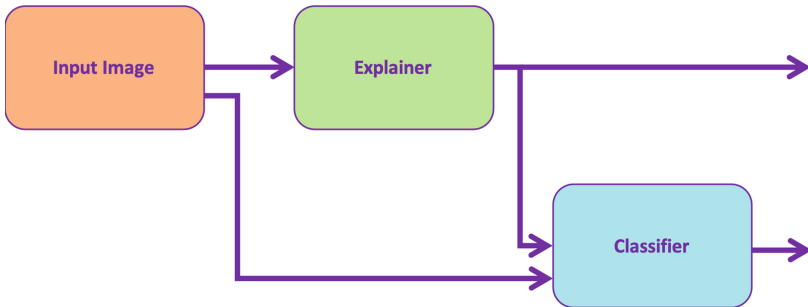
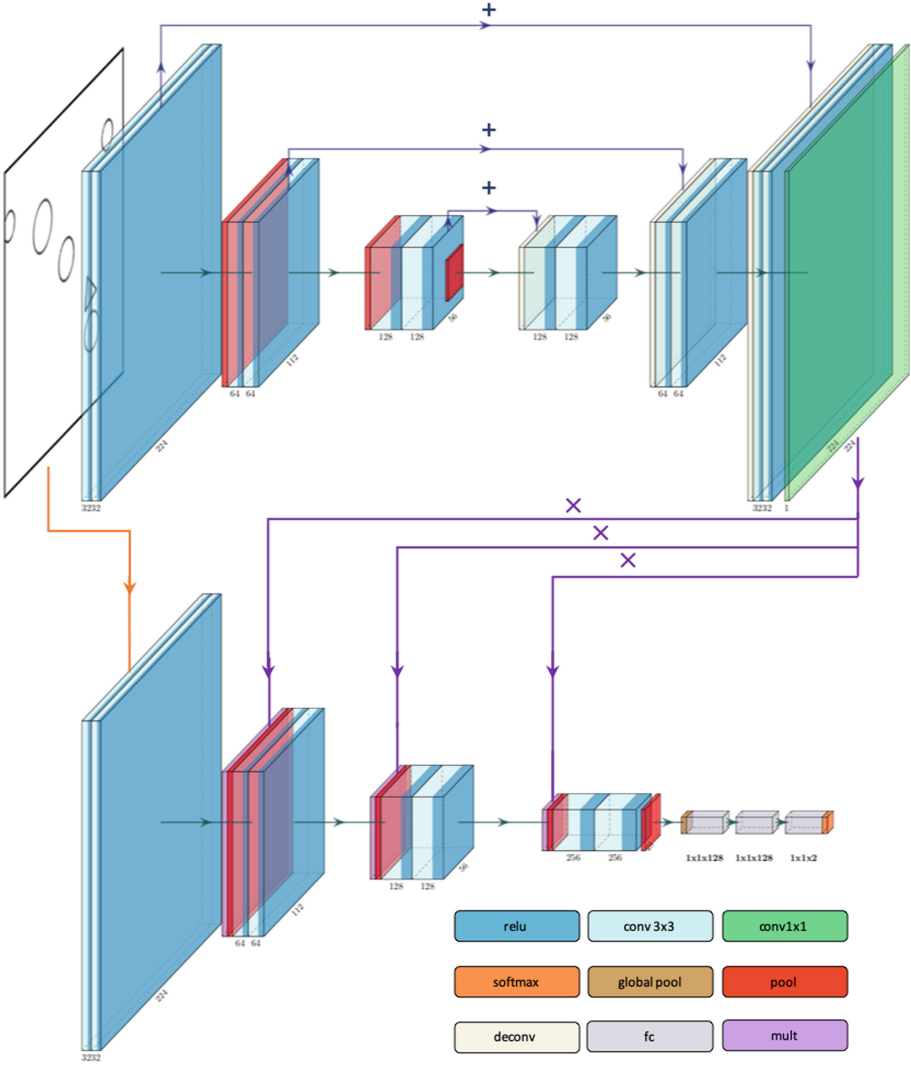


Fig. 1. Block diagram of the proposed explainer+classifier model.



**Fig. 2.** Diagram of the proposed architecture. It is composed by an explainer (top row) and a classifier (bottom row). The sums correspond to the simple addition of the outputs of the respective layers. The multiplications involve the concatenation and resizing of the explainer’s output before computing the element-wise multiplication with the involved classifier layer. This architecture outputs a class label, as well as an explanation for that decision.

The classifier (bottom row) is inspired by the VGG architecture [13], having 4 consecutive convolution-convolution-pool stages, ending with 2 fully connected layers, followed by a softmax layer. However, an important modification to the original VGG is made: each pooling layer takes as input the multiplication of the output of its preceding layer with the output of the explainer. This way, the classifier is trained using the outputted explanation, allowing for it to focus on the relevant parts of the input image and to discard regions where the explanation for the class being predicted is not present.

In both classifier and explainer,  $3 \times 3$  kernels are used in the convolutional layers. All pooling operations resort to max pooling, downsampling by a factor of 2. The number of filters starts at 32 for the first stage and afterwards increases as a power of 2 according to its stage level.

Training involves three steps. First, only the classifier is trained, taking a “white image” as explanation, meaning that the initial explanation is the whole image. Then the explainer is trained with the classifier frozen, outputting an explanation that highlights relevant areas to the classification task. Finally, the whole architecture is fine-tuned end to end.

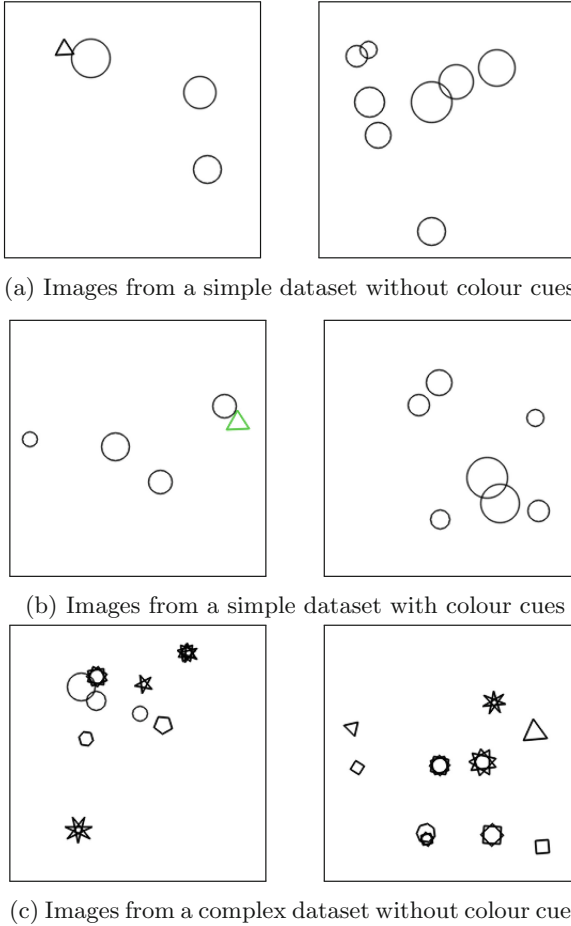
While the explainer is trained unsupervised, the classifier is trained using the categorical cross-entropy loss. The Keras Adadelta default optimizer was used, which employs an adaptive learning rate based on a moving window of gradient updates [16].

## 2.2 Synthetic Dataset

For the experimental assessment of the proposed architecture, a synthetic dataset generation framework was developed. The use of a synthetic dataset entails numerous advantages, such as:

- the ability to generate as many images as one needs
- the possibility of defining the number of instances for each class
- automatic annotation for different problems ranging from classification to detection
- definition of custom-made characteristics like overlap, occlusion, object type, object colour, image dimensions, etc.

This framework generates images consisting of several polygons, such as triangles, circles and stars. Examples of such images can be observed in Fig. 3. Moreover, for each image, an XML annotation file in PASCAL VOC format is created, containing information on how many target polygons exist in the image and their respective bounding boxes. The developed code is available at <https://github.com/icrto/xML>.



**Fig. 3.** Example images of 3 generated datasets. For each row, the left column illustrates an example of the positive class, while the right column illustrates the negative class. On the first two rows the target polygon is a triangle, while on the third row it is a 5 pointed star.

### 2.3 Experiments

The synthetic dataset used in the experiments consists of 1000  $224 \times 224$  RGB images, each containing a triangle (target polygon) and a variable number of circles. For each image it is only taken into account the presence or absence of the target polygon, thus making these experiments binary classification tasks. The positive class has 618 instances, while the remaining 382 instances belong to the negative class. The data was split into a 75–25 training-validation partition. Each of the three training phases involved 50 epochs with 50 steps each, with a batch size of 8. The classifier was evaluated in terms of its accuracy and the explainer was qualitatively evaluated, by means of human visual inspection.

All experiments were conducted on the Google Collaboratory environment and involved applying to the explainer different L1 regularisation factors, ranging from  $10^{-8}$  to  $10^{-4}$ . Without regularisation, one can obtain a degraded solution, in which everything is considered an explanation. Therefore, L1 regularisation is employed, so that only a small part of the whole image constitutes an explanation.

A qualitative and quantitative comparison of the proposed architecture with various methods available in the iNNvestigate toolbox [1] is also made. This toolbox aims to facilitate the comparison of reference implementations of post-model interpretability methods, by providing a common interface and out-of-the-box implementation of various analysis methods. The toolbox is, then, used to compare the proposed architecture with methods like SmoothGrad [14], DeconvNet [17], Guided Backprop [15], Deep Taylor Decomposition [7] or Layer-Wise Relevance Propagation (LRP) [2]. In the proposed architecture, the explainer is the component that produces a visual representation of the reasoning behind the classifier’s decisions, just like the analysers available in the iNNvestigate toolbox. As such, these analysis methods are applied only to the classifier of the proposed architecture, in order to compare only the explanation generators, i.e. the proposed architecture’s explainer and the different analysis methods. Since these methods are applied after the model is trained, we started by training the classifier on the simple dataset without colour cues. Then, the various analysis methods are applied to the trained classifier and their generated visual explanations are compared to the ones outputted by the explainer trained in the previous experiments with  $10^{-6}$  L1 regularisation factor. Furthermore, the classifier’s accuracy with and without explainer are also compared. The obtained results are described in Sect. 3.

## 2.4 Experiments on Real Datasets

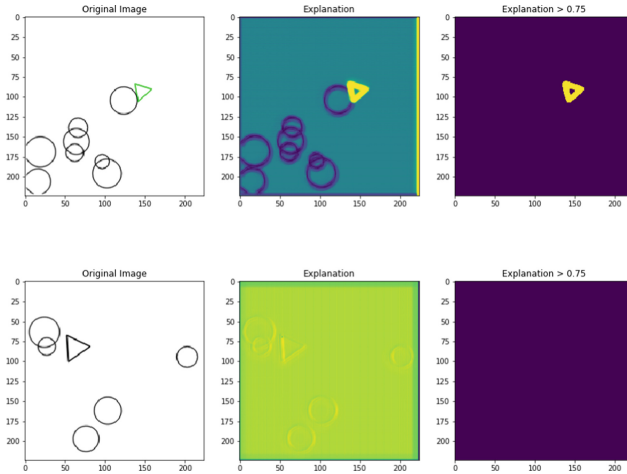
Experiments were also conducted on a real dataset, available at <https://github.com/rgeirhos/texture-vs-shape>. This dataset was created in the context of the work developed by Geirhos et al. [3], where the authors validate that Imagenet-trained CNNs are biased towards texture. In order to validate this hypothesis, the authors propose a cue conflict experiment in which style transfer is employed, introducing texture in the Imagenet images. This dataset contains 16 classes, with 80 images each. The proposed architecture was trained on this dataset without any regularisation. Results for this experiment are shown in Sect. 3.

## 3 Results and Discussion

For all of the following images it is worth noting that the colour code ranges from purple to yellow, where yellow represents higher pixel values. The left column corresponds to the original image, the middle column to the outputted explanation and the right column to the explanation after an absolute threshold of 0.75 is applied.

Figure 4 constitutes examples of the obtained results for simple datasets with and without a target polygon of different colour. Both images are the result of training without any kind of regularisation. For such simple datasets, it is expected that the explanation focuses on the target polygon, rendering the rest of the image as irrelevant for the predicted class.

While on the dataset with colour cues the resulting explanation consists only of the target polygon, as expected, in the slightly more complex dataset without colour cues the whole image is considered an explanation, which corroborates the need for regularising the explainer output. As stated in Sect. 2.3, we use an L1 regularisation factor, because it allows for the selection of the relevant parts of the explanation, ensuring that only a small part of the image is in fact the explanation of the classifier’s decision. Thus, this regularisation ensures that the explanation is not only interpretable, but also complete, as desired.

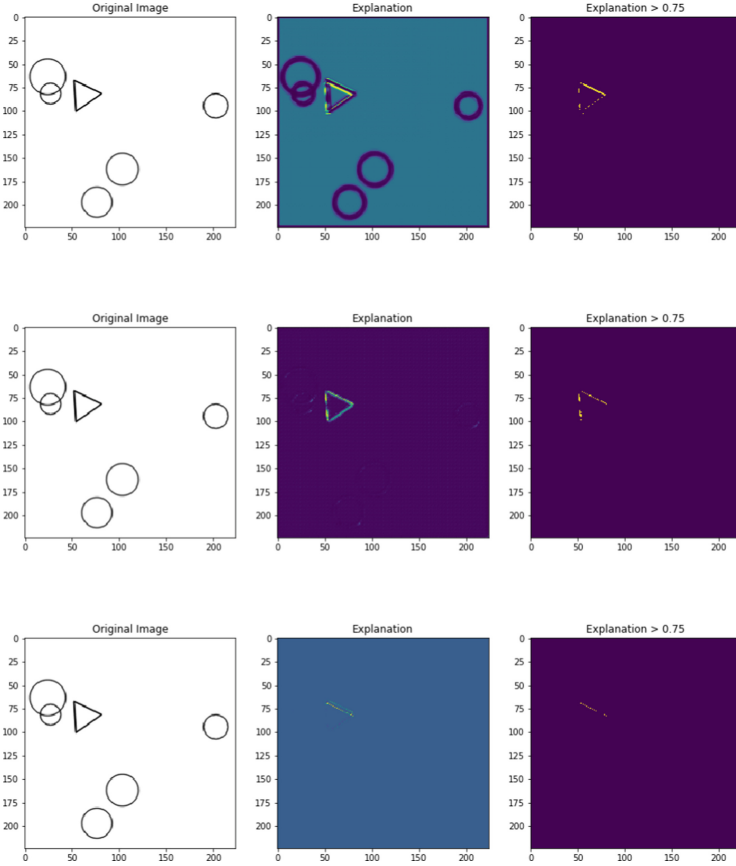


**Fig. 4.** Positive instance and respective explanation. These results were obtained without any kind of regularisation of the explainer’s output while training on a simple dataset without (top) and with (bottom) colour cues

Figure 5 is the result of the experiments with different regularisation factors, namely  $10^{-8}$ ,  $10^{-6}$  and  $10^{-4}$ , on the dataset without colour cues. With a factor of  $10^{-8}$ , not only the target polygon is considered relevant, as well as the circles, which may imply that such a small regularisation is still not enough to limit the relevant parts of the explanation. In fact, increasing L1 to  $10^{-6}$ , produces much better results, with the target polygon clearly highlighted. Finally, increasing L1 a bit further, to  $10^{-4}$ , proved to be too much regularisation, causing the explanation to “disappear”.

Moreover, the proposed architecture was compared to several other methods available in the iNNvestigate framework [1]. As can be seen in Fig. 6, the majority of the methods are unable to produce reasonable explanations for the chosen

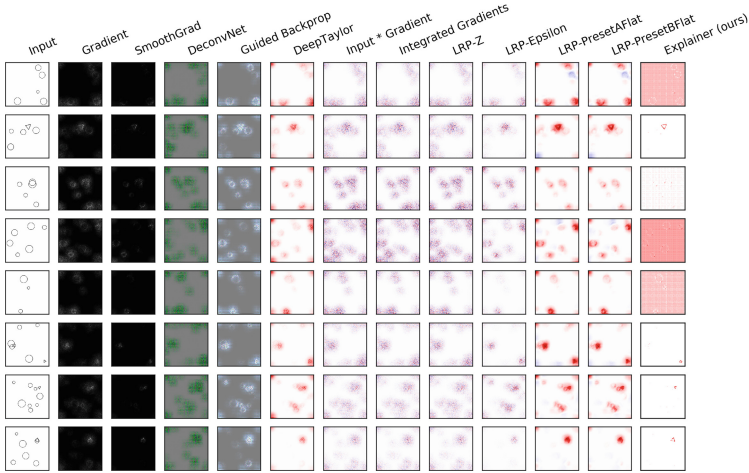




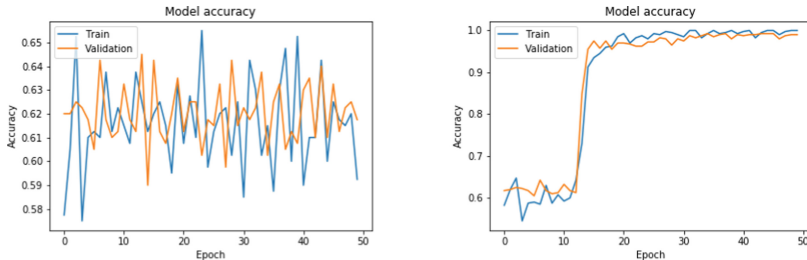
**Fig. 5.** Positive instance and respective explanation. These results were obtained with  $10^{-8}$  (top),  $10^{-6}$  (middle) and  $10^{-4}$  (bottom) L1 regularisation of the explainer’s output while training on a simple dataset without colour cues.

dataset, highlighting corners of the image, for example, while the proposed architecture is able to only highlight the relevant regions for the classifier’s decision (see Fig. 5 middle). Furthermore, training only the classifier, as was done when applying the iNNvestigate toolbox’s analysis methods, yields accuracies close to 62%, while the accuracy of the proposed architecture reaches 100%, as illustrated in Fig. 7. For this dataset, the proposed network not only produces explanations alongside with predictions, as well as improves accuracy, by forcing the classifier to focus only on relevant parts of the image.

Finally, Fig. 8 depicts the obtained results of the experiment on the cue conflict dataset. One can see that the generated explanations are oriented towards semantic components of the objects. For example, for the bottle case the explanations focus more on the neck of the bottle and on its label. In the car example, the explanation highlights the car’s bumper and in the bicycle case, the handles



**Fig. 6.** Results of the application of 10 analysis methods available on the iNNvestigate toolbox [1] to the proposed classifier and comparison with the proposed end-to-end architecture. The color map of the right column’s images was adjusted to help visualization due to the small size of each image and for easier comparison. (Color figure online)



**Fig. 7.** Evolution of classifier accuracy per training epoch for the same classifier trained alone (left) and within the proposed explainer-classifier architecture (right).

and the seat are highlighted, while in the chair example the chair’s legs are highlighted. It is worth noting that although the resulting explanations highlight different semantic components of the objects, they do not appear connected to each other (for example, the handle and the seat of the bicycle). This result hints that improving the quality of these explanations can be made by ensuring that explanations are sparse, i.e., cover a smaller part of the whole image, and also connected.

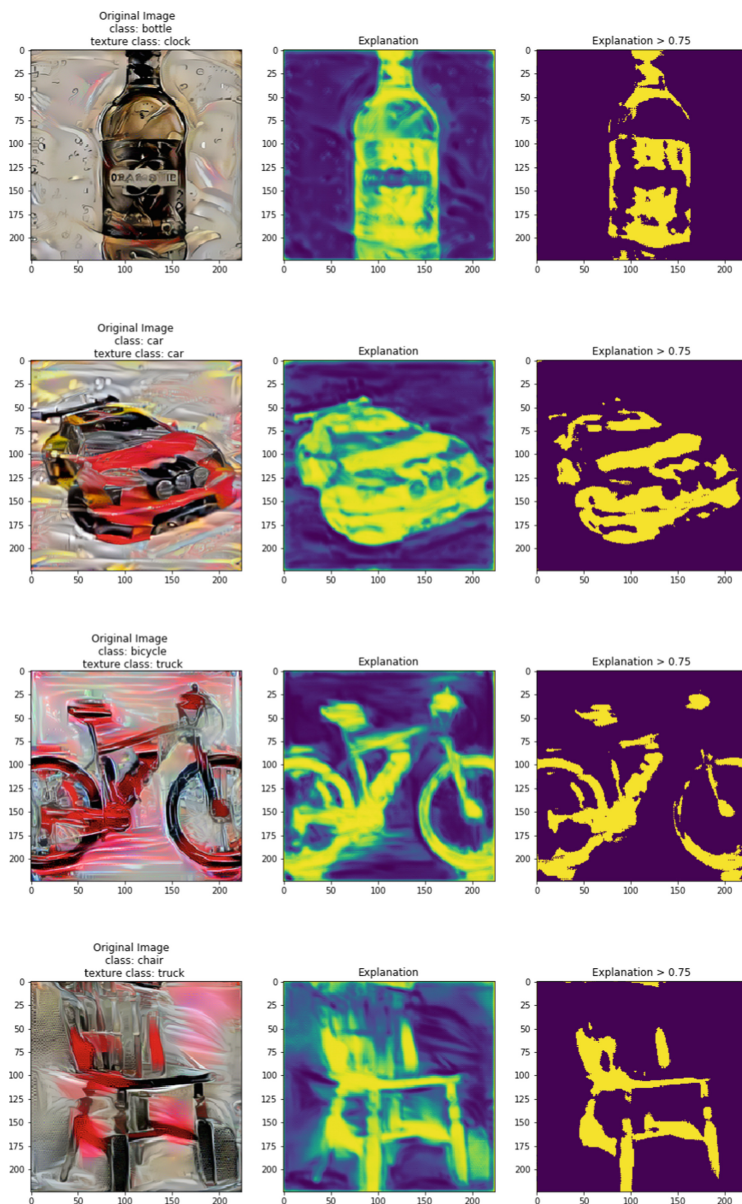


Fig. 8. Results of the cue conflict experiment.

## 4 Conclusion

We propose a preliminary in-model joint approach to produce decisions and explanations using CNNs, capable of producing not only interpretable explana-

tions, but also complete ones, i.e., explanations that are able to describe the system’s internals accurately. We also developed a synthetic dataset generation framework with automatic annotation.

The proposed architecture was tested with a simple generated synthetic dataset, for which explanations are intuitive and do not need to employ expert knowledge. Results show the potential of the proposed architecture, especially when compared to existing methods and when adding L1 regularisation. These also hint at the need for regularisation in order to better balance the interpretability-completeness trade off. As such, future research will study the effect of adding total variation regularisation as a way of making explanations sparse. Also, we will explore the possible advantages of supervising the explanations, as well as develop a proper annotation scheme and evaluation metrics for such task.

## References

1. Alber, M., et al.: iNNvestigate neural networks! (2018)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One* (2015). <https://doi.org/10.1371/journal.pone.0130140>
3. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, November 2018. <http://arxiv.org/abs/1811.12231>
4. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: an overview of interpretability of machine learning. In: *Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics DSAA 2018*, pp. 80–89 (2019). <https://doi.org/10.1109/DSAA.2018.00018>
5. Goodman, B., Flaxman, S.: European Union regulations on algorithmic decision-making and a “right to explanation”, June 2016. <https://doi.org/10.1609/aimag.v38i3.2741>, <https://arxiv.org/abs/1606.08813>
6. Lipton, Z.C.: The Mythos of Model Interpretability (2016). <https://doi.org/10.1145/3233231>
7. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognit.* (2017). <https://doi.org/10.1016/j.patcog.2016.11.008>
8. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* **73**, 1–15 (2018). <https://doi.org/10.1016/J.DSP.2017.10.011>. <https://www.sciencedirect.com/science/article/pii/S1051200417302385>
9. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why Should I Trust You? Explaining the Predictions of Any Classifier, February 2016. <http://arxiv.org/abs/1602.04938>
10. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(11), 2660–2673 (2017)
11. Silva, W., Fernandes, K., Cardoso, J.S.: How to produce complementary explanations using an ensemble model. In: *2019 International Joint Conference on Neural Networks (IJCNN)* (2019)

12. Silva, W., Fernandes, K., Cardoso, M.J., Cardoso, J.S.: Towards Complementary Explanations Using Deep Neural Networks (2018). [https://doi.org/10.1007/978-3-030-02628-8\\_15](https://doi.org/10.1007/978-3-030-02628-8_15)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
14. Smilkov, D., Thorat, N., Kim, B., Vi, F.: SmoothGrad: removing noise by adding noise (2017)
15. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for Simplicity: The All Convolutional Net, December 2014. <https://arxiv.org/abs/1412.6806>
16. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. arXiv preprint [arXiv:1212.5701](https://arxiv.org/abs/1212.5701) (2012)
17. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10590-1\\_53](https://doi.org/10.1007/978-3-319-10590-1_53)