International Journal of Neural Systems, Vol. 30, No. 8 (2020) 2050040 (17 pages)

DOI: 10.1142/S0129065720500409

World Scientific
www.worldscientific.com

Middle-Level Features for the Explanation of Classification Systems by Sparse Dictionary Methods

A. Apicella, F. Isgrò, R. Prevete* and G. Tamburrini
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione
Università degli Studi di Napoli Federico II, 80125 Napoli, Italy
**roberto.prevete@unina.it*

Accepted 17 April 2020
Published Online 14 July 2020

Machine learning (ML) systems are affected by a pervasive lack of transparency. The eXplainable Artificial Intelligence (XAI) research area addresses this problem and the related issue of explaining the behavior of ML systems in terms that are understandable to human beings. In many explanation of XAI approaches, the output of ML systems are explained in terms of low-level features of their inputs. However, these approaches leave a substantive explanatory burden with human users, insofar as the latter are required to map low-level properties into more salient and readily understandable parts of the input. To alleviate this cognitive burden, an alternative model-agnostic framework is proposed here. This framework is instantiated to address explanation problems in the context of ML image classification systems, without relying on pixel relevance maps and other low-level features of the input. More specifically, one obtains sets of middle-level properties of classification inputs that are perceptually salient by applying sparse dictionary learning techniques. These middle-level properties are used as building blocks for explanations of image classifications. The achieved explanations are parsimonious, for their reliance on a limited set of middle-level image properties. And they can be contrastive, because the set of middle-level image properties can be used to explain why the system advanced the proposed classification over other antagonist classifications. In view of its model-agnostic character, the proposed framework is adaptable to a variety of other ML systems and explanation problems.

Keywords: XAI and explainable artificial intelligence; machine learning; sparse coding.

1. Introduction

Machine Learning (ML) approaches have been effectively used to address image[1–3] and text classification[4] problems, multi-target regression,[5] robot navigation,[6] times series forecasting,[7] signal analysis as EEG[8–10] and other major challenges in Artificial Intelligence (AI). Critical aspects of many systems developed on the basis of powerful ML techniques — such as Support Vector Machines (SVM), Probabilistic Neural networks (PNN)[11,12] and Deep Neural Networks (DNN)[13]— are their pervasive lack of transparency and the related difficulty of explaining their behavior in terms that are understandable to human beings. Indeed, it seems that the better ML systems perform in terms of their results and predictions, the harder it is to understand the underlying mechanisms and explain their behaviors.[14] The black-box character[14] of many ML systems is a major predicament in various application domains. To illustrate, consider recent ML work in the healthcare domain which is aimed at developing medical diagnostic tools which identify diseases from biophysical features or medical imaging inputs[15–18] or in the civil engineering domain to identify structural

*Corresponding author.

problems or to infer information from infrastructures data.[19–25] There, explanations of system outcomes would be useful to identify undesirable artifacts and biases in training sets, to build trust in their responses and, for example, to prescribe confidently related clinical treatments in the healthcare domain. Thus, generating explanations for ML system behaviors that are *understandable to human beings* is a central scientific and technological issue addressed in the rapidly growing AI research area of eXplainable Artificial Intelligence (XAI). Several notions of interpretability and explainability for ML systems have been proposed in XAI.[26] And various approaches have been pursued to make ML systems increasingly interpretable and explainable.[27–29] XAI approaches to the explanation problem are usefully grouped into *global* or *local* approaches. In the former case, the goal is to produce a single explanation for the whole behavior of the ML system, in the latter case, for each given input, one has to find a specific explanation. In both cases, explanations are based on collections of features from images (arrays of pixels), texts (sequences of words) or other humanly interpretable domains.[29]

We focus here on local explanation problems using a *model agnostic* approach.[30] This approach is independent from any internal feature of ML engines, for the latter is treated as a black-box. Hence, model-agnostic solutions to explanation problems that one finds for some given ML system are more easily transferred to other ML systems. Furthermore, the model-agnostic solutions to explanation problems that we explore here are meant to mitigate complementary defects of other model-agnostic explanations, that are based on high-level and low-level image features, respectively. In *high-level* explanations, a system output is explained by providing a *class prototype* of input data.[27–29,31] For example, if an image x is classified as "fox", and the corresponding explanation request is "why is x a fox?", one answers this request by exhibiting a fox-prototype, and by asserting that the input image resembles the prototype. This is "because it looks like this" response is often insufficiently informative to be counted as a good explanation, for it leaves with human users, the burden of identifying middle-level features (parts) of the prototype and matching them with middle-level (parts) of the input image x. In *low-level* explanations, a system output is explained by appeal to low-level features of the input image. To illustrate, consider the explanation-generating method called layer-wise relevance propagation.[32,33] The key idea is to back-propagate the classifier output, by associating a relevance value to each input element (to each pixel in the case of images). This value indicates how much that input element (pixel) contributed to obtain the classification result for the overall input (image). In this case too, human users are left with a significant interpretive burden: starting from the relevance values of each input element (pixel), one has to identify properties of the overall input that are perceptually salient for the human visual system.

To alleviate these complementary defects of low-level and high-level approaches to explanation, we focus here on explanations of image classifications that are built starting from perceptually salient parts of the input.[30] In the case of an image x classified as "fox", and a corresponding explanation request "why is x a fox?", salient image parts are those associated with a fox tail, nose, ears parts, and so on. We refer to salient image parts as *middle-level* properties of input images, to distinguish them from both high-level image prototypes and low-level image features as, for instance, single pixels. As we shall see, the present approach enables one to isolate humanly understandable explanations that are *parsimonious*, insofar as they are based on a restricted list of middle-level properties, excluding elements that hardly help humans to understand why that classification result was achieved.[34] To this end, we make use of sparse dictionary learning techniques. These techniques provide data representations in terms of sparse linear combinations of elements (atoms) from a dictionary which is learned from data. These atoms often are found to be directly interpretable by humans.[35] As we discuss extensively in Sec. 3.3, according to the present approach explanations are based on structured collections of dictionary atoms which better reconstruct the input and maximize the probability to obtain the class associated with the input.

It has been emphasized that good explanations often take the form of *contrastive* explanations,[34,36] that is, of explanations including reasons why that solution was offered by the system and why some alternative solution was not offered. To illustrate, when an input image is classified as fox, a contrastive explanation should provide reasons for this

classification outcome and reasons for excluding competitive classifications, so as to answer such questions as "Why was this input image *not* classified as a cat?". The present model-agnostic framework can be used to provide explanations answering both basic explanation requests ("why this outcome?") and contrastive explanation requests ("why this outcome rather than this other one?").

This paper builds upon and extends previous work.[37–39] However, some major novelties are introduced here. First, we now distinguish sharply between a general functional architecture which captures many of the XAI approaches proposed in the literature and our specific approach, which instantiates this architecture by imposing specific constraints on each one of its functional components (see Sec. 3.1). Consequently, our approach is more organically presented, and more thoroughly compared with other approaches. Equally important, two main advances are made here with respect to both dictionary building procedures (see Secs. 3.2 and 5). learning procedures used to find conventional and contrastive explanations (see Sec. 3.3). Finally, a more unified approach is introduced in the experimental framework, insofar as the same datasets are used to build both conventional and contrastive explanations, in order to obtain qualitatively more comparable results.

The paper is organised as follows: Section 2 briefly reviews related approaches; Sec. 3 describes the proposed model-agnostic architecture; experiments and results are discussed in Sec. 4; the concluding Sec. 5 summarizes the main high-level features of the proposed explanation framework and outlines some future developments.

2. Related Work

The widespread need for transparency and explainability in AI is discussed in many papers,[34,40–42] and many different methods have been proposed over the years.[31,43–46] Depending on the task to be performed, explainability may become a fundamental requirement for any adequate AI solution. Thus, the lack of transparency and explainability in ML models may become a major predicament in various AI application domains. In some approaches to transparency and explainability issues, one tries to extract rules from some trained system to infer the learned behaviors.[13,47–49] In other approaches, one builds more interpretable surrogate models which approximate the original ones.[50–54] In other approaches yet, one analyzes the relationship of the model's output to the originating input. These latter methods are informatively divided[55] into two main categories: *perturbation-based* and *backpropagation-based*[32,33,56–58] methods, depending on whether the obtained maps describe how the system's output changes on input perturbations or how much each input variable contributes to the output. In backpropagation-based methods, one usually relies on the idea that a classifier response is backpropagated to the input layer of the classifier, so that one assigns to each input element (each pixel in the case of images) a relevance value (pixel-wise decomposition paradigm), which estimates how much the input element has contributed to obtain the corresponding classification response. Thus, the output of this analysis process is a *relevance map* (or *contribution map* or *attribution map*[55]) which belongs to the input domain. Major techniques for analysing the relationship of the model's output to the originating input are Layer-Wise Relevance propagation and Deep Taylor Decomposition,[32,33,58] where the back-propagation process has to satisfy some conservation rules.[32] These methods can be applied to a wide range of nonlinear classification architectures, which notably include DNN and SVM. However, these methods are not model-agnostic. Moreover, a low-level analysis of the input relevance to the system responses is made, thus providing as an explanation relevance maps which represent an evaluation of low-level input properties. Accordingly, most of the interpretive process is left to the human capacity to identify salient input middle-level properties starting from low-level relevance maps. Similar approaches have been pursued in Zintgraf *et al.*,[59] and Robnik-Šikonja and Kononenko,[60] where an estimate is proposed of the importance of individual features for the classifier's response. Alternative explanations have been given in terms of humanly interpretable sentences associated with the input data[61] or approximate input reconstructions (also named class prototypes) which the input data can be associated with.[29] Many of these approaches are based on the *Activation Maximization* (AM) method.[27] In a nutshell, this method enables one to find the input that maximizes the output of a neural unit. Several approaches

based on AM attempted to enhance the produced results in terms of explainability, e.g. by bounding the solution search space in a human *interpretable* domain, using image priors[28,31,62,63] to avoid useless solutions. In Nguyen *et al.*,[64] a useful survey of AM methods is given. In this case too, as already pointed out in Sec. 1, much of the interpretive work must be performed by human users, who must isolate the input middle-level properties which determined the answer of the classifier.

Similar approaches to interpretability were proposed in the context of Convolutional Neural Networks (CNN), as Deconvolutional Network (already presented in Zeiler *et al.*,[65] as a way to do unsupervised learning) and *Up-convolutional network*[66,67] unlike our proposed model, which can apply in principle to any classifier (model agnostic), these proposed approaches seem to be model-specific for CNN. A major model-agnostic model is the explainer LIME.[30] LIME takes into account the model behavior in the proximity of the instance being predicted, partitioning it in a collection of *components* (super-pixel in the image case). Thus, LIME builds a more straightforward model from which it is possible to infer an explanation of the original model behavior. LIME's outputs have some similarities with the outputs of the system proposed here, insofar as it provides explanations in terms of middle-level properties of the input. However, in the LIME framework, it is not easy to find a kernel of "common components" shared between different input elements. The authors proposed to use super-pixels as essential components of the input image, but this solution limits the possibility of comparing produced explanations, due to the fact that different super-pixels sets produced by different images. Additional methods based on LIME, as in Ribeiro *et al.*,[68] and Guidotti *et al.*,[69] return explanations in terms of decision rules that are used as local conditions for decisions.

Importantly, to the best of our knowledge, *contrastive explanations* have received much less attention in the literature than conventional explanations, i.e. explanations conceived as appropriate answers to questions of the following type: "why this outcome??". As pointed out in Sec. 1, by the expression "contrastive explanation" we mean explanations that provide answers to questions of the following type: "why this outcome? And why not these other outcomes?". Currently, in this direction, the method proposed in Guidotti *et al.*,[69] provides explaination also in counterfactual terms while a model-specific method was proposed in Zhang *et al.*,[70] which works on CNN architectures. Instead, our approach to contrastive explanations is model-agnostic.

3. Proposed Approach

This section, which provides an overall description of our approach, starts from a functional description, given in the next section, which is cast in terms of three interacting functional entities ot modules. Then, we describe an implementation of the three functional modules which enables one to obtain both conventional and contrastive explanations.

3.1. *Functional description*

The proposed explanation approach is based on a general functional architecture comprising three processing entities or modules (Fig. 1): (a) the *Oracle*, an ML system, e.g. a classifier, whose inner mechanism is not necessarily known; (b) the *Interrogator*, typically a human being, requesting explanations about the Oracle's responses; (c) the *Mediator*, helping the Interrogator to understand the Oracle's behavior by providing explanations of Oracle's responses, possibly taking advantage of the support of some background knowledge. The Interrogator has access to both inputs and outputs of the Oracle. Furthermore, it interacts with the Mediator.

The Mediator, in addition to interacting with the Interrogator, interacts with the Oracle too. The Mediator interacts with the Oracle in two distinct forms. It may consider the Oracle as a black-box (Fig. 1, continuous line) or else it may have access to its internal operations (Fig. 1, dotted line). The former interaction mode corresponds to model-agnostic approaches, while the latter one corresponds to model-specific approaches. The Mediator fulfils the crucial explanatory role, by advancing hypotheses on what humanly interpretable elements are likely to have influenced the Oracle output. In other words, the mediator role is to provide explanations to the interrogator using, if necessary, background knowledge together with the output given by the Oracle. The double-headed arrow between Oracle and Mediator in Fig. 1 indicates that the latter can "ask" new questions and get an answer from the Oracle. More specifically, in our case, given an Oracle's

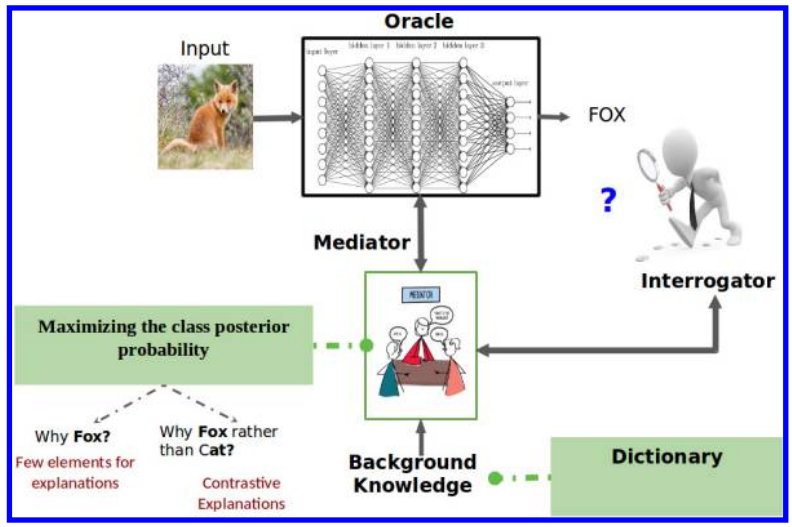


Fig. 1. The three-entity proposed framework. See text for details.

input image $\mathbf{x} \in R^d$ and the Oracle's response $\hat{c}$, as input the Mediator receives a request of conventional or/and contrastive explanation and the pair $(x, \hat{c})$. Its output is a collection of atoms $\{V_i \in R^d\}$ which are selected from a pre-learned dictionary V. This Mediator output is built based on Oracle's outputs when it receives as input atom-composed images during the iterative explanation-building process.

Thus, explanations are provided by a system (the Mediator) which does not coincide with the system (the Oracle) whose behavior the Interrogator needs to be explained.

Given this three-entity functional architecture described above, our approach instantiates this architecture by imposing specific constraints on each one of its functional entities (see Fig. 1):

- Oracle: any image classifier whose output can be interpreted as the posterior probability of the class given the input;
- Interrogator: any user authorized to request explanations; the Interrogator can choose whether to ask for conventional or contrastive explanations.
- Background Knowledge: an expressive dictionary composed of elements that are interpretable by the Interrogator. The elements of the dictionary (atoms) provide middle-level input features.
- Mediator: It cannot access the internal operations of the Oracle (model agnostic approach). The Mediator can provide both standard and contrastive explanations in terms of collections of dictionary elements.

3.2. *Dictionary-based background knowledge*

Following Gilpin *et al.*,[71] and Miller,[34] we consider an explanation as humanly understandable when it uses a few elements extracted from a dictionary of items that are meaningful to the user. From now on, we call this type of explanation framework *dictionary-based explanation framework*. To achieve dictionary-based explanations, we take advantage in our approach of sparse dictionary learning methods.[72–74] As already mentioned in Sec. 1, using these methods, one may obtain data representations in terms of sparse linear combinations of sparse essential elements. The coefficients of the linear combination are referred to as *sparse coding*. The essential elements, usually known as *atoms*, compose the dictionary. These atoms often exhibit satisfactory interpretability levels[35,73,75,76] as they enable one to highlight local aspects of the data corresponding to middle-level properties.

Formally, a sparse dictionary learning problem is a minimization problem that one can describe as follows. Given a set $\{\mathbf{x}^{(i)}\}_{i=1}^n$, where each $\mathbf{x}^{(i)} \in \mathbb{R}^d$ is a column vector representing an experimental observation, these elements can be arranged column-wise in a matrix $X \in R^{d \times n}$. Then, the learning problem can be solved by finding a matrix $V \in R^{d \times k}$ such that each column of X can be approximated by a linear combination of the k columns of V, subjects to some sparsity constraint. V is the dictionary, and the k columns $\mathbf{v}^{(i)}$ of V are the dictionary elements

or atoms, subject to some sparsity constraint in turn. Let us call $H \in R^{k \times n}$ the matrix of the linear combination coefficients, i.e. the ith column of H, $\mathbf{h}^{(i)}$, corresponds to the k coefficients of the linear combination of the k columns of V to approximate $\mathbf{x}^{(i)}$, the ith column of X. Consequently, VH is an approximation of X, where V and H can be both subject to sparsity constraints. In a general formulation,[73] this learning problem can be expressed formally as

$$\arg\min_{H,V} ||X - VH||_F^2 + \gamma_1 \sum_{i=1}^{k} \Omega_V(\mathbf{v}_i) \quad \text{s.t } \forall j, \Omega_H(\mathbf{h}_i) < \gamma_2, \tag{1}$$

where Ω_V and Ω_H are some norms or quasi-norms that constrain or regularize the solutions of the minimization problem, and $\gamma_1 \geq 0$ and $\gamma_2 \geq 0$ are parameters that control to what extent the dictionary and the coefficients are regularized.

In this paper, we obtain dictionaries from a specific sparse dictionary learning method based on Nonnegative Matrix Factorization (NMF).[77] However, one may use any other dictionary learning/sparse coding method producing dictionaries that are sufficiently understandable to humans. These dictionaries constitute the background knowledge of the proposed explanation framework. In our knowledge, the literature does not supply yet procedures capable to build dictionaries composed of atoms that are understandable/interpretable to human users. A major difficulty towards general solutions to this problem is the imprecise and subjective character of the idea of "understandable" atom. Indeed, what is understandable for some person may not necessarily be understandable for someone else. In this paper, we pursue the more modest goal of showing experimentally that established algorithms, such as NMF with sparsity constraints (NMFSC),[78] do in fact produce dictionaries which enjoy the requested property of being understandable to human users. Furthermore, NMFSC builds dictionaries which respect the *nonnegativity* constraint, ensuring only additive operations in data representations. Nonnegativity guarantees an improved human understanding, since it allows one to obtain atoms which are composed of only positive real values, so that they are easily interpreted as parts of the input images. Moreover, data reconstruction is a purely additive linear combination of atoms so that each reconstruction can be viewed as a positive superposition of parts of the input images.

3.3. *Conventional explanation*

To obtain a data representation which helps the Interpreter to explain the decision made by the Oracle, we search for a dictionary-based encoding which maximizes the Oracle answer by being both "similar enough" to the given input and sparse enough to ensure that few dictionary atoms are used. Therefore, instead of returning only an accurate input representation based on dictionary atoms, we propose a method that returns a subset of dictionary's atoms taking into account the Oracle choice.

More formally, we optimize the following objective function:

$$\mathbf{h}_{c^*} = \arg\max_{\mathbf{h} \geq 0} \log \Pr(\hat{c}|V\mathbf{h}) + \lambda_1 ||V\mathbf{h} - \mathbf{x}||_2 + \lambda_2 S(\mathbf{h}), \tag{2}$$

where $\Pr(\hat{c} \,|\, \mathbf{x})$ is the probability given by the Oracle that input $\mathbf{x}$ belongs to the class $\hat{c}$, V is the dictionary chosen as background knowledge, and $S(\cdot)$ is a sparsity measure. The λ_1 hyper-parameter is used to control the proximity of the reconstruction $V\mathbf{h}$ to the original input $\mathbf{x}$ leading the optimization algorithm to prefer atoms which, combined together, can result in a good input representation, while λ_2 influences the encoding sparsity, that is "how many" atoms the reconstruction effectively uses. To avoid having to manually search for good values of λ_1 and λ_2, we adopted an update rule similar to those used for adaptive learning rate,[79] which can be formalized as follows:

$$\lambda_1^t = \begin{cases} (1-\alpha)\lambda_1^{t-1} & \text{if } |l^t - l^{t-1}| < \epsilon \\ (1+\alpha)\lambda_1^{t-1} & \text{otherwise,} \end{cases} \tag{3}$$

where l^t is the value of the objective function at the tth iteration in the AM-like procedure, α is a small factor (in our case, we set $\alpha = 0.01$) and $\lambda_2^t = 1 - \lambda_1^t$.

Furthermore, the $\mathbf{h} \geq 0$ constraint ensures that one obtains the output in a purely additive form. Let us call the complete architecture, depicted in Fig. 1, Explanation Maximization (EM).

In the final output vector $\mathbf{h}_{c*}$ each component $h_i, i = 1, 2, \ldots, d$ can be interpreted as a measure of the "importance" of the ith atom in the result that the Oracle associates with the input $\mathbf{x}$.

Algorithm 1. Explanation Maximization procedure (EMExplanationBuilder)

Input: data point $\mathbf{x} \in \mathbb{R}^d$, the output class $\hat{c}$,learned model Γ, a dictionary $V \in \mathbb{R}^{d\times k}$, $\lambda_1, \lambda_2, \alpha, \epsilon$
Output: the encoding $\mathbf{h} \in \mathbb{R}^d$

$\mathbf{h} \sim U^d(0,1)$;
while $\neg$ *converge* **do**
  $\vec{r} \leftarrow V\vec{h}$;
  $\vec{h} \leftarrow \arg\max_{\vec{h}} \log \Pr\left(\hat{c}|\vec{r};\Gamma\right) + \lambda_1||\vec{r} - \vec{x}||_2$;
  $\vec{h} \leftarrow \text{proj}(\vec{h}, \lambda_2)$ ▷ for proj$(\cdot,\cdot)$ see Hoyer;[78]
  if *loss difference* $< \epsilon$ **then**
    $\lambda_1 \leftarrow (1-\alpha)\lambda_1$;
  else
    $\lambda_1 \leftarrow (1+\alpha)\lambda_1$;
  end
  $\lambda_2 \leftarrow 1 - \lambda_1$;
  $l^{prec} = l$;
end
return $\vec{h}$;

Equation (2) can be solved by combining any standard gradient ascent technique with a projection operator[78] that ensures both sparsity and non-negativity. The complete procedure is reported in Algorithm 1.

3.4. *Contrastive explanation*

The procedure described in Sec. 3.3 builds conventional explanations, that is, explanations providing an answer to questions of type "why does the Oracle output c^* on input "x"?". In this section, we present a method using this procedure to produce explanations in contrastive terms, that is, explanations providing an answer to questions of type "why does the Oracle returns the class c^* and why not the alternative class $\overline{c}$ on input "x"?". The procedure described in Sec. 3.3 can be easily adapted to this purpose, by maximizing the Oracle probability of some given contrastive class, rather than the probability of the predicted class. In other terms, one searches for a proper subset of atoms and relative encoding values whose combination is again both similar to the input and sparse enough, but that leads the Oracle to provide a different outcome if fed to it. On this basis, one can develop a contrastive explanation by inspecting the difference between the atoms in the explanations generated for the same input but pushing the Oracle's answer toward different classes. Let us consider, for example, that we have as Oracle, a classifier trained to recognize images of different sorts of clothes. The Oracle correctly classifies an input image x as a t-shirt, and we want to know why the Oracle does not classify the input as a jumper. Let us consider then a second input image y that is very *similar* to x, but is classified as a jumper by the Oracle. We may expect that the result of a conventional explanation answering 'why is x a t-shirt?' should differ from a conventional explanation answering 'why is y a jumper?' by some atoms representing long sleeves in the jumper instead of the short sleeves which we expect to appear in the t-shirt explanation.

More formally, we search for two encodings h_{c^*} and $h_{\overline{c}}$ such that

$$\mathbf{h}_{c^*} = \arg\max_{\mathbf{h}\geq 0} \log \Pr(c^*|V\mathbf{h}) + \lambda_1||V\mathbf{h} - \mathbf{x}||_2 + \lambda_2 S(\mathbf{h}) \quad (4)$$

$$\mathbf{h}_{\overline{c}} = \arg\max_{\mathbf{h}\geq 0} \log \Pr(\overline{c}|V\mathbf{h}) + \lambda_1||V\mathbf{h} + \mathbf{x}||_2 + \lambda_2 S(\mathbf{h}) \quad (5)$$

where c^* is the real classifier outcome for the input $\mathbf{x}$ and $\overline{c} \neq c^*$ is the contrastive class.

4. Experimental Assessment

To test our framework, we chose as Oracle the LeNet-5[80] CNN architecture, commonly used for recognition tasks on simple datasets. We trained the

Algorithm 2. Contrastive Explanation Maximization procedure

Input: data point $\vec{x} \in \mathbb{R}^d$, the number of antagonist classes q, the Oracle Ω, a dictionary $V \in \mathbb{R}^{d\times k}$
Output: the encoding $\vec{h} \in \mathbb{R}^d$

$\vec{p} \leftarrow$ getClassProbabilities $(\vec{x}, \Omega)$;
$(c_1, c_2, \ldots, c_{q+1}) \leftarrow$ getBestClasses $(\vec{p},\ q+1)$;
$\vec{h}_{expl} \leftarrow$ EMExplanationBuilder $(\vec{x},\ c_1, \Omega, V)$;
for $i = 2$ ***to*** $q+1$ **do**
  $\vec{h}^{(i)}_{anta} \leftarrow$ EMExplanationBuilder $(\vec{x}, c_i, \Omega, V)$;
end
return $\vec{h}_{expl}, \vec{h}^{(2)}_{anta}, \ldots, \vec{h}^{(q+1)}_{anta}$

network from scratch using two different datasets: MNIST[80] and Fashion-MNIST,[81] obtaining an accuracy of 98.86% and 91.43% on the test sets, respectively. Each dataset is composed of 28×28 grayscale images belonging to one of 10 possible classes. These classes are digits for MNIST and clothes for Fashion MNIST. The training sets and the test sets are composed, respectively, of $50,000$ images and $10,000$ images for each dataset; the model is learned using the Adam algorithm.[82]

An hyper-parameter needed by NMFSC to build sparse dictionary atoms is the number of atoms which compose the dictionary. In our cases, we set it to 200, relying on PCA analysis which revealed that at least 100 principal components are needed to explain more than 95% of the data variance. NMFSC necessitates two further hyperparameters, γ_1 and γ_2, which control the sparsity on the dictionary and the encoding, respectively. A proper sparsity on the dictionary can be useful to obtain atoms which highlight a local region of the input; however, an exceedingly large sparsity index may lead to atoms that are hard to understand by humans (e.g. a cloud of points or atoms composed by sparse points).

We claim that a *good* dictionary for our proposed procedure must satisfy the following properties: (i) a low reconstruction error, so that it represents the data reliably; (ii) a high sparsity on the encoding, so that it uses only a small number of atoms to represent the data; (iii) sparsity on the atoms, so that each atom may capture middle-level features of the input; (iv) atoms must be as distinct as possible from each other, so that each atom represents a different (distinct) input feature.

We constructed different dictionaries with different sparsity values in the range $\gamma_1, \gamma_2 \in [0.6, 0.8]$[78] and then we selected those dictionaries showing a good trade-off between reconstruction error and sparsity level. However, this method of selecting the dictionaries may present a drawback: it allows for the presence of atoms that are too similar to each other violating the last property above mentioned. This condition might lead the explanation system to spread the influence on multiple similar atoms, thereby biasing the explanation.

To compensate for this drawback, one may think of adding a constraint to ensure the atoms dissimilarity during the creation of the dictionary. But this procedure might be too expansive in computational terms. For this reason, we prefer to apply a simple *a posteriori* clustering procedure to clean the data, assuming that visually similar atoms must be close to each other in terms of Euclidean distance, thereby increasing the likelihood of their falling in the same cluster. More in detail, after a dictionary is created, we apply a k-medoids clustering,[83] using just the selected k-medoids as dictionary and discarding the other atoms. In our experiment, we obtain good results by setting $k = 25$, which results in a dictionary whose atoms are very dissimilar from each other. To give a quantitative measure of how different the explanation elements are, we compute the mean distance between the atoms composing the explanations of 10 random inputs using both a full dictionary without any post-processing procedure and a dictionary selected using k-medoids clustering. The results are reported in the following Table 1, showing then that using a reduced dictionary produces explanations whose atoms are more different from each other on average.

Table 1. Mean distance between the atoms composing the explanations.

Atoms' mean distance	
Without post-processing	Using k-medoids
2.3 ± 0.1	2.8 ± 0.2

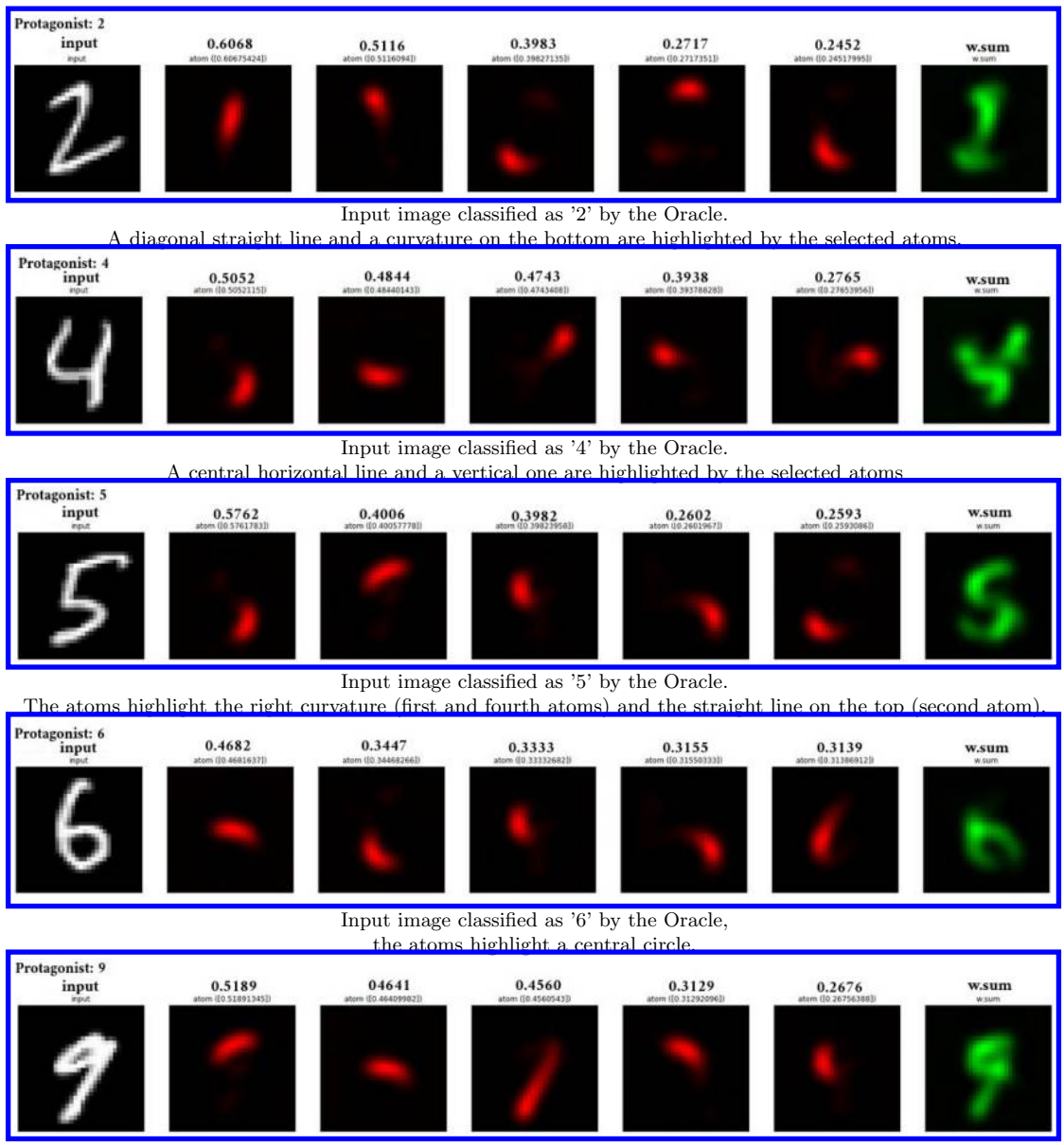
4.1. *Results*

In this paragraph, we show a set of explanations generated by the proposed Conventional Explanation Maximization procedure and Contrastive Explanation Maximization procedure, using images taken from the MNIST and Fashion MNIST test sets and the corresponding Oracle's answers.

4.1.1. *Conventional EM explanations*

We build our explanation from a selection of atoms with larger encoding values (i.e., those that are more important in the representation, since we enforce nonnegativity in the solution of Eq. (2)).

In Fig. 2, we show the atoms producing the explanation on some input images taken from the MNIST dataset for which the Oracle gave the correct answer. The selected atoms arguably describe the



Input image classified as '2' by the Oracle.
A diagonal straight line and a curvature on the bottom are highlighted by the selected atoms.

Input image classified as '4' by the Oracle.
A central horizontal line and a vertical one are highlighted by the selected atoms.

Input image classified as '5' by the Oracle.
The atoms highlight the right curvature (first and fourth atoms) and the straight line on the top (second atom).

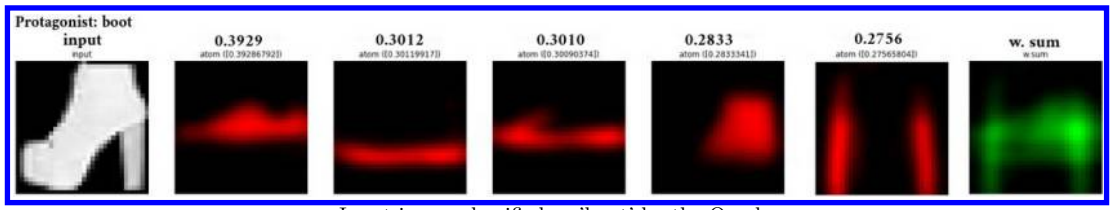
Input image classified as '6' by the Oracle,
the atoms highlight a central circle.

Input image classified as '9' by the Oracle,
the atoms highlight a top circle and a straight diagonal line (third atom).

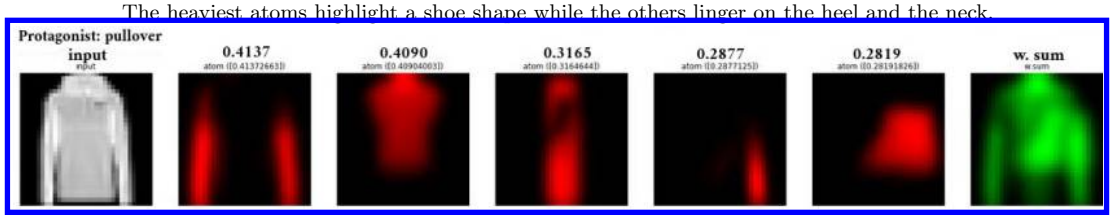
Fig. 2. (Color online) Explanations on images taken from the MNIST dataset. The five atoms with the largest encoding values (red columns) and their linear combination weighted (green column) on the encoding values (displayed on top of each atom) provide an explanation for the answer given by the Oracle on the input image (first column). The weighted sum shows that the chosen atoms are sufficient to allow for a human interpretation of the given answer. The parts highlighted by the selected atoms suggest that these parts are essential to the Oracle's decision.

visual impact of the input digits in a thorough fashion, by providing elements that appear to be discriminative, such as the intersection between a horizontal line with a vertical one for the number 4, or the circle on the bottom for number 6, and the straight top part together with the curved bottom one for 5. To probe the impact of sparsity on this representation empirically, we performed the same experiment using a dictionary with a very low sparsity (0.1), obtaining encodings without any prevailing value, thereby making it challenging to select appropriate atoms for an explanation.

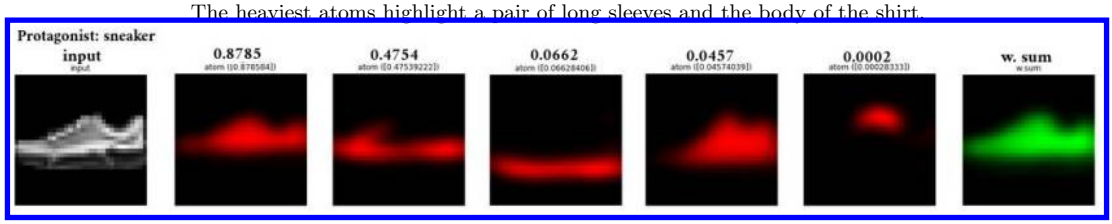
In Fig. 3, we show the more important atoms obtained on some input images taken from the



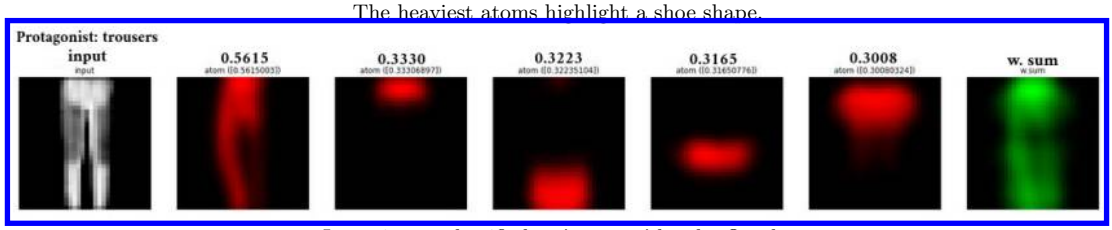
Input image classified as 'boot' by the Oracle.
The heaviest atoms highlight a shoe shape while the others linger on the heel and the neck.



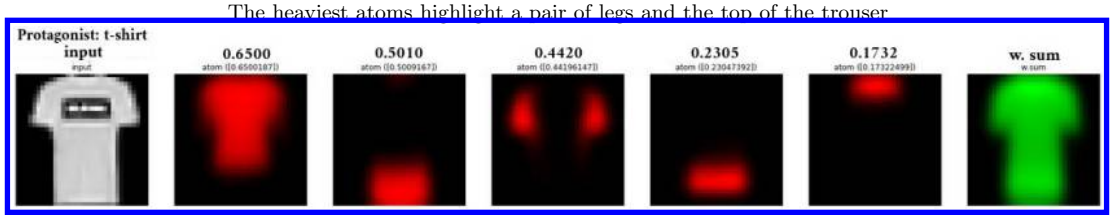
Input image classified as 'jumper' by the Oracle.
The heaviest atoms highlight a pair of long sleeves and the body of the shirt.



Input image classified as 'sneaker' by the Oracle.
The heaviest atoms highlight a shoe shape.



Input image classified as 'trousers' by the Oracle.
The heaviest atoms highlight a pair of legs and the top of the trouser.



Input image classified as 't-shirt' by the Oracle.
The heaviest atoms highlight the body of a shirt and a pair of short sleeves

Fig. 3. (Color online) Explanations on images taken from the Fashion MNIST dataset. The five atoms with largest encoding values (red columns) and their linear combination weighted (green column) on the encoding values (displayed on top of each atom) provide an explanation for the answer given by the Oracle on the input image (first column). The weighted sum shows that the chosen atoms are sufficient to allow for a human interpretation of the given answer. The parts highlighted by the selected atoms suggest that these parts are essential to the Oracle's decision.

Fashion MNIST dataset, all of them correctly classified by the Oracle. Selecting the atoms with more significant encoding values seems to enhance representative parts of the input image that can be easily interpreted by a human interrogator, (e.g. the neck and the sole for a boot, the long sleeves for a pullover and the short ones for a t-shirt).

As for MNIST, we performed the same experiment using a dictionary with low sparsity, ending up with results that are difficult to interpret.

4.1.2. *Contrastive EM explanations*

Figures 4 and 5 show a set of explanations produced by the Contrastive EM framework using

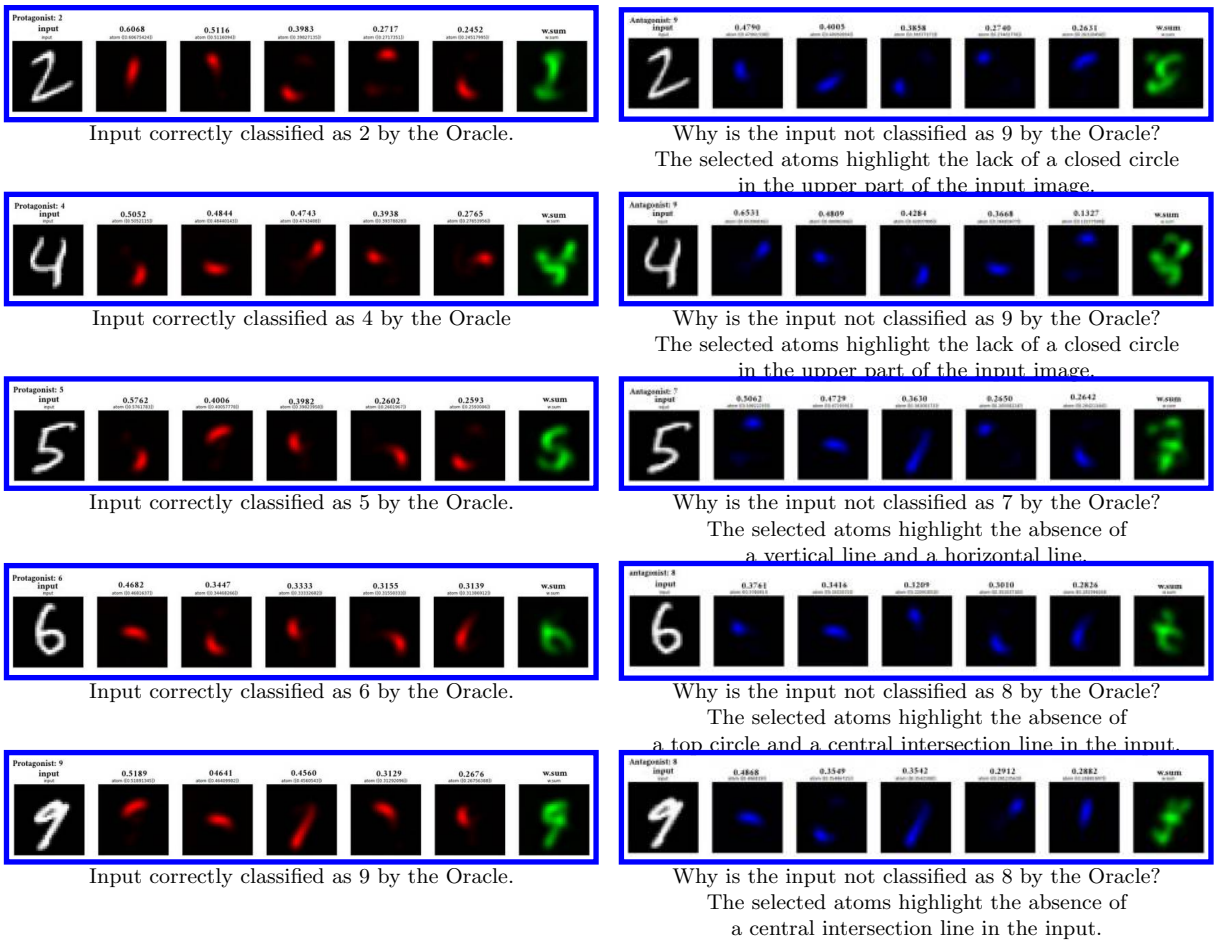


Fig. 4. (Color online) Examples of EM contrastive explanations obtained on images taken from the MNIST dataset. The explanation for the answer given by the Oracle on each input is expressed with two different sets of atoms: the first one (in red) highlights why the input is classified as the given Oracle outcome, the second one (in blue) highlights what features should the input exhibit to be classified as the antagonist class.

different input images taken from MNIST and Fashion MNIST dataset. Each explanation is expressed in terms of two different subsets of atoms which were selected by computing h_{c*} and $h_{\overline{c}}$, respectively, as described in Sec. 3.4. The first selected subset is formed by the atoms which most contribute to the Oracle outcome(as the standard EM procedure described in Sec. 3.3). The second one is formed by the atoms which most contribute to some given contrastive outcome. Only the first five atoms are shown.

As already discussed in Sec. 3.3, the atoms selected by h_{c*} (in red in the figures) provide elements which can be considered discriminative for the selected outcome. For example, in Fig. 4 for an input correctly classified as a 4, EM selects several components which represent an intersection between a horizontal and a vertical line, showing that this is likely to be one of the main features used by the classifier to make its choice. The second set (blue) is computed by choosing as contrast class a 9, and asking the algorithm to provide an explanation. One can see that the selected components are mostly different and varied, showing that the input image (representing a 4), to be classified as a 9, should also have other characteristics, including a further horizontal line on the top, which helps to generate the typical circular shape that one finds in the top of a 9 digit. Similar considerations can be made for an input representing a 5 compared to the desired outcome of a 7. In this case, the classifier's choice of a 5 might be motivated by the presence of the components shown in red, whereas the the total absence

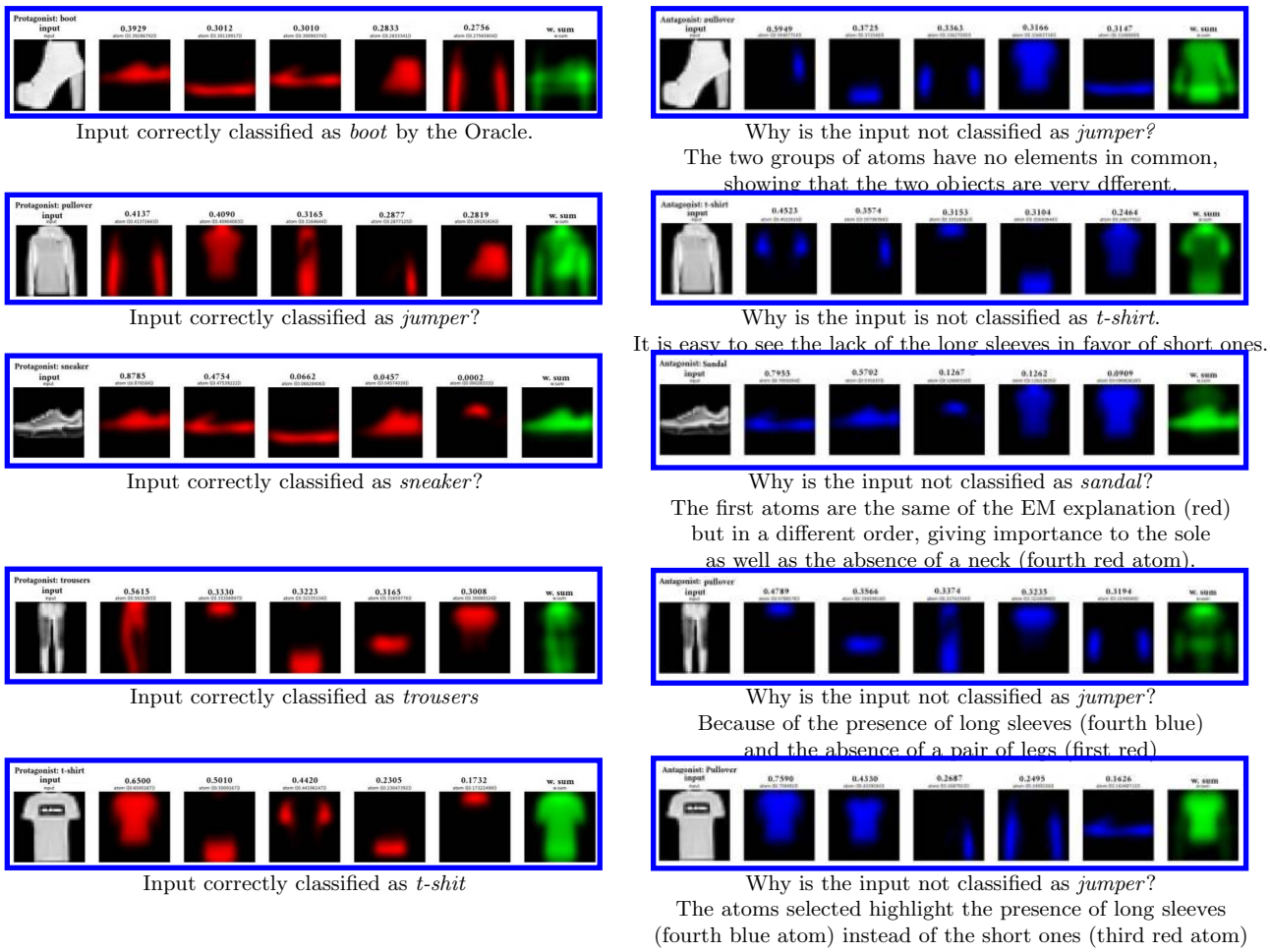


Fig. 5. (Color online) Examples of EM contrastive explanations obtained on images taken from the Fashion MNIST dataset. The explanation for the answer given by the Oracle on each input is expressed with two different sets of atoms: the first one (in red) highlights why the input is classified as the given Oracle outcome, the second one (in blue) highlights what features should the input exhibit to be classified as the antagonist class.

of components highlighted by the blue atoms, such as the central horizontal line on the left side, suggest that the absence of this feature from the input image can be a good explanation of why it was not been classified as a 7 by the given Oracle.

Similar considerations can be made about the Fashion MNIST dataset in Fig. 5. For example, an input correctly classified as *jumper* (second row) is explained by the presence of long sleeves (red atoms), while the chosen contrastive class *t-shirt* highlights the absence of short sleeves (blue atoms). The same difference, albeit with inverted roles, can be used to explain an input correctly classified as *t-shirt*, when compared to the contrastive class *jumper* (fifth row).

5. Conclusions

A model-agnostic framework was introduced here to build humanly interpretable explanations for the output provided by a black-box image classifier which is fed with some given input. The returned explanations are both conventional and contrastive. Moreover, they are obtained in terms of what we have called middle-level input properties, expressing perceptually salient features for the human visual system, rather than in terms of input low-level properties, such as pixel relevance maps. And the returned explanations are parsimonious, respecting the constraint that a good explanation should use few elements extracted from dictionaries that are meaningful to human users.

Our framework relies on a three-entities or three-modules functional model, which is composed of an Oracle (providing answers to be explained), an Interrogator (posing explanation requests), and a Mediator (helping the Interrogator to interpret the Oracle's outcomes using elements given by Background Knowledge). The Background Knowledge is, in our case, a dictionary built using well-established techniques of sparse dictionary learning, such as NMF. Note that the more common dictionary learning methods do not pay attention to factors as the "redundancy" on the obtained atoms. For this reason, a clustering procedure was applied to the obtained dictionary to clean it from redundant atoms. However, any other dictionary learning technique that meets the requirements sketched in Sec. 3 may be successfully used. Regarding the computational complexity of our approach, we note that it consists of two distinct phases: (1) the dictionary learning phase, and (2) the explanation construction phase. The computational complexity of the former depends on both the selected dictionary learning method and the dataset size; however, this run only once and has no impact on the actual construction of the explanation. The computational complexity of the explanation construction is an iterative process, and its cost is independent of the dataset size: it depends linearly on the number of dictionary elements only, for each iteration.

The Mediator instantiated here can explain both in conventional and contrastive terms, giving answers to "why this class?" and "why this class and not this other one?" types of question. Conventional and contrastive explanations address complementary explanation needs, ranging from plain difficulty to understand why an outcome was advanced by the Oracle, to user claims of misclassifications and corresponding requests for an alternative classification.

The results obtained so far appear to be encouraging, although more experiments are needed and several open issues remain to be tackled. In general, it is not easy to determine objectively whether an explanation method is satisfactory. Some strategies to quantitatively assess the quality of explanations are being proposed in the literature,[29,84–86] but no general solution has been found yet. Similarly, it is a difficult task to objectively assess whether dictionary elements are understandable to humans. However, by sparse dictionary learning methods, one obtains dictionary atoms which may selectively play the role of humanly interpretable elements insofar as they afford a local representation of the data. Indeed, these techniques provide data representations that can be considered to be accessible to human interpretation.[35] Consequently, our approach is limited by the absence of a standard criterion to determine whether an atom is humanly understandable as much as any other explanation methods proposed so far.

Furthermore, a critical aspect is the dataset to be used to construct the required dictionaries. For some special domains and contexts, one may reasonably expect that it is possible to find suitable dictionaries regardless of which dataset was used to train the classifier (the Oracle). But in general, this is not the case. More realistically, one should consider a dataset with a certain degree of overlap with the the dataset used to train the model. Finally, it would be interesting to test our approach on more complex and massive datasets such as Imagenet.[87] As a next step in our inquiry on conventional and contrastive explanations, we are now planning to get and apply the computing power needed to perform experiments on more complex and more significant state-of-art neural networks and datasets.

Acknowledgments

The research presented in this paper was partially supported by the national project Perception, Performativity and Cognitive Sciences (PRIN Bando 2015, cod. 2015TM24JS_009).

References

1. J. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, Striving for simplicity: The all convolutional net, in *Proc. It. Conf. Learning Representation* (*Workshop Track*) (San Diego, CA, 2015).
2. F. Vera-Olmos, E. Pardo, H. Melero and N. Malpica, Deepeye: Deep convolutional network for pupil detection in real environments, *Integr. Comput.-Aided Eng.* **26**(1) (2019) 85–95.
3. T. Yang, C. Cappelle, Y. Ruichek and M. El Bagdouri, Multi-object tracking with discriminant correlation filter based deep learning tracker, *Integr. Comput.-Aided Eng.* **26**(3) (2019) 273–284.
4. J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *NAACL-HLT* **1** (2019) 4171–4186.

5. O. Gabriel, R. Pupo and S. Ventura, Performing multi-target regression via a parameter sharing-based deep network, *Int. J. Neural Syst.* **29**(9) (2019) 1950014:1–1950014:22.
6. C. Richter, W. Vega-Brown and N. Roy, Bayesian learning for safe high-speed navigation in unknown environments, *Robotics Research*, eds. A. Bicchi and W. Burgard (Springer, 2018), pp. 325–341.
7. J. F. Torres, A. Galicia, A. Troncoso and F. Martínez-Álvarez, A scalable approach based on deep learning for big data time series forecasting, *Integr. Comput.-Aided Eng.* **25**(4) (2018) 335–348.
8. A. Antoniades, L. Spyrou, D. Martin-Lopez, A. Valentin, G. Alarcon, S. Sanei and C. C. Took, Deep neural architectures for mapping scalp to intracranial EEG, *Int. J. Neural Syst.* **28**(8) (2018) 1850009.
9. C. Hua, H. Wang, H. Wang, S. Lu, C. Liu and S. M. Khalid, A novel method of building functional brain network using deep learning algorithm with application in proficiency detection, *Int. J. Neural Syst.* **29**(1) (2019) 1850015.
10. A. H. Ansari, P. J. Cherian, A. Caicedo, G. Naulaers, M. De Vos and S. Van Huffel, Neonatal seizure detection using deep convolutional neural networks, *Int. J. Neural Syst.* **29**(4) (2019) 1850011.
11. M. Ahmadlou and H. Adeli, Enhanced probabilistic neural network with local decision circles: A robust classifier, *Integr. Comput.-Aided Eng.* **17**(3) (2010) 197–210.
12. M. H. Rafiei and H. Adeli, A new neural dynamic classification algorithm, *IEEE Trans. Neural Netw. Learn. Syst.* **28** (2017) 3074–3083.
13. B. Letham, C. Rudin, T. H. McCormick, D. Madigan *et al.*, Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model, *Ann. Appl. Stat.* **9**(3) (2015) 1350–1371.
14. A. Adadi and M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (xai), *IEEE Access* **6** (2018) 52138–52160.
15. A. Ortiz, J. Munilla, J. M. Gorriz and J. Ramirez, Ensembles of deep learning architectures for the early diagnosis of the alzheimer's disease, *Int. J. Neural Syst.* **26**(7) (2016) 1650025.
16. F. C. Morabito, M. Campolo, N. Mammone, M. Versaci, S. Franceschetti, F. Tagliavini, V. Sofia, D. Fatuzzo, A. Gambardella, A. Labate *et al.*, Deep learning representation from electroencephalography of early-stage creutzfeldt-jakob disease and features for differentiation from rapidly progressive dementia, *Int. J. Neural Syst.* **27**(2) (2017) 1650039.
17. F. J. Martínez-Murcia, J. M. Górriz, J. Ramirez and A. Ortiz, Convolutional neural networks for neuroimaging in parkinson's disease: Is preprocessing needed?, *Int. J. Neural Syst.* **28**(10) (2018) 1850035–1850035.
18. J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. ODonoghue, D. Visentin *et al.*, Clinically applicable deep learning for diagnosis and referral in retinal disease, *Nat. Med.* **24**(9) (2018) 1342.
19. H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama and H. Omata, Road damage detection and classification using deep neural networks with smartphone images, *Comput.-Aid. Civil Infrastruct. Eng.* **33**(12) (2018) 1127–1141.
20. T. Nguyen, A. Kashani, T. Ngo and S. Bordas, Deep neural network with high-order neuron for the prediction of foamed concrete strength, *Comput.-Aided Civil Infrastruct. Eng.* **34**(4) (2019) 316–332.
21. S. Li, X. Zhao and G. Zhou, Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network, *Comput.-Aided Civil Infrastruct. Eng.* **34**(7) (2019) 616–634.
22. K. Maeda, S. Takahashi, T. Ogawa and M. Haseyama, Convolutional sparse coding-based deep random vector functional link network for distress classification of road structures, *Comput.-Aided Civil Infrastruct. Eng.* **24** (2019) 654–676.
23. R.-T. Wu, A. Singla, M. R. Jahanshahi, E. Bertino, B. J. Ko and D. Verma, Pruning deep convolutional neural networks for efficient edge computing in condition assessment of infrastructures, *Comput.-Aided Civil Infrastruct. Eng.* **34** (2019) 774–789.
24. Y. Zhang, Y. Miyamori, S. Mikami and T. Saito, Vibration-based structural state identification by a 1-dimensional convolutional neural network, *Comput.-Aided Civil Infrastruct. Eng.* **34** (2019) 822–839.
25. B. K. Oh, B. Glisic, Y. Kim and H. S. Park, Convolutional neural network-based wind-induced response estimation model for tall buildings, *Comput.-Aided Civil Infrastruct. Eng.* **34** (2019) 843–858.
26. D. Doran, S. Schulz and T. R. Besold, What does explainable AI really mean? A new conceptualization of perspectives, *CoRR* abs/1710.00794.
27. D. Erhan and Y. Bengio, A. Courville and P. Vincent, Visualizing higher-layer features of a deep network, Technical Report University of Montreal **1341**(3) (2009) 1.
28. A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox and J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, *Advances in Neural Information Processing Systems* 29, eds. D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett (Curran Associates, 2016), pp. 3387–3395.
29. G. Montavon, W. Samek and K. Müller, Methods for interpreting and understanding deep neural networks, *Dig. Sign. Process.* **73** (2018) 1–15.
30. M. T. Ribeiro, S. Singh and C. Guestrin, why should i trust you?: Explaining the predictions of any classifier, in *Proc. 22Nd ACM SIGKDD Int.*

Conf. Knowledge Discovery and Data Mining, KDD '16, (ACM, New York, NY, USA, 2016), pp. 1135–1144.
31. K. Simonyan, A. Vedaldi and A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, *2nd Int. Conf. Learning Representations, Workshop Track Proc.* (Banff, Canada, 2014).
32. S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller and W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, *PloS One* **10**(7) (2015) e0130140.
33. A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller and W. Samek, Layer-wise relevance propagation for neural networks with local renormalization layers, *Int. Conf. Artificial Neural Networks* (Springer, Barcelona, Spain, 2016), pp. 63–71.
34. T. Miller, Explanation in artificial intelligence: Insights from the social sciences, *Artif. Intell.* **267** (2019) 1–38.
35. A. Mensch, J. Mairal, B. Thirion and G. Varoquaux, Dictionary learning for massive matrix factorization, in *Proc. 33rd Int. Conf. Machine Learning* (New York, USA, 2016), pp. 1737–1746.
36. D. J. Hilton, Conversational processes and causal explanation, *Psychol. Bull.* **107**(1) (1990) 65.
37. A. Apicella, F. Isgrò, R. Prevete and G. Tamburrini, Contrastive explanations to classification systems using sparse dictionaries, in *Int. Conf. Image Analysis and Processing*, (Springer, Cham, 2019), pp. 207–218.
38. A. Apicella, F. Isgro, R. Prevete, G. Tamburrini and A. Vietri, Sparse dictionaries for the explanation of classification systems, *PIE*, (Rome, Italy, 2019), p. 009.
39. A. Apicella, F. Isgro, R. Prevete, A. Sorrentino and G. Tamburrini, Explaining classification systems using sparse dictionaries, in *Proc. European Symp. Artificial Neural Networks, Computational Intelligence and Machine Learning, Special Session on Societal Issues in Machine Learning: When Learning from Data is Not Enough*, (Bruges, Belgium, 2019).
40. A. Weller, Transparency: Motivations and challenges, arXiv:1708.01870 [cscy].
41. W. Samek and K.-R. Müller, Towards explainable artificial intelligence, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, eds. W. Samek, G. Montavon, A. Vedaldi, L. Hansen and K. Muller (Springer, 2019), pp. 5–22.
42. A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, *Inf. Fusion* **58** (2020) 82–115.
43. D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen and K.-R. MÃžller, How to explain individual classification decisions, *J. Mach. Learn. Res.* **11** (2010) 1803–1831.
44. S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J. Haynes, B. Blankertz and F. Bießmann, On the interpretation of weight vectors of linear models in multivariate neuroimaging, *Neuroimage* **87** (2014) 96–110.
45. R. C. Fong and A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in *Proc. Int. Conf. Computer Vision*, (Venice, Italy, 2017).
46. R. Fong and A. Vedaldi, Explanations for attributing deep neural network predictions, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, eds. W. Samek, G. Montavon, A. Vedaldi, L. Hansen and K. Muller (Springer, 2019), pp. 149–167.
47. G. Bologna, A study on rule extraction from several combined neural networks, *Int. J. Neural Syst.* **11**(3) (2001) 247–255.
48. H. Núñez, C. Angulo and A. Català, Rule extraction from support vector machines, in *Proc. ESANN* (Bruges, Belgium, 2002), pp. 107–112.
49. H. Núñez, C. Angulo and A. Català, Rule-based learning systems for support vector machines, *Neural Process. Lett.* **24**(1) (2006) 1–18.
50. M. Craven and J. W. Shavlik, Extracting tree-structured representations of trained networks, *Advances in Neural Information Processing Systems* (Denver, CO, USA, 1996), pp. 24–30.
51. J. Kauffmann, K.-R. Mller and G. Montavon, Towards explaining anomalies: A deep taylor decomposition of one-class models, arXiv:1805.06230.
52. R. Caccavale and A. Finzi, Learning attentional regulations for structured tasks execution in robotic cognitive control, *Auton. Robots* **43** (2019) 2229–2243.
53. J. Kauffmann, M. Esders, G. Montavon, W. Samek and K.-R. Mller, From clustering to cluster explanations via neural networks, arXiv:org1906.G633.
54. S. J. Oh, B. Schiele and M. Fritz, Towards reverse-engineering black-box neural networks, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, eds. W. Samek, G. Montavon, A. Vedaldi, L. Hansen and K. Muller (Springer, 2019), pp. 121–144.
55. M. Ancona, E. Ceolini, C. Oztireli and M. Gross, Towards better understanding of gradient-based attribution methods for deep neural networks, *6th Int. Conf. Learning Representations* (Vancouver, Canada, 2018).
56. A. Shrikumar, P. Greenside and A. Kundaje, Learning important features through propagating activation differences, in *Proc. 34th Int. Conf. Machine*

Learning-Vol. 70, JMLR. org (Sydney, Australia, 2017), pp. 3145–3153.
57. M. Sundararajan, A. Taly and Q. Yan, Axiomatic attribution for deep networks, in *Proc. 34th Int. Conf. Machine Learning*, Vol. 70 (Sydney, Australia, 2017), pp. 3319–3328.
58. G. Montavon, S. Lapuschkin, A. Binder, W. Samek and K.-R. Müller, Explaining nonlinear classification decisions with deep taylor decomposition, *Pattern Recogn.* **65** (2017) 211–222.
59. L. M. Zintgraf, T. S. Cohen, T. Adel and M. Welling, Visualizing deep neural network decisions: Prediction difference analysis, in *Proc. Int. Conf. Learning Representation*, (Toulon, France, 2017).
60. M. Robnik-Šikonja and I. Kononenko, Explaining classifications for individual instances, *IEEE Trans. Knowl. Data Eng.* **20**(5) (2008) 589–600.
61. L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele and T. Darrell, Generating visual explanations, in *Proc. European Conf. Computer Vision* (Amsterdam, The Netherlands, 2016), pp. 3–19.
62. J. Yosinski, J. Clune, A. Nguyen, T. Fuchs and H. Lipson, Understanding neural networks through deep visualization, in *Proc. ICML Deep Learning Workshop* (Lille, France, 2015).
63. A. Mahendran and A. Vedaldi, Visualizing deep convolutional neural networks using natural preimages, *Int. J. Comput. Vis.* **120**(3) (2016) 233–255.
64. A. Nguyen, J. Yosinski and J. Clune, Understanding neural networks via feature visualization: A survey, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, eds. W. Samek, G. Montavon, A. Vedaldi, L. Hansen and K. Muller (Springer, 2019), pp. 55–76.
65. M. D. Zeiler, G. W. Taylor and R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, *2011 IEEE Int. Conf. Computer Vision (ICCV)* (Barcelona, Spain, 2011), pp. 2018–2025.
66. M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, *European Conf. Computer Cision* (Springer, Zurich, Switzerland, 2014), pp. 818–833.
67. A. Dosovitskiy and T. Brox, Inverting visual representations with convolutional networks, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Las Vegas, USA, 2016), pp. 4829–4837.
68. M. T. Ribeiro, S. Singh and C. Guestrin, Anchors: High-precision model-agnostic explanations, *Thirty-Second AAAI Conf. Artificial Intelligence* (New Orleans, Louisiana, USA, 2018).
69. R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini and F. Giannotti, Local rule-based explanations of black box decision systems, *CoRR*, abs/1805.10820.
70. J. Zhang, S. A. Bargal *et al.*, Top-down neural attention by excitation backprop, *Int. J. Comput. Vis.* **126** (2017) 1084–1102.
71. L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter and L. Kagal, Explaining explanations: An overview of interpretability of machine learning, *2018 IEEE 5th Int. Conf. Data Science and Advanced Analytics* (*DSAA*), IEEE, (Turin, Italy, 2018), pp. 80–89.
72. K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee and T. J. Sejnowski, Dictionary learning algorithms for sparse representation, *Neural Comput.* **15**(2) (2003) 349–396.
73. G. Tessitore and R. Prevete, Designing structured sparse dictionaries for sparse representation modeling, *Computer Recognition Systems* 4, eds. R. Burduk, M. Kurzynski, M. Wozniak and A. Zolnierek (Springer, 2011), pp. 157–166.
74. C. Bao, H. Ji, Y. Quan and Z. Shen, Dictionary learning for sparse coding: Algorithms and convergence analysis, *IEEE Trans. Pattern Analysis and Mach. Intell.* **38**(7) (2015) 1356–1369.
75. J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, Discriminative learned dictionaries for local image analysis, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Anchorage, AK, USA, 2008).
76. S.-C. Hsu, I.-C. Chang and C.-L. Huang, Vehicle verification between two nonoverlapped views using sparse representation, *Pattern Recogn.* **81** (2018) 131–146.
77. D. D. Lee and H. S. Seung, Algorithms for non-negative matrix factorization, in *Proc. 2000 Conf. Advances in Neural Information Processing Systems* (Vancouver, British Columbia, Canada, 2001), pp. 556–562.
78. P. O. Hoyer, Non-negative matrix factorization with sparseness constraints, *J. Mach. Learn. Res.* **5** (2004) 1457–1469.
79. C. M. Bishop *et al.*, *Neural Networks for Pattern Recognition* (Oxford University Press, 1995).
80. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* **86**(11) (1998) 2278–2324.
81. H. Xiao, K. Rasul and R. Vollgraf, Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms, *CoRR*, abs/1708.07747.
82. D. Kingma and J. Ba, Adam: A method for stochastic optimization, *Int. Conf. Learning Representations*, (Banff, Canada, 2014).
83. L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley Series in Probability and Statistics (Wiley, 1990).
84. D. Alvarez-Melis and T. S. Jaakkola, On the robustness of interpretability methods, *CoRR*, abs/1806.08049.

85. S. A. Friedler, C. D. Roy, C. Scheidegger and D. Slack, Assessing the local interpretability of machine learning models, *CoRR*, abs/1902.03501.
86. P. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan and B. Kim, The (un)reliability of saliency methods, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, eds. W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen and K. Müller, Lecture Notes in Computer Science, Vol. 11700 (Springer, 2019), pp. 267–280.
87. J. Deng, W. Dong, R. Socher, L. Li, K. Li and F. Li, Imagenet: A large-scale hierarchical image database, *CVPR* (IEEE Computer Society, 2009), pp. 248–255.