A general approach for Explanations in terms of Middle Level Features

Andrea Apicella, Francesco Isgrò, Roberto Prevete
Laboratory of Augmented Reality for Health Monitoring (ARHeMLab),
Laboratory of Artificial Intelligence, Privacy & Applications (AIPA Lab),
Department of Electrical Engineering and Information Technology,
University of Naples Federico II
Corresponding author: Andrea Apicella, and.api.univ@gmail.com

Abstract

Nowadays, it is growing interest to make Machine Learning (ML) systems more understandable and trusting to general users. Thus, generating explanations for ML system behaviours that are understandable to human beings is a central scientific and technological issue addressed by the rapidly growing research area of eXplainable Artificial Intelligence (XAI). Recently, it is becoming more and more evident that new directions to create better explanations should take into account what a good explanation is to a human user, and consequently, develop XAI solutions able to provide user-centred explanations. This paper suggests taking advantage of developing an XAI general approach that allows producing explanations for an ML system behaviour in terms of different and user-selected input features, i.e., explanations composed of input properties that the human user can select according to his background knowledge and goals. To this end, we propose an XAI general approach which is able: 1) to construct explanations in terms of input features that represent more salient and understandable input properties for a user, which we call here Middle-Level input Features (MLFs), 2) to be applied to different types of MLFs. We experimentally tested our approach on two different datasets and using three different types of MLFs. The results seem encouraging.

1 Introduction

A large part of Machine Learning(ML) techniques – including Support Vector Machines (SVM) and Deep Neural Networks (DNN) – give rise to ML systems, the behaviour of which is often complex to interpret [1]. More precisely, although ML techniques with reasonably interpretable mechanisms and outputs exist, as, for example, decision trees, the most significant part of ML techniques give responses whose relationships with the input are often difficult to understand. In this sense, it is common to consider them as black-box systems. In particular,

as ML systems are frequently being used in more and more domains and, so, by a more varied audience, there is the need for making them understandable and trusting to general users [34, 6]. Hence, generating explanations for ML system behaviours that are understandable to human beings is a central scientific and technological issue addressed by the rapidly growing research area of eXplainable Artificial Intelligence (XAI). Several definitions of interpretability/explainability for ML systems have been discussed in the XAI context [15, 6], and many approaches to the problem of overcoming their opaqueness are now pursued [31, 7, 5]. For example, in [30] a series of techniques for the interpretation of DNN is discussed, and in [27] the authors examine the motivations underlying interest in interpretability, discussing and refining the notion of interpretability in ML systems. In the context of this multifaceted interpretability problem, it is essential to note that in making an explanation understandable for a user is to be taken into consideration what information the user desires to receive [23, 34, 4]. Recently, it is becoming more and more evident that new directions to create better explanations should take into account what a good explanation is for a human user, and consequently to develop XAI solutions able to provide user-centred explanations [23, 34, 26, 2]. By contrast, much of the current XAI methods provide specific ways to build explanations that are based on the researchers' intuition of what constitutes a "good" explanation [26, 29].

Based on these considerations, in this paper, we suggest taking advantage of developing a XAI general approach which allows producing explanations for an ML System behaviour in terms of different and user-selected input features, i.e., explanations composed of input properties which the human user can select according to his background knowledge and goals.

To this end, we note that in the literature, one of the most successful strategies is to provide explanations in terms of "visualisations" [34, 42], and, more specifically, in terms of low-level input features such as relevance or heat maps of the input by model-agnostic or model-specific methods, like sensitivity analysis[37] or Layer-wise Relevance Propagation (LRP) [7] methods. For example, LRP associates a relevance value to each input element (to each pixel in case of images) to explain the ML model answer. The main problem with such methods is that human users are left with a significant interpretive burden. Starting from each low-level feature's relevance, the human user needs to identify the overall input properties perceptually and cognitively salient to him [5]. Thus, an XAI general approach should alleviate this weakness of low-level approaches and overcome their limitations, allowing the possibility to construct explanations in terms of input features that represent more salient and understandable input properties for a user, which we call here Middle-Level input Features (MLFs) (see Figure 1). Although there is recent research line which attempts to give explanations in terms of visual human-friendly concepts [23, 17, 2] (we will discuss them in Section 2), however we notice that the goal to learn data representations that are easily factorised in terms of meaningful features is, in general, pursued in the *representation learning* framework [8], and more recently in the *feature disentanglement learning* context [28]. These meaningful features may represent parts of the input such as nose, ears and paw (similarly to the

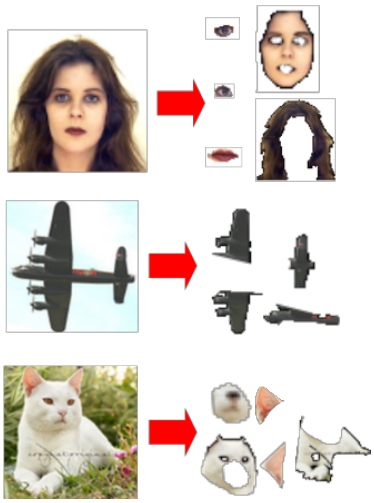


Figure 1: Examples of Middle Level input Features (MLFs). Each MLF represents a part of the input which is perceptually and cognitively salient to a human being, as for example the ears of a cat or the wings of an airplane. These features are intuitively more humanly interpretable respect to low-level features (as for example raw unrelated image pixels), so a decision explanation expressed in terms of middle feature relevance can be easier to understand for the humans being respect to explanations expressed in terms of low level features.

outcome of a clustering algorithm) or more abstract input properties such as shape, viewpoint, thickness, and so on, leading to data representations perceptually and cognitively salient to the human being. We propose to develop an XAI general approach able to give explanations in terms of features which are obtained by standard representation learning methods such as sparse dictionary learning [14], variational auto-encoder [11] and hierarchical image segmentation [16]. Keeping in mind the general potential advantages of the use of this approach, in particular notice that the *hierarchical* organisation of the data in terms of more elementary factors can be a crucial point: the underlying idea is that the data can be described as a hierarchy of increasingly abstract concepts. For example, natural images can be described in terms of the objects they show at various levels of granularity [40, 43, 39].

To the best of our knowledge, in the XAI literature, however, there are relatively few approaches that pursue this line of research. For example, in [33], the authors proposed LIME, a successful method which is based, in case of image classification problems, on explanations expressed as sets of regions, clusters of the image, said superpixels which are obtained by a clustering algorithm. These superpixels can be interpreted as MLFs. In [5] the explanations are formed of elements selected from a dictionary of MLFs, which is obtained by sparse dictionary learning methods. In [18] authors propose to exploit the latent representations learned through an adversarial autoencoder for generating

a synthetic neighbourhood of the image for which an explanation is required. However, these approaches propose specific solutions which cannot be generalised to different types of input properties.

By contrast, in this paper, we investigate the possibility of obtaining explanations using a general approach that can be applied to different types of MLFs, which we call *General MLF Explanations* (GMLF). More precisely, we develop an XAI framework that can be applied whenever a) the input of an ML system can be encoded and decoded based on MLFs, and b) any Explanation method producing a Relevance Map (ERM method) can be applied on both the ML model and the decoder. In this sense, we propose a general framework insofar as it can be applied to several different computational definitions of MLFs and a large class of ML models. In particular, we tested our novel approach using MLFs extracted by three different methods: 1) Image segmentation algorithm; 2) Hierarchical image segmentation algorithm; 3) Variational autoencoder.

The paper is organised as follows: Section 3 describes in detail the proposed approach; in Section 2 we discuss differences and advantages of GMLF with respect similar approaches presented in the literature; experiments and results are discussed in Section 5; the concluding Section summarises the main high-level features of the proposed explanation framework and outlines some future developments.

2 Related works

Recently, a growing number of studies [45, 23, 17, 2] have focused on providing explanations in the form of middle-level or high-level human "concepts" as we are addressing it in this paper. In particular, in [23] the authors introduce Concept Activation Vectors (CAV) as a way of visually representing the neural network' inner states associated with a given class. CAVs should represent human-friendly concepts. The basic ideas can be described as follow: firstly, the authors suppose the availability of an external labelled dataset XC where each label corresponds to a human-friendly concept. Then, given a pre-trained neural network classifier to be explained, say NC, they consider the functional mapping f_l from the input to the l-layer of NC. Based on f_l, for each class c of the dataset XC, they build a linear classifier composed of f_l followed by a linear classifier to distinguish the element of XC belonging to the class c from randomly chosen images. The normal to the learned hyperplane is considered the CAV for the user-defined concept corresponding to the class c. Finally, given all the input belonging to a class K of the pre-trained classifier NC, the authors define a way to quantify how much a concept c, expressed by a CAV, influences the behaviour of the classifier, using directional derivatives to computes NC 's conceptual sensitivity across entire class K of inputs.

Building upon the paper discussed above, in [2] the authors provide explanations in terms of *fault-lines*[22]. Fault-lines should represent "high-level semantic aspects of reality that humans zoom in on when imagining an alternative to it". Each fault-line is represented by a minimal set of semantic *xconcepts*

that need to be added to or deleted from the classifier's input to alter the class that the classifier outputs. Xconcepts are built following the method proposed in [23]. In a nutshell, given a pre-trained convolutional neural network CN whose behaviour is to be explained, xconcepts are defined in terms of super-pixels (images or parts of images) related to the feature maps of the l-th CN's convolutional layer, usually the last convolutional layer before the full-connected layer. In particular, these super-pixels are collected when the input representations at the convolution layer l are used to discriminate between a target class c and an alternate class c_{alt}, and they are computed based on the Grad-CAM algorithm [36]. In this way, one obtains xconcepts in terms of images related to the class c and able to distinguish it from the class c_{alt}. Thus, when the classifier CN responds that an input x belongs to a class c, the authors provide an explanation in terms of xconcepts which should represent semantic aspects of why x belongs c instead of an alternate class c_{alt}.

In [17] the authors propose a method to provide explanations related to an entire class of a trained neural classifier. The method is based on the CAVs introduced in [23] and above described. However, in this case, the CAVs are automatically extracted without the need an external labelled dataset expressing human-friendly concepts.

Now, we discuss some key aspects and differences to our proposal. Many of these approaches focus on *global* explanations, i.e., explanations related to en entire class of the trained neural network classifier (see [17, 23]). Instead, in our approach, we are looking for *local* explanations, i.e., explanations for the response of the ML model concerning the single input. However, some authors, see for example [23], provide methods to obtain *local* explanations, but in this case, the explanations are expressed in terms of high-level visual concepts which do not necessarily belong to the input. Thus, again human users are left with a significant interpretive load: starting from external high-level visual concepts, the human user needs to identify the input properties perceptually and cognitively related to these concepts. On the contrary, the input (MLFs) high-level properties are expressed, in our approach, in terms of elements of the input itself.

Another critical point is that high-level or middle-level user-friendly concepts are computed on the basis of the neural network classifier to be explained. In this way, an unsafe short-circuit can be created in which the visual concepts used to explain the classifier are closely related to the classifier itself. This fact could lead to the creation of false human-friendly visual concepts if the classifier is not reliable. By contrast, in our approach, MLFs are extracted independently from the classifier.

A crucial aspect that distinguishes our proposal from the above-discussed research line is grounded on the fact that we propose an XAI *general* approach. Our methodology only needs that MLFs can be obtained using methods framed into data representation research, and, in particular, any auto-encoder architecture for which an explanation method producing a relevance map can be applied on the decoder (see Section 3.1).

To summarise, our GMLF approach, although shares with the above describe

research works the idea to obtain explanations based on middle-level or high-level human-friendly concepts, it has the following distinctive properties with respect them:

1. It is a XAI general framework where middle-level or high-level input properties can be built on the basis of standard methods of data representation learning.
2. It outputs local explanations.
3. The middle-level or high-level input proprieties are computed independently from the ML classifier to be explained.

Regarding points 2) and 3) we notice that a XAI method that has significant similarity with our approach is LIME [33] or its variants (see, for example, [44]). LIME, especially in the context of images, is one of the predominant XAI methods discussed in the literature [13, 44]. It can provide *local* explanations in terms of superpixels which are regions or parts of the input that the classifier receives, as we have already discussed in Section 1. These superpixels can be interpreted as middle-level input properties, which can be more understandable for a human user than low-level features such as pixels. In this sense, we view a similarity in the output between our approach GMLF and LIME. The explanations built by LIME can be considered comparable with our proposed approach but different in the construction process. While LIME builds explanation relying on a proxy model different from the model to explain, the proposed approach relies only on the model to explain, without needing any other model that approximates the original one. To highlight the difference between the produced explanations, in section 4 a comparison between LIME and GMLF outputs is made. For this reason, we compared GMLF just with LIME in the experimental phase (see Section 4). However, unlike GMLF, LIME can use just one type of middle-level input property.

3 Approach

Our approach stems from the following observations. The development of data representations from raw low-level data usually aims to obtain distinctive explanatory features of the data, which are more conducive to subsequent data analysis and interpretation. This critical step has been tackled for long using specific methods which were developed exploiting expert domain knowledge. However, this type of approach can lead to unsuccessful results and requires a lot of heuristic experience and complex manual design [25]. This aspect is similar to what commonly occurs in many XAI approaches, where the explanatory methods are based on the researchers' intuition of what constitutes a "good" explanation. By contrast, representation learning successfully investigates ways to obtain middle/high-level abstract feature representations by automatic machine learning approaches. In particular, a large part of these approaches is based on Auto-Encoder (AE) architectures [9, 25]. AEs correspond to neural

networks composed of at least one hidden layer and logically subdivided into two components, an *encoder* and a *decoder*. From a functional point of view, an AE can be seen as the composition of two functions E and D: E is an encoding function (the encoder) which maps the input space onto a feature space (or latent encoding space), D is a decoding function (the decoder) which inversely maps the feature space on the input space. A meaningful aspect is that by AEs, one can obtain data representations in terms of latent encodings $\mathbf{h}$, where each h_i may represent a MLF ξ_i of the input , such as parts of the input (for example, nose, ears and paw) or more abstract features which can be more salient and understandable input properties for a user. See for example variational AE [24, 32] or image segmentation [10, 41] (see Figure 1). Furthermore, different AEs can extract different data representations which are not mutually exclusive.

Based on the previous considerations, we want to build upon the idea that the elements composing an explanation can be determined by an AE which extracts relevant input features for a human being, i.e., MLFs, and that one might change the type of MLFs changing the type of auto-encoder or obtain multiple and different explanations based on different MLFs.

3.1 General description

Given an ML classification model M which receives an input $\mathbf{x} \in R^d$ and outputs $\mathbf{y} \in R^c$, our approach can be divided into two consecutive steps.

In the first step, we build an auto-encoder $AE \equiv (E, D)$ such that each input $\mathbf{x}$ can be encoded by E in a latent encoding $\mathbf{h} \in R^m$ and decoded by D. As discussed above, to each value h_i is associated a MLF ξ_j , thus each input x is decomposed in a set of m MLFs $\xi = \{\xi_i\}_{i=1}^m$, where to each ξ_i is associated the value h_i. Different choices of the auto-encoder can lead to MLFs $\xi_\mathbf{i}$ of different nature, so to highlight this dependence we re-formalise this first step as follows: we build an encoder $E_\xi : \mathbf{x} \in R^d \rightarrow \mathbf{h} \in R^m$ and a decoder $D_\xi : \mathbf{h} \in R^m \rightarrow \mathbf{x} \in R^d$, where $\mathbf{h}$ encodes $\mathbf{x}$ in terms of the MLFs ξ.

In the second step of our approach, we use an ERM method (an explanation method producing a relevance map of the input) on both M and D_ξ, i.e., we apply it on the model M and then use the obtained relevance values to apply the ERM method on D_ξ getting a relevance value for each middle-level feature. In other words, we stack D_ξ on the top of M thus obtaining a new model DM_ξ which receives as input an encoding $\mathbf{u}$ and outputs $\mathbf{y}$, and uses an ERM method on DM_ξ from $\mathbf{y}$ to $\mathbf{u}$. In Figure 2 we give a graphic description of our approach GMLF, and in algorithm 1) it is described in more details considering a generic auto-encoder, while in algorithms 3 and 4 our approach (GMLF) is described in case of specific autoencoders (see Section 3.2 and 3.3). Thus, we search for a relevance vector $\mathbf{u} \in R^m$ which informs the user how much each MLF of ξ has contributed to the ML model answer $\mathbf{y}$. Note that, GMLF can be generalised to any decoder D_ξ to which a ERM method applies on. In this way, one can build different explanations for a M's response in terms of different MLFs ξ.

In the remainder of this section, we will describe three alternative ways (segmentation, hierarchical segmentation and VAE) to obtain a decoder such that

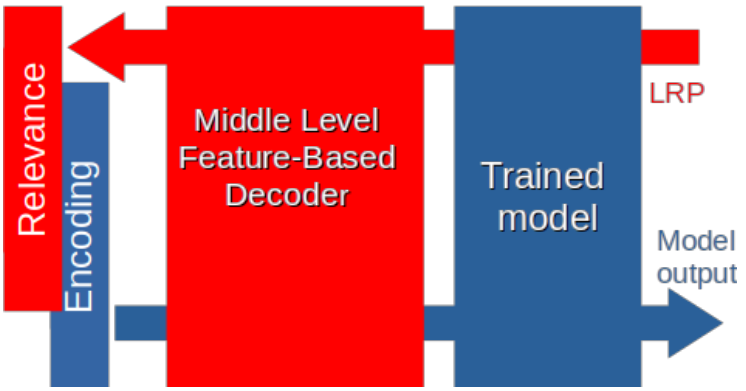


Figure 2: A general scheme of the proposed explanation framework. Given a middle-level feature encoder and the respective decoder, this last one is stacked on the top of the model to inspect. Next, the encoding of the input is fed to the decoder-model system. A backward relevance propagation algorithm is then applied.

Algorithm 1: Proposed method GMLF

Input: data point $\mathbf{x}$, trained model M, an ERP method RP
Output: Feature Relevances U

$\mathbf{y} \leftarrow M(\mathbf{x})$;
build an autoencoder $AE \equiv (E_\xi, D_\xi)$;
$\mathbf{h} \leftarrow E_\xi(\mathbf{x})$;
define $R : \mathbf{h} \mapsto \mathbf{x} - D_\xi(\mathbf{h})$;
define $DM_\xi : \mathbf{h} \mapsto M(D_\xi(\mathbf{h}) + R(\mathbf{h}))$;
$U \leftarrow RP(DM_\xi, \mathbf{h}, \mathbf{y})$;
return U;

a ERM method can be applied to, and so three ways of applying our approach GMLF. We experimentally tested our framework using all the methods.

3.2 MLFs from image segmentation with and without a hierarchical organization

Here we describe the implementation of the GMLF approach to the case of an autoencoder built of the basis of hierarchical segmentation. The approach is depicted in Figure 3, while we give an algorithmic formalisation in algorithms 2 and 3.

Given an image $\mathbf{x} \in R^d$, a segmentation algorithm returns a partition of $\mathbf{x}$ composed of m regions $\{q_i\}_{i=1}^m$. Some of the existing segmentation algorithms can be considered *hierarchical segmentation algorithms*, i.e., they return partitions hierarchically organised with increasingly finer levels of details.

More precisely, following [20], we consider a segmentation algorithm to be

hierarchical if it ensures both the causality principle of multi-scale analysis [19] (that is, if a contour is present at a given scale, this contour has to be present at any finer scale) and the location principle (that is, even when the number of regions decreases, contours are stable). These two principles ensure that the segmentation obtained at a coarser detail level can be obtained by merging regions obtained at finer segmentation levels.

In general, given an image, a possible set of MLFs can be the result of a segmentation algorithm. Given an image $\mathbf{x} \in R^d$, and a partition of $\mathbf{x}$ consisting of m regions $\{q_i\}_{i=1}^m$, each image's region q_i can be represented by a vector $\mathbf{v}_i \in R^d$ defined as follows: $v_{ij} = 0$ if $x_j \notin q_i$, otherwise $v_{ij} = x_j$, and $\sum_{i=1}^m \mathbf{v_i} = \mathbf{x}$. Henceforth, for simplicity and without loss of generality, we will use $\mathbf{v}_i$ instead of q_i since they represent the same entities. Consequently, $\mathbf{x}$ can be expressed as linear combination of the $\mathbf{v}_i$ with all the coefficients equal to 1, which represent the encoding of the image $\mathbf{x}$ on the basis of the m regions. More in general, given a set of K different segmentations $\{S_1, S_2, \ldots, S_K\}$ of the same image sorted from the coarser to the finer detail level, it follows that, if the segmentations have a hierarchical relation, each coarser segmentation can be expressed in terms of the finer ones.

More in detail, each region $\mathbf{v}_i^k$ of S_k can be expressed as a linear combination $\sum_j \alpha_j \mathbf{v}_j^{k+1}$ where α_j is 1 if all the pixels in $\mathbf{v}_j^{k+1}$ belong to $\mathbf{v}_i^k$, 0 otherwise. We can apply the same reasoning going from S_K to the image $\mathbf{x}$ considering it as a trivial partition S_{K+1} where each region represents a single image pixel, i.e., $S_{K+1} = \{\mathbf{v}_1^{K+1}, \mathbf{v}_2^{K+1}, \ldots, \mathbf{v}_d^{K+1}\}$, with $v_{ij}^{K+1} = x_j$ if $i = j$, otherwise $v_{ij}^{K+1} = 0$.

It is straightforward to construct a feed-forward full connected neural network of $K+1$ layers representing an image $\mathbf{x}$ in terms of a set of K hierarchically organised segmentations $\{S_k\}_{k=1}^K$ as follows (see Figure 3): the k-th network layer has $|S_k|$ inputs and $|S_{k+1}|$ outputs, the identity as activation functions, biases equal to 0 and each weights w_{ij}^k equal to 1 if the $\mathbf{v}_j^{k+1}$ region belongs to the $\mathbf{v}_i^k$ region, 0 otherwise. The last layer $K+1$ has d outputs and weights equal to $(\mathbf{v}_p^{K+1})_{p=1}^d$. The resulting network can be viewed as a decoder that, fed with the $\mathbf{1}$ vector, outputs the image $\mathbf{x}$.

Note that if one considers $K = 1$, it is possible to use the same approach in order to obtain an $\mathbf{x}$'s segmentation without a hierarchical organisation. In this case the corresponding decoder is a network composed of just one layer.

3.3 MLF from Variational Autoencoders

The concept of "entangled features" is strictly related to the concept of "interpretability". As stated in [21], a disentangled data representation is most likely more interpretable than a classical entangled data representation. This fact is due to the generative factors representation into separate latent variables representing single features of the data (for example, the size or the colour of the represented object in an image).

Using Variational Auto Encoders (VAE) is one of the most affirmed neural network-based methods to generate disentangled encodings. In general, a VAE

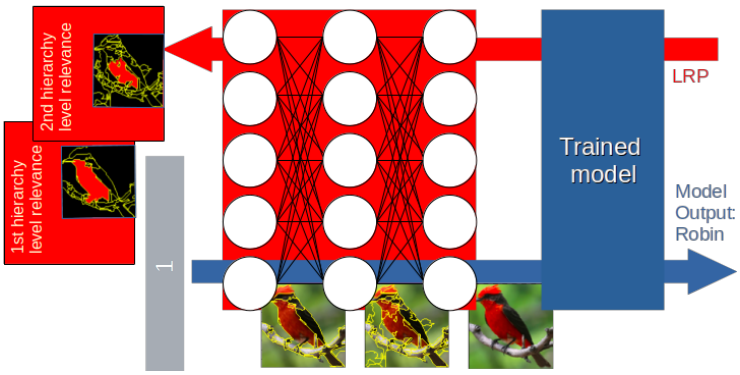


Figure 3: A segmentation-based GMLF framework. The middle-level features decoder is built as a neural networks having as weights the segments returned by a hierarchical segmentation algorithm (see text for further details). The initial encoding is the "1" vector since all the segments are used to compose di input image. The relevance backward algorithm returns the most relevant segments.

is composed of two parts. First, an encoder generates an entangled encoding of a given data point (in our case, an image). Then a decoder generates an image from an encoding. Once trained with a set of data, the VAE output $\tilde{\mathbf{x}}$ on a given input $\mathbf{x}$ can be obtained as the composition of two functions, an encoding function $E(\cdot)$and a decoding function $D(\cdot)$, implemented as two stacked feed-forward neural networks.

The encoding function generates a data representation $E(\mathbf{x}) = \mathbf{h}$ of an image $\mathbf{x}$, the decoding function generates an approximate version $D(\mathbf{h}) = \tilde{\mathbf{x}}$ of $\mathbf{x}$ given the encoding $\mathbf{h}$, with a residual $\mathbf{r} = \mathbf{x} - \tilde{\mathbf{x}}$. So, it is possible to restore the original image data simply adding the residual to $\tilde{\mathbf{x}}$, that is $\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{r}$. Consequently, we stack the decoder neural networks with a further dense layer $R(\cdot)$ having d neurons with weights set to 0 and biases set to $\mathbf{r}$. The resulting network $R(E(\mathbf{h}))$ generates $\mathbf{x}$ as output, given its latent encoding $\mathbf{h}$.

In Figure 4 it is shown a pictorial description of GMLF approach when the autoencoder is built based on VAE, the algorithmic description is reported in algorithm 4.

4 Experimental assessment

In this section, we describe the chosen experimental setup. The goal is to examine the applicability of our approach for different types of MLFs obtained by different encoders. As stated in Section 3.1, three different types of MLFs are evaluated: flat (non hierarchical) segmentation, hierarchical segmentation and VAE latent coding. For non-hierarchical/hierarchical MLF approaches, the segmentation algorithm proposed in [20] was used to make MLFs, since its segmentation constraints respect the causality and the location principles reported

Algorithm 2: Hierarchical segmantion-based Enconder-Decoder Generator

Input: data point $\mathbf{x}$, hierarchical segmentation procedure seg, hierarchical segmentation parameters $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_K)$

Output: A Decoder D_ξ, an Encoder E_ξ

$\{S_1, S_2, \ldots, S_K\} \leftarrow seg(\mathbf{x}, \lambda)$;
$S_{K+1} \leftarrow \emptyset$;
for $x_j \in \mathbf{x}$ **do**
    let $\mathbf{v}^{K+1} \in \{0\}^d$;
    $v_{jj}^{K+1} \leftarrow x_j$;
    $S_{K+1} \leftarrow S_{K+1} \cup \{\mathbf{v}^{K+1}\}$;
end
for $1 \leq k \leq K$ **do**
    let $W^k \in \{0\}^{|S_k| \times |S_{k+1}|}$;
    let $\mathbf{b}^k \in \{0\}^{|S_{k+1}|}$;
    for $1 \leq i \leq |S_k|$ **do**
        for $1 \leq j \leq |S_{k+1}|$ **do**
            if $\mathbf{v}_j^{k+1}$ ***belongs to*** $\mathbf{v}_i^k$ **then**
                $W_{ij} \leftarrow 1$;
            end
        end
    end
    define $identity : \mathbf{a} \mapsto \mathbf{a}$;
    $D_\xi \leftarrow generateNeuralNetwork($weights $= \{W^k\}_{k=1}^{K+1}$, biases $= \{\mathbf{b}^k\}_{k=1}^{K+1}$, activation function $= identity)$;
    define $E_\xi : \mathbf{x} \mapsto e \in \{1\}^{|S_1|}$;
    return D_ξ, E_ξ;
end

in Section 3.2. However, for the non-hierarchical method, any segmentation algorithm can be used (see for example [3]).

For the Variational Auto-Encoder (VAE) based GMLF approach, we used a β-VAE [21] as MLFs builder, since it results particularly suitable for generating interpretable representations. In all the cases, we used as image classifier a VGG16 [38] network pre-trained on ImageNet. MLF relevances are computed with the LRP algorithm using the $\alpha - \beta$ rule[7].

In Section 5 we show a set of possible explanations of the classifier outputs on image sampled from STL-10 dataset [12] and the Aberdeen data set from University of Stirling (http://pics.psych.stir.ac.uk). The STL10 data-set is composed of images belonging to 10 different classes (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck), and the Aberdeen database is composed of

Algorithm 3: GMLF approach in case of Hierarchical segmentation-based autoencoder

Input: a data point $\mathbf{x} \in R^d$, a $trainedNeuralNet$ returning the class scores given a data point, a hierarchical segmentation procedure seg, hierarchical segmentation parameters $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_K)$, a relevance propagation algorithm RP returning a relevance vector for each network layer given: i) a neural network, ii) an input and iii) its class probabilities, a $generateNeuralNetwork$ function that returns a neural networks with weights, biases and activation function given as parameters

Output: relevances for the first K layers $\{\mathbf{u}_1, \ldots, \mathbf{u}_K\}$

1 $\mathbf{y} \leftarrow M(\mathbf{x})$;
   a) $\mathbf{y} \leftarrow TrainedNeuralNet(\mathbf{x})$;
2 build an autoencoder $AE \equiv (E_\xi, D_\xi)$;
   a) $(E_\xi, D_\xi) \leftarrow buildAE(\mathbf{x}, seg, \lambda)$ ▷ see algorithm 2;
3 $\mathbf{h} \leftarrow E_\xi(\mathbf{x})$;
4 define $R : \mathbf{h} \mapsto \mathbf{x} - D_\xi(\mathbf{h})$:
   a) let $W_{res} \in \{0\}^{d \times d}$;
   b) $\mathbf{r} = \mathbf{x} - D_\xi(\mathbf{x})$;
   c) $\mathbf{b}_{res} \leftarrow \mathbf{r}$;
   d) define $identity : \mathbf{a} \mapsto \mathbf{a}$;
   e) $R \leftarrow generateNeuralNetwork$(weights = $\{W_{res}\}$, biases = $\{\mathbf{b}_{res}\}$, activation function = $identity$);
5 define $DM_\xi : \mathbf{h} \mapsto M(D_\xi(\mathbf{h}) + R(\mathbf{h}))$:
   a) $DM_\xi \leftarrow$ stackTogether(D,R,M);
6 $U \leftarrow RP(DM_\xi, \mathbf{h}, \mathbf{y})$;
7 return $\{\mathbf{u}_1, \ldots \mathbf{u}_K\}$;

images belonging to 2 different classes (Male, Female). Only for the Aberdeen data-set the classifier was fine-tuned using an subset of the whole data-set as training set.

4.1 Flat Segmentation approach

For the flat (non-hierarchical) segmentation approach, images from the STL-10 and the Aberdeen data sets are used to generate the classifier outputs and corresponding explanations. For each test image, a set of segments (or superpixels) S are generated using the image segmentation algorithm proposed [20] considering just one level. Therefore, a one-layer neural network decoder as described in Section 3.2 was constructed using the segmentation S. The resulting decoder is stacked on the top of the VGG16 model and fed with the "1" vector (see figure 3). The relevance of each superpixel/segment was then computed using

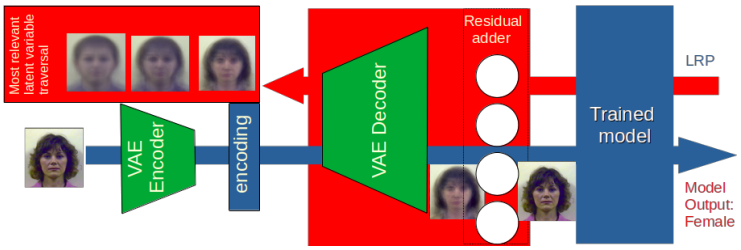


Figure 4: A VAE-based GMLF framework. The middle-level features decoder is built as a neural networks composed of the VAE decoder module followed by a full-connected layer containing the residual of the input (see text for further details). The initial input encoding is given by the VAE encoder module. The relevance backward algorithm returns the most relevant latent variables.

Algorithm 4: GMLF approach in case of VAE autoencoder

Input: a data point $\mathbf{x} \in R^d$, a $trainedNeuralNet$ returning the class scores given a data point, a $getTrainedVAE$ procedure returning a trained VAE, a relevance propagation algorithm RP returning a relevance vector given: i) a neural network, ii) an input and iii) its class probabilities, a $generateNeuralNetwork$ function returning a neural networks with weights, biases and activation function given as parameters

Output: relevances $\mathbf{u}$ of each latent variable

1 $\mathbf{y} \leftarrow M(\mathbf{x})$;
   a) $\mathbf{y} \leftarrow TrainedNeuralNet(\mathbf{x})$;
2 build an autoencoder $AE \equiv (E_\xi, D_\xi)$;
   a) $(E_\xi, D_\xi) \leftarrow getTrainedVAE()$;
3 $\mathbf{h} \leftarrow E_\xi(\mathbf{x})$;
4 define $R : \mathbf{h} \mapsto \mathbf{x} - D_\xi(\mathbf{h})$:
   a) let $W_{res} \in \{0\}^{d \times d}$;
   b) $\mathbf{r} = \mathbf{x} - D_\xi(\mathbf{x})$;
   c) $\mathbf{b}_{res} \leftarrow \mathbf{r}$;
   d) define $identity : \mathbf{a} \mapsto \mathbf{a}$;
   e) $R \leftarrow generateNeuralNetwork(\text{weights} = \{W_{res}\}, \text{biases} = \{\mathbf{b}_{res}\}, \text{activation function} = identity)$;
5 define $DM_\xi : \mathbf{h} \mapsto M(D_\xi(\mathbf{h}) + R(\mathbf{h}))$:
   a) $DM_\xi \leftarrow \text{stackTogether}(D_\xi, R, M)$;
6 $\mathbf{u} \leftarrow RP(DM_\xi, \mathbf{h}, \mathbf{y})$;
7 return $\mathbf{u}$;

---

the LRP algorithm.

4.2 Hierarchical Image Segmentation Approach

As for the non-hierarchical segmentation approach, the segmentation algorithm proposed in [20] was used, but in this case, three hierarchically organised levels were considered. Thus, for each test image, 3 different sets of segments (or superpixels) $\{S_i\}_{i=1}^{3}$ related between them in a hierarchical fashion are generated, going from the coarsest ($i = 1$) to the finest ($i = 3$) segmentation level. Next, a hierarchical decoder is made as described in section 3.2 and stacked on the classifier (see Figure 3). As for the non-hierarchical case, the decoder is then fed with the "1"s vector. Finally, LRP is used to obtain hierarchical explanations as follows: 1) first, at the coarsest level $i = 1$, the most relevant segment $\mathbf{s_i}_{max}$ is selected; 2) then, for each finer level $i > 1$, the segment $\mathbf{s_i}_{max}$ corresponding to the most relevant segment belonging to $\mathbf{s_{i-1}}_{max}$ is chosen.

4.3 Variational auto-encoders

Images from the Aberdeen dataset are used to construct an explanation based on VAE encoding latent variables relevances. The VAE model was trained on an Aberdeen subset using the architecture suggested in [21] for the CelebA dataset. Then, an encoding of 10 latent variables is made using the encoder network for each test image. The resulting encodings were fed to the decoder network stacked on top of the trained VGG 16. Next, the LRP algorithm was applied on the decoder top layer to compute the relevance of each latent variable.

5 Results

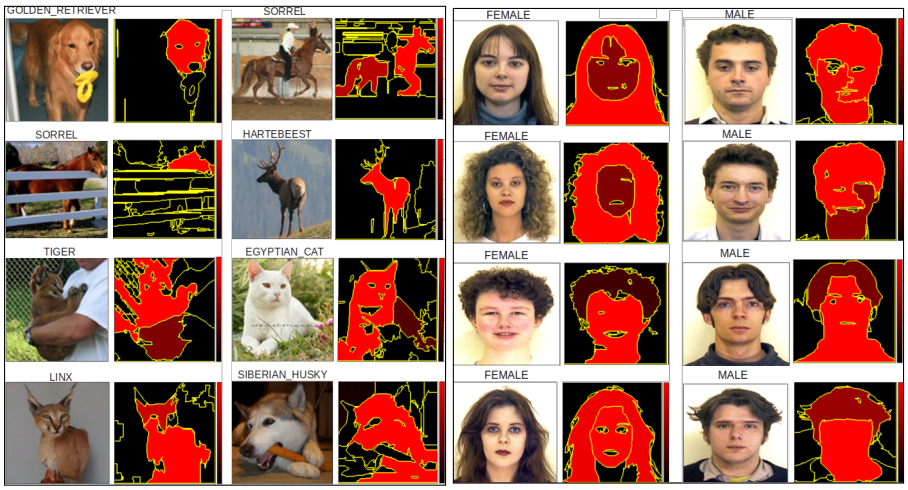
In this section we report the evaluation assessment of the different realisation of the GMLF described in the previous section. For the evaluation we show both qualitative and quantitative results.

5.1 Flat Segmentation

In Figure 5 we show some of the explanations produced for a set of images using the flat (non hierarchical) segmentation-based experimental setup described in Section 3.2. The proposed explanations are reported considering the first two more relevant segments according to the method described in Section 3.2. For each image, the real class and the assigned class are reported. From a qualitative visual inspection, one can observe that the selected segments seem to play a relevant role for distinguishing the classes.

5.2 Hierarchical Image Segmentation

In figures 6 and 7 we show a set of explanations using the hierarchical approach described in Section 3.2 on images of the STL10 and the Aberdeen data sets are. In this case, we exploit the hierarchical segmentation organisation to provide MLF explanations. In particular, for each image, a three layers decoder has



(a) STL10 (b) Aberdeen

Figure 5: Explanations obtained by GMLRF for VGG16 network responses using images from STL10 (a) and Aberdeen datasets (b). In both (a) and (b), for each input (first and the third column) the explanation in terms of most relevant segments are reported (second and fourth column). For better clarity, we report a colormap where only the first two most relevant segments are highlighted.

been used, obtaining three different image segmentations S_1,cS_2 and S_3, from the coarsest to the finest one, which are hierarchically organised (see Section 3.2). For the coarsest segmentation (S_1), the two most relevant segments s_1^1 and s_2^1 are highlighted in the central row. For the image segmentation S_2 the most relevant segment s_1^2 belonging to s_1^1 and the most relevant segment s_2^2 belonging to s_2^1 are highlighted in the upper and the lower row (second column). The same process is made for the image segmentation S_3, where the most relevant segment s_1^3 belonging to s_1^2 and the most relevant segment s_2^3 belonging to s_2^2 are shown in the third column. From a qualitative perspective, one can note that the proposed approach seems to select relevant segments for distinguishing the classes. Furthermore, the hierarchical organisation provides more clear insights about the input image's parts, contributing to the classifier decision.

The usefulness of a hierarchical method can also be seen in cases of wrong classifier responses. See, for example, Figure 8 where a hierarchical segmentation MLR approach was made on two images wrongly classified: 1) a dog wrongly classified as a poodle although it is evidently of a completely different race, and 2) a cat classified as a bow tie. Inspecting the MLR explanations at different hierarchy scales, it can be seen that, in the dog case, the classifier was misled by the wig (which probably led the classifier toward the poodle class), while, in the other case, the cat head position near the neck of the shirt, while the remaining part of the body is hidden, could be responsible for the wrong classification.

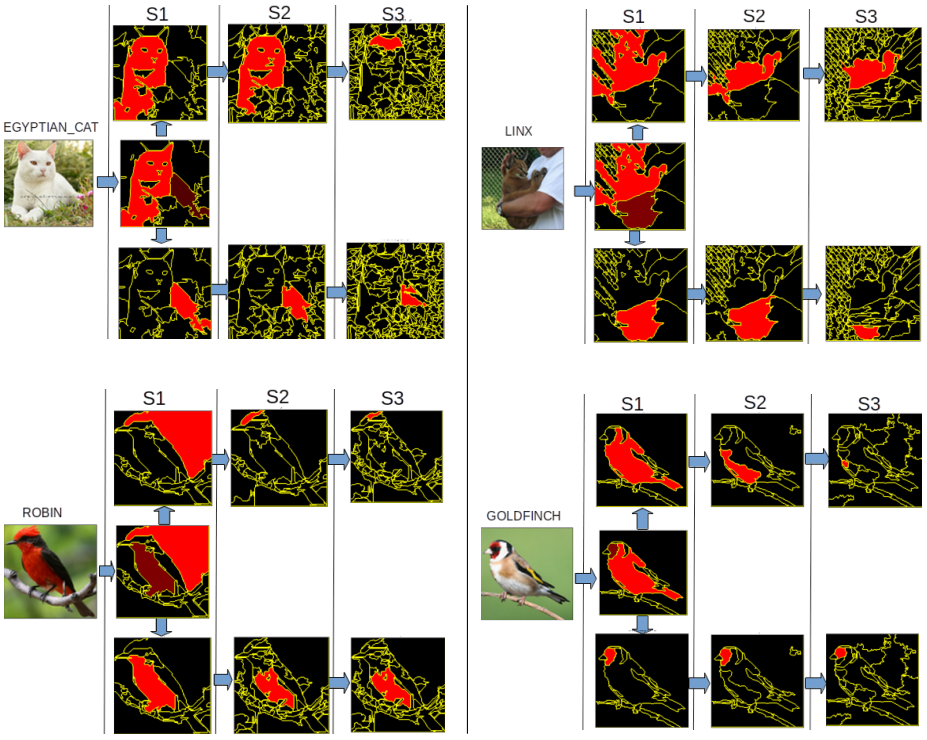


Figure 6: Results obtained from Hierarchical GMLF approach (described in section 3.2) using VGG16 network on the STL10 image dataset. For each image, three different image segmentations S_1,S_2 and S_3 have been computed, from the coarsest to the finest one. For the coarsest segmentation (S_1), the two most relevant segments s_1^1 and s_2^1 are highlighted in the central row. The top and the bottom row show the hierarchy of MLF (see Section 5.2 for details).

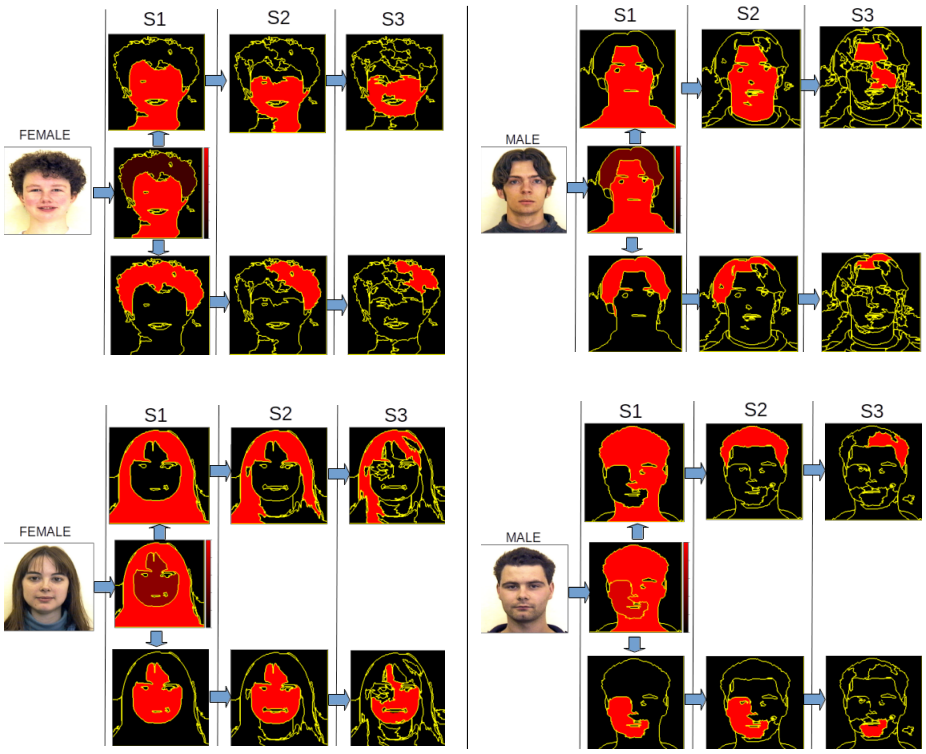
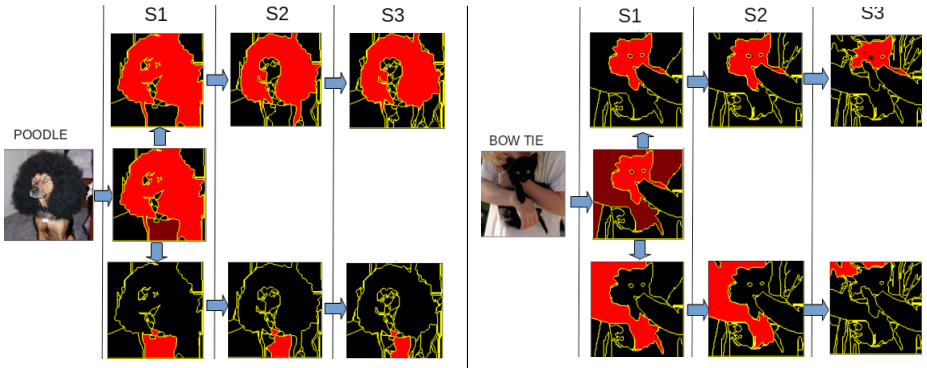


Figure 7: Results obtained from Hierarchical GMLF approach (described in Section 3.2) using VGG16 network on Aberdeen image dataset. For each image, three different image segmentations S_1,S_2 and S_3 have been computed, from the coarsest to the finest one. For the coarsest segmentation (S_1), the two most relevant segments s_1^1 and s_2^1 are highlighted in the central row. The top and the bottom row show the hierarchy of MLF (see Section 5.2 for details).



(a) classified as poodle (b) classified as bow tie

Figure 8: Results obtained from Hierarchical GMLF approach (described in Section 3.2) using VGG16 network on STL10 image dataset in case of wrong model classification. (a) A dog wrongly classified as a poodle, although it is evidently of a completely different race. Inspecting the MLR explanations at different hierarchy scales, it can be seen that the classifier was probably misled by the wig (which probably led the classifier toward the poodle class), (b) A cat wrongly classified as a bow tie. Inspecting the MLR explanations at different hierarchy scales, it can be seen that the shape and the position of the cat head near the neck of the shirt, together with the remaining of its body hidden, could be responsible for the wrong class.

5.3 VAE-based MLF explanations

In Figure 9 a set of results using the VAE-based experimental setup described in Section 4 is shown. For each input, a relevance vector on the latent variable coding is computed. Then, a set of decoded images are generated varying the two most relevant latent variables fixing the other ones to the original encoding values. One can observe that varying the most relevant latent variables it seems that relevant image properties for the classifier decision are modified such as hair length and style.

5.4 Multiple MLF explanations

For the same classifier input-output, we show the possibility to provide multiple and different MLF explanations based on the three types of previously mentioned MLFs. In Figure 10, for each input, three different types of explanations are shown. In the firs row, an explanation based on a flat image segmentation is reported. In the second row, an explanation based on an hierarchical segmentation. In the last row, a VAE-based MLF explanation is showed. Notice that the three types of explanations, although based on different MLFs, seem coherent to each other.

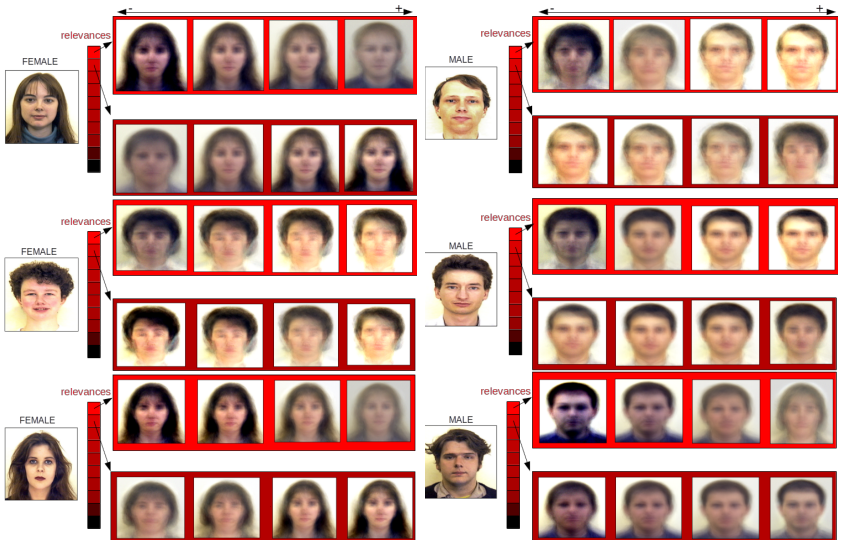


Figure 9: Results obtained from VAE MLF approach (described in Section 3.3) using a VGG16 network on Aberdeen image dataset. For each image, a VAE is constructed. For each input, the resulting relevance vector on the latent variable is computed. Then, decoded images are generated varying the two most relevant latent variables and fixing the other ones to the original values.

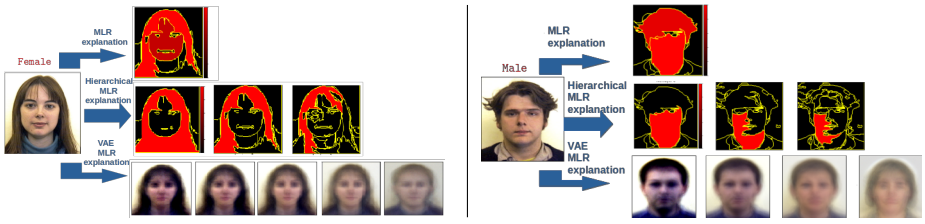


Figure 10: For each input, three different types of explanations obtained by GMLF approach are shown. In the first row, an explanation based on a flat image segmentation is reported. In the second row, an explanation based on an hierarchical segmentation. In the last row, a VAE-based MLF explanation is showed.

5.5 Quantitative evaluation

A quantitative evaluation is performed adopting the MoRF (Most Relevant First) [7, 35] curve analysis. In this work, MoRF curve is computed following the *region flipping* approach, a generalisation of the *pixel-flipping* measure proposed in [7]. In a nutshell, given an image classification, image regions (in our case segments) are iteratively replaced by random noise and fed to the classifier, following the descending order with respect to the relevance values returned by the explanation method. At each step, we compute the difference between the class score p, returned by the model on the original input, and the score p', returned on the perturbed input, generating a curve. We expect that the better the explanation method is, the greater is the difference $p - p'$.

A first evaluation is made comparing our approach using the flat segmentation as MLFs with LIME (Figure 11). The plots show that MLRF outperforms LIME in terms of MoRF curve, suggesting that the flat MLRF approach, on average, gives a better relevance score with respect to the latter. Indeed, a visual inspection of the resulting explanations shows that the distinction between male and female seems to be mainly due to the haircut and the face in the Aberdeen dataset. By contrast, eyes or mouth do not seem to be decisive for the classifier output. This observation seems to be confirmed by the MoRF curve: in the most significant part of the cases, the MoRF curve highlighted that the selected MLFs had greater importance for the classifier outputs.

To evaluate the hierarchical approach with respect to the flat segmentation approach, a most relevant segment analysis is made. More in detail, for each input image, the MoRF curve for the S_1 image segmentation is computed. Next, the MoRF curve is computed on the S_2 and S_3 segmentations considering only the segments belonging to the most relevant one of S_1. This is done to assess if a "finer" explanation can be more decisive than one at a coarser level. In fact, in several cases, removing a finer level feature results in approximately equivalent, in terms of classifier response, to remove a feature of a coarser level. However, a finer middle-level feature can give a piece of more human-interpretable information to the user about the classifier behaviours.

6 Conclusion

A general framework (GMLF) to generate explanations in terms of middle-level features is proposed in this work. With the expression *Middle Level Features* (MLF), (see Section 1, we mean input features that represent more salient and understandable input properties for a user, such as parts of the input (for example, nose, ears and paw, in case of images of humans) or more abstract input properties (for example, shape, viewpoint, thickness and so on). The use of middle-level features is motivated by the need to decrease the human interpretative burden in artificial intelligence explanation systems.

Our approach can be considered a general approach. In fact it can be applied to different types of middle-level features as long as an encoder/decoder system is

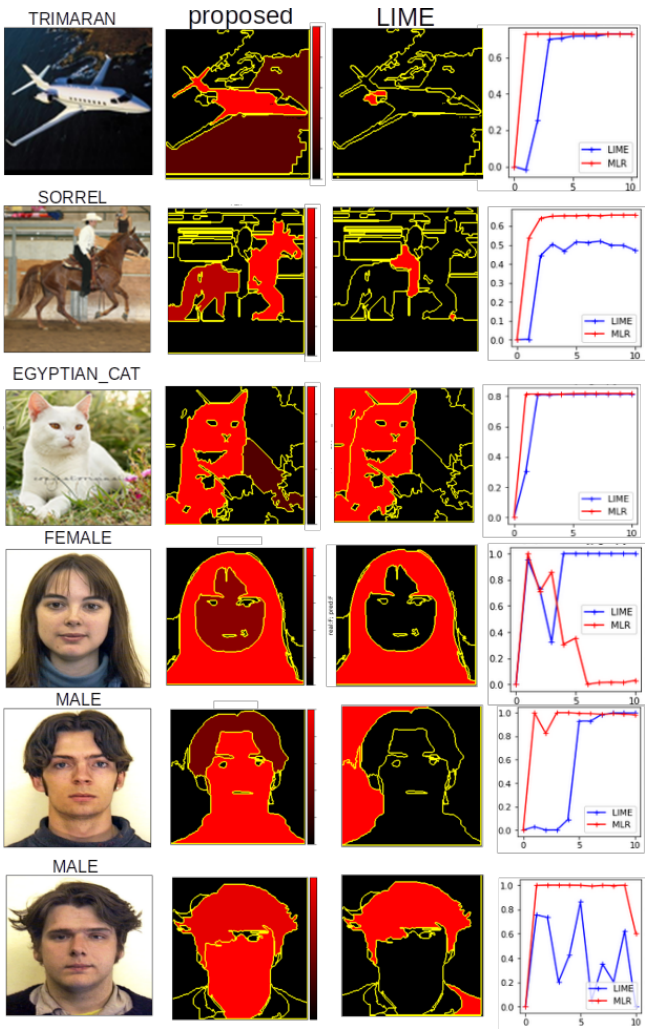
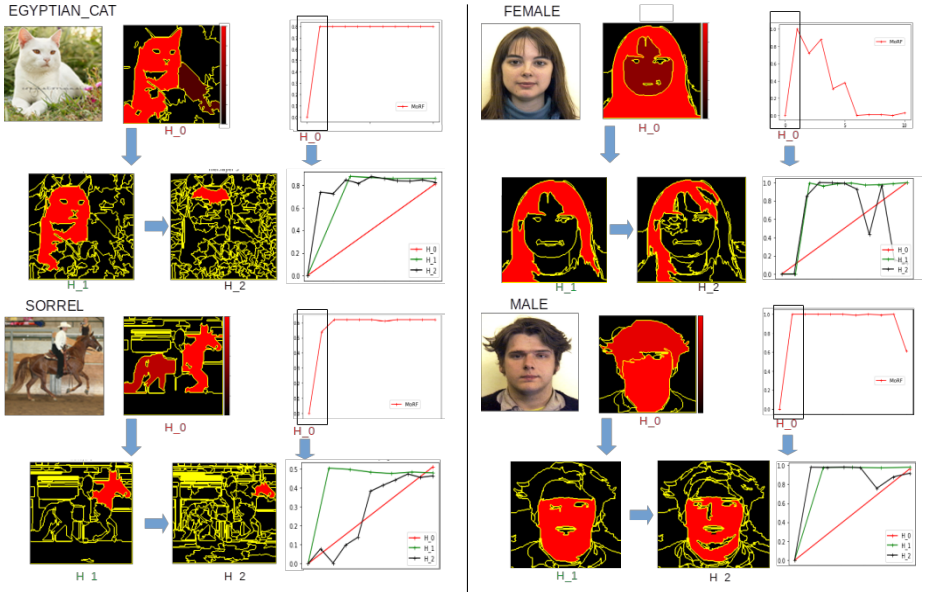


Figure 11: A comparison between the GMLRF flat approach and LIME applied on inputs taken from STL10 and Aberdeen datasets. In the last column a quantitative comparison between the two methods is made through the MoRF curve. See text for further details.



(a) STL10 (b) Aberdeen

Figure 12: A quantitative evaluation of the hierarchical GMLF approach. To evaluate the hierarchical approach respect to the flat GMLF approach, a most relevant segment analysis is made. More in detail, for each input, the MoRF curve for the S_1 image segmentation is shown (upper row). Next, the MoRF curve is computed on the S_2 and S_3 segmentations considering the segments belonging to the most relevant segment of S_1 (lower row) only. It is possible to see that, in several cases, removing a feature of a finer level is equivalent to remove a feature of a coarser level, but it can give a more specific information to the user about the real discriminative feature.

provided (for example image segmentation or latent coding) and an explanation method producing heatmaps can be applied on both the decoder and the ML system whose decision is to be explained (see Section 3.1). Consequently, the proposed approach enables one to obtain different types of explanations in terms of different MLFs for the same pair input/decision of an ML system, allowing to develop XAI solutions able to provide user-centred explanations according to several research directions proposed in literature [34, 26].

We experimentally tested (see Section 4 and 5) our approach using three different types of MLFs: (non hierarchical) segmentation, hierarchical segmentation and VAE latent coding. Two different datasets were used: STL-10 dataset and the Aberdeen dataset from the University of Stirling . We evaluated our results from both a qualitative and a quantitative point of view. The quantitative evaluation was obtained using MoRF curves [35].

The results are encouraging, both under the qualitative point of view, giving easily human interpretable explanations, and the quantitative point of view, giving comparable performances to LIME. Furthermore, we show that a hierarchical approach can provide, in several cases, clear explanations about the reason behind classification behaviours.

References

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

[2] A. Akula, S. Wang, and S.-C. Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2594–2601, 2020.

[3] A. Apicella, S. Giugliano, F. Isgrò, and R. Prevete. A general approach to compute the relevance of middle-level input features. In *Pattern Recognition. ICPR International Workshops and Challenges*, pages 189–203, Cham, 2021. Springer International Publishing.

[4] A. Apicella, F. Isgro, R. Prevete, A. Sorrentino, and G. Tamburrini. Explaining classification systems using sparse dictionaries. In *Proceedings of the European Symposium on 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.

[5] A. Apicella, F. Isgro, R. Prevete, and G. Tamburrini. Middle-level features for the explanation of classification systems by sparse dictionary methods. *International Journal of Neural Systems*, 30(08):2050040, 2020.

[6] 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. *Information Fusion*, 58:82–115, 2020.

[7] 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):e0130140, 2015.

[8] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[9] D. Charte, F. Charte, M. J. del Jesus, and F. Herrera. An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges. *Neurocomputing*, 404:93–107, 2020.

[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[11] R. T. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

[12] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[13] J. Dieber and S. Kirrane. Why model why? assessing the strengths and limitations of lime. *arXiv preprint arXiv:2012.00093*, 2020.

[14] F. Donnarumma, R. Prevete, D. Maisto, S. Fuscone, E. M. Irvine, M. A. van der Meer, C. Kemere, and G. Pezzulo. A framework to identify structured behavioral patterns within rodent spatial trajectories. *Scientific reports*, 11(1):1–20, 2021.

[15] D. Doran, S. Schulz, and T. R. Besold. What does explainable AI really mean? A new conceptualization of perspectives. *CoRR*, abs/1710.00794, 2017.

[16] F. L. Galvão, S. J. F. Guimarães, and A. X. Falcão. Image segmentation using dense and sparse hierarchies of superpixels. *Pattern Recognition*, 108:107532, 2020.

[17] A. Ghorbani, J. Wexler, J. Zou, and B. Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019.

[18] R. Guidotti, A. Monreale, S. Matwin, and D. Pedreschi. Explaining image classifiers generating exemplars and counter-exemplars from latent representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13665–13668, 2020.

[19] L. Guigues, J. P. Cocquerez, and H. Le Men. Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317, 2006.

[20] S. J. F. Guimarães, J. Cousty, Y. Kenmochi, and L. Najman. A hierarchical image segmentation algorithm based on an observation scale. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 116–125. Springer, 2012.

[21] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[22] D. Kahneman and A. Tversky. The simulation heuristic. Technical report, Stanford Univ Ca Dept Of Psychology, 1981.

[23] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[24] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] B. Li and D. Pi. Network representation learning: a systematic literature review. *Neural Computing and Applications*, pages 1–33, 2020.

[26] B. Y. Lim, Q. Yang, A. M. Abdul, and D. Wang. Why these explanations? selecting intelligibility types for explanation goals. In *IUI Workshops*, 2019.

[27] Z. C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[28] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[29] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[30] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[31] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks. *ArXiv e-prints*, Feb. 2016.

[32] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

[33] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM.

[34] M. Ribera and A. Lapedriza. Can we do better explanations? a proposal of user-centered explainable ai. In *IUI Workshops*, 2019.

[35] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[37] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, Workshop Track Proceedings*, Banff, Canada, 2014.

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[39] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3745–3753, 2016.

[40] M. Tschannen, O. Bachem, and M. Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[41] J. Yu, D. Huang, and Z. Wei. Unsupervised image segmentation via stacked denoising auto-encoder and hierarchical patch indexing. *Signal Processing*, 143:346–353, 2018.

[42] Q. Zhang and S. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[43] S. Zhao, J. Song, and S. Ermon. Learning hierarchical features from deep generative models. In *International Conference on Machine Learning*, pages 4091–4099. PMLR, 2017.

[44] X. Zhao, X. Huang, V. Robu, and D. Flynn. Baylime: Bayesian local interpretable model-agnostic explanations. *arXiv preprint arXiv:2012.03058*, 2020.

[45] B. Zhou, Y. Sun, D. Bau, and A. Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.