Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06806-w

S.I. : AI-BASED E-DIAGNOSIS

Classification of Covid-19 chest X-ray images by means of an interpretable evolutionary rule-based approach

Ivanoe De Falco[1] · Giuseppe De Pietro[1] · Giovanna Sannino[1]

Received: 1 August 2021 / Accepted: 26 November 2021

Abstract
In medical practice, all decisions, as for example the diagnosis based on the classification of images, must be made reliably and effectively. The possibility of having automatic tools helping doctors in performing these important decisions is highly welcome. Artificial Intelligence techniques, and in particular Deep Learning methods, have proven very effective on these tasks, with excellent performance in terms of classification accuracy. The problem with such methods is that they represent black boxes, so they do not provide users with an explanation of the reasons for their decisions. Confidence from medical experts in clinical decisions can increase if they receive from Artificial Intelligence tools interpretable output under the form of, e.g., explanations in natural language or visualized information. This way, the system outcome can be critically assessed by them, and they can evaluate the trustworthiness of the results. In this paper, we propose a new general-purpose method that relies on interpretability ideas. The approach is based on two successive steps, the former being a filtering scheme typically used in Content-Based Image Retrieval, whereas the latter is an evolutionary algorithm able to classify and, at the same time, automatically extract explicit knowledge under the form of a set of IF-THEN rules. This approach is tested on a set of chest X-ray images aiming at assessing the presence of COVID-19.

Keywords Covid-19 disease · Chest X-ray images · Classification · Interpretable machine learning · Evolutionary algorithms

1 Introduction

Images are nowadays of paramount importance in medicine for diagnosis. By looking at them, expert clinicians can hypothesize the presence or absence of a specific disease, or even its degree, thus performing human-based classification.

Advances in machine learning have opened this field to automatic classification tools. In the last years, Deep Learning methods [1] relying on Deep Neural Networks (DNNs) have become the de-facto standard for automatic image classification [2]. These methods often allow obtaining very high classification quality measured in terms of indices as, e.g., accuracy and F-score.

The problem with DNNs is that they behave like black boxes, i.e., they do not provide any explanation on the reasons why they assign an item to a given class. Interest is rising, instead, in getting information on this issue. This is true both for experts who make use of DNN-based classification systems and for subjects whose lives are influenced by those decisions. As a few examples, just consider problems related to risk assessment, as credit assignment in finance, recidivism risk prediction in court trials, and diagnosis in medicine as well. In this latter field, when doctors examine images related to a patient, they can motivate their decisions and can explain them to patients. An automatic tool should be able to convince doctors by telling them why that image represents a positive or a negative case and should reassure patients that the decision is correct and will not threaten their lives.

Consequently, there is wide research aiming at providing DNNs and other black-box classifiers with ways to

✉ Ivanoe De Falco
ivanoe.defalco@icar.cnr.it

Giuseppe De Pietro
giuseppe.depietro@icar.cnr.it

Giovanna Sannino
giovanna.sannino@icar.cnr.it

[1] ICAR-CNR, Naples, Italy

Published online: 08 January 2022

explain their decisions. This leads to the concept of *explainable* Artificial Intelligence / Machine Learning [3], and consists of finding a posteriori another model that can inform people with the reasons for the choices, the underlying hypothesis being that the behavior of the inner model of the DNN and of this external one is exactly the same on all the examined items, and on new ones that could be presented in the future.

Of course, the hypothesis above is very strong. It is often the case that, as Cynthia Rudin notes in [4], "Explainable ML methods provide explanations that are not faithful to what the original model computes". Unfortunately, the current ways to add explainability to black boxes are far from being satisfactory. As Rudin writes, "explanations often do not make sense, or do not provide enough detail to understand what the black box is doing."

Unlike explainability, an alternative approach aims at creating classification tools that are *interpretable*, meaning with this that they directly build an explicit model so that they can provide users with explicit knowledge about the problem and with the reasons for classifying items the way they do. This knowledge may be represented as, e.g., decision trees or sets of rules. A common criticism about interpretable models is that they perform worse than explainable ones. Yet, in [4] it is argued that "It is a myth that there is necessarily a trade-off between accuracy and interpretability".

In this paper, we follow this latter approach and apply it to the medical field, with specific reference to the detection of Covid-19 [5]. This is the disease caused by the new SARS-CoV-2 coronavirus. Since its first official announcement on December 31, 2019, this disease has been defined as a pandemic and has caused more than 197 million cases worldwide, and more than 4.2 million people died. Symptoms of Covid-19 are highly variable, the most common being fever, dry cough, and fatigue, followed by, among many, loss of taste or smell, nasal congestion, nausea, vomiting, diarrhea, conjunctivitis. Symptoms of severe Covid-19 disease are shortness of breath, persistent pain or pressure in the chest, high temperature (above 38°C).

The standard examination to diagnose Covid-19 is the Reverse Transcription Polymerase Chain Reaction (RT-PCR), which shows problems related to low accuracy, delay, and low sensitivity [6]. Given its problems, other examinations may help. One of the routine ways to help objectively diagnose the presence of Covid-19 in a subject consists of letting them undergo a chest X-ray radiography (CXR) examination [7]. This has the advantage of being performed easily, also by means of portable X-ray machines that can provide faster, and accurate Covid-19 diagnosis. CXRs, when coupled with AI, can be very useful in the detection of Covid-19 [7]. The outcome of a CXR examination is a set of gray-scale images, through which experts can tell whether or not the subject suffers from Covid-19.

As far as we know, never have Evolutionary Algorithms (EAs) been used to perform the classification task starting from a data set of images, *a fortiori* related to Covid-19. This is confirmed by a wide and recent literature survey conducted in 2020 by Nakane et al. [8]. In it, they report that surveys on the application of EAs and swarm algorithms (SAs) to the field of computer vision have not been updated during the last decade, so their paper is the most reliable source on this. Importantly, they report no EAs being used to classify images of any type. The only way in which EAs have been utilized for image classification consists of their use to evolve good DNN structures and hyper-parameter values [9, 10], yet those structures remain black boxes.

Therefore, in this paper, we propose a new general-purpose methodology to classify images that is based on two steps. The first step performs pre-processing, namely, it filters each image and transforms it into a set of real values, so that the image data set is transformed into a numerical one. In the second step, an evolutionary-based interpretable classifier acts on this numerical data set by performing classification, and at the same time provides users with explicit knowledge. This latter has the form of a set of IF-THEN rules each of which contains AND-connected literals on the data set variables.

This paper is a feasibility study for the proposed approach. To test if it may be useful in the medical domain, and to which degree of accuracy, we make use here of a freely downloadable data set containing X-ray images of subjects' chests. The task is to discriminate people with Covid-19 from healthy ones.

2 Related works

The use of Artificial Intelligence and Machine Learning techniques has proved extremely useful in managing images, as these methodologies have turned out very effective in facing problems as image segmentation, feature detection and selection, image matching, visual tracking, face recognition, human action recognition, and so on. For a recent survey on this with reference to Evolutionary Algorithms, the reader is referred to [8].

As regards the task of image classification, the state of the art is currently largely represented by Deep Learning structures, with specific reference to Convolutional Neural Networks (CNNs) [11]. Their use started in the last eighties of the past century, when, in 1989, LeCun et al. proposed the first multilayered CNN named ConvNet [11], and, after a first wave of interest, their use met a stagnation phase

mainly due to the high computational time, in some cases even some weeks, needed to obtain very small improvements in performance. This was mainly due to the lack of parallel processing techniques and limited hardware resources necessary to train such networks. This problem was overcome in around 2010, and in about the same period some important advancements were made in the activation function, with ReLu or Tanh replacing the Sigmoid [12]. More recently, improvements have been made in the use of parameter optimization strategies, and in the design of new architectural ideas, as proposed in, e.g., [13] (2018), [14] (2018), and [15] (2019).

Basically, the current idea consists of the observation that hyper-parameters as filter dimensions, stride, padding, and so on are difficult to determine for each layer, as this would mean to optimize hundreds, if not thousands, of parameters. Consequently, the idea is to start with a CNN block with fixed topology and to repeat this multiple times. All this research led to significant improvements in CNN performance taking place in the period 2015–2019.

As of today, many different structures exist and are typically used "as they are". Just to mention a few, we can recall here LeNet [16] (1995), AlexNet [17] (2012), GoogleNet [18] (2015), ResNet [19] (2016), and DenseNet [20] (2017),

It is worth pointing out that, apart from CNNs, other Machine Learning algorithms have been used and are still being used on their own to face image classification. From among them, we can recall here at least K-Nearest Neighbour [21], the use of which together with texture features has turned out useful in classifying medical images containing either normal or abnormal tissues [22] (2019). Also used for image classification are methodologies as Support Vector Machines [23], Decision Trees [24], shallow Artificial Neural Networks [25]. These methodologies exhibit good performance when small- or medium-sized data sets are considered, yet they are typically outperformed by Deep Learning algorithms when large or very large data sets, possibly with high numbers of classes, are considered.

However powerful for image classification, CNNs are not immune from drawbacks. Firstly, CNNs are well suited for large data sets allowing good training, whereas in several cases the size is small; this holds frequently true for Covid-19-related data sets. A possible solution to this is represented by transfer learning, in which a CNN is previously trained on a large image data set and is then applied to face the small data set available. Secondly, the execution of CNNs needs high amounts of computational resources in terms of both memory and storage. Thirdly, their working mechanism implies that they build a black-box model based on the implicit extraction of features, yet these cannot be checked and approved by humans. It may happen that these extracted features may mislead classification, thus leading to bad performance.

Consequently, feature selection techniques can be used in conjunction with CNNs. In particular, meta-heuristic techniques are very helpful in this task. For feature selection in images, we can recall here at least the use of a genetic algorithm (GA) on a data set of lung and breast nodules [26] (2011), a Flower Pollination Algorithm for lung cancer detection [27] (2020), a Simulated Annealing scheme hybridized with GA for the classification of brain tumors starting from MR images [28] (2019), a fuzzy particle swarm optimization (PSO) scheme for CT imagery related to emphysema [29] (2019), a Bat Algorithm for lung X-ray images [30] (2019), a hybrid algorithm consisting of PSO and fuzzy C-means for the segmentation of MR images [31] (2020), and an Artificial Bee Colony applied to Parkinson's disease [32] (2020). The years of publication of these papers show that the problem of automatically selecting features is a currently open one in image classification.

As concerns Evolutionary Algorithms and Swarm Intelligence algorithms, it should be remarked here that they are being applied as well for image classification with respect to some specific tasks. As a first task, many papers exist in which these algorithms are used for the automatic design of CNN structures, as for example in [33] (2017), [9] (2018), and [10] (2020). As a second task, other papers show the use of EAs to perform feature selection, as, e.g., [34] (2013), [35] (2015), and [36] (2016).

To the best of our knowledge, instead, no paper describes the use of an EA to directly classify images, which, we believe, is the big novelty of our paper.

3 The method

Our method consists of the sequential application of two steps. Figure 1 shows that the image data set is firstly given as input to a filter that outputs a numerical data set. Then, this latter is input to a classifier that performs explicit knowledge extraction. These steps are detailed in the following subsections.

3.1 Image filter

The image filter we have considered was originally proposed in [37] by Mingjing Li. It was introduced in the area of Context-Based Image Retrieval and allows encoding an image as an array of 64 attributes each represented by a real number. Actually, these attributes can be grouped into three sets, each of which takes into account different features of the image.

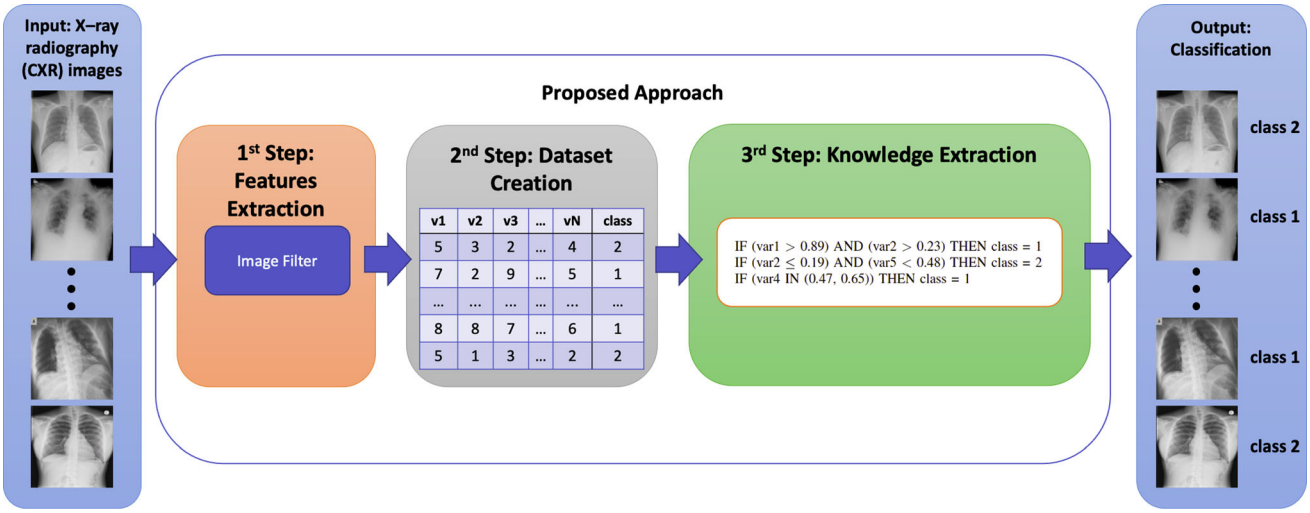

Fig. 1 The method. Input: X-ray radiography (CXR) images. 1st step: the feature extraction through the use of a filter detailed in 3.1 - each image is processed in order to extract 64 features that will constitute an item of the data set. 2nd step: the data set creation - to each item of 64 features it is associated the class (1: *Covid-19* or 2: *Normal*). 3rd step: the DEREx classifier, which also provides the explicit knowledge extraction under the form of IF-THEN rules. Output: the classification - each X-ray image is classified

Firstly, six features make reference to the first two-color moments of the image in the RGB color space. Their values are normalized based on the idea of histogram normalization so that their sum equals 1.0.

Secondly, 14 attributes are related to the texture moment of the image. In this case, the feature extraction is confined to the grayscale representation of an image only. Therefore, rather than to the color, these attributes are somehow related to the structural information of the image. In fact, for each interior pixel of the image, seven different attributes are calculated, which can be seen as representing detected edge strengths at that pixel. Then, the mean and variation of each attribute are calculated separately over the whole set of interior pixels of the image. These values too are normalized in a way that the sum of their values is equal to 1.0.

Thirdly, 44 attributes represent the color correlogram in HSV color space. This incorporates the spatial correlation of image pixels. To obtain them, the HSV color space is quantized into 44 bins, and the auto correlogram is only considered between a pixel and its 8 neighboring ones. In HSV color space, the area deemed corresponding to black color is quantized into 1 bin, and that corresponding to gray levels, white color included, is quantized into 8 bins. This is important whenever the images are not in color but in grayscale, because, in this case, only these nine bins will correspond to non-zero attribute values. In this case too, the values are normalized so that their sum equals 1.0.

A very important consequence of the above is that, if the filter is applied to grayscale images, not all the 64 features will be meaningful. In fact, the only significant variations are, obviously, those related to black, white, and gray levels, which are represented by nine attributes as a whole. This reduces the total number of filter features from 64 to 29.

Readers interested in further details on what these attributes represent, or in why they were chosen rather than others, or in why exactly that number of features was considered, can make reference to the seminal paper [37] by Li.

The application of this filter allows us to pass from the field of images to that of numbers, thus transforming a data set of images into a numerical one. This latter can be easily deal with by EAs in general, and by the DEREx algorithm in particular.

3.2 DEREx

The classifier we use is the Differential-Evolution-based Rule Extractor (DEREx) [38] developed by us. Its core is an EA, specifically one relying on Differential Evolution (DE) [39, 40]. EAs are an optimization methodology relying on mimicking in a computer the evolution of a population of individuals that takes place in nature. Given a problem needing optimization, an EA iteratively updates a set of a number of *Pop_Size* solutions over a number of *Max_Gens* iterations. An objective function, called the *fitness* function, drives this optimization process. Details on how this takes place in DE can be found in [39, 40]. Other DE parameters include the crossover ratio *CR* and the mutation factor *FV*: they concern the way currently available solutions are used to create new ones. The values assigned to all the parameters cited influence the evolution and, hence, the final best solution found. DEREx users can

also choose the specific DE algorithm *DE_Algo* to be run among a set of ten possible ones.

Each possible solution proposed by DEREx is a set of rules each of which is of the type IF (condition) THEN (class). The condition part of each rule is composed of AND-connected literals, each of which is in the form

$(var_i\ OP\ const_1\ const_2)$

where var_i is an attribute in the data set, $const_1$ and $const_2$ are two constant values, and OP is a relational operator in the set $<, \leq, \geq, >, IN, OUT$. Actually, for the first four operators, just $const_1$ is considered, whereas the two latter represent the variable being within a given range or outside it, and hence require the use of both constants.

Users can set the maximum number of rules a set can contain, represented by *N_Max_Rules*. They can also set a rule threshold *Rule_Thr* in 0.0 – 1.0: the lower this value, the more likely a rule set will contain less than *N_Max_-Rules*. Similarly, a literal threshold *Lit_Thr* can be set within 0.0 - 1.0: the higher its value, the lower the number of literals in a rule. In this way, users can modulate the number of rules and their size in the proposed rule sets. For example, if DEREx is applied to a data set with two classes and six attributes, a possible solution proposed could be:

- IF (var1 $>$ 0.89) AND (var2 $>$ 0.23) THEN class = 1
- IF (var2 $\leq$ 0.19) AND (var5 $<$ 0.48) THEN class = 2
- IF (var4 IN (0.47, 0.65)) THEN class = 1

About the assignment of an item to a class, three possible cases can take place. In the first case, the item is taken by just one rule or by more rules related to the same class, in which case it is assigned to the class contained in that rule(s). In the second case, the item is taken by more rules related to different classes, which is what is defined as a *yes–yes* indeterminate situation. In the third case, the item is taken by no rule, which represents what is called a *no–no* indeterminate case.

DEREx is endowed with a recovery mechanism that allows resolving both *yes–yes* and *no–no* indeterminate cases, so that each item will always be assigned to one and only one class.

Details on all of this can be found in the seminal paper [38].

4 The data set

We downloaded from Kaggle the COVID-19 Radiography Dataset [41]. This data set was collected as a joint effort by a team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh in cooperation with collaborators from Pakistan and Malaysia. This took place under the supervision of medical doctors. The data set consists of a set of chest X-ray images for COVID-19 positive cases along with Normal items and Viral Pneumonia images as well [6, 7]. Each image in it is in portable network graphics (PNG) file format, is in gray-scale, and has a resolution of 299 · 299 pixels. This data set is under constant updating.

In addition to the COVID-19 Radiography Dataset used in this study, several data sets containing chest images related to Covid-19 exist, as it can be seen at, e.g., [42]. We chose this specific data set because it has a lot of positive features. Firstly, it contains images related to Covid-19, which is a current problem highly impacting our society and our goal here. Secondly, as a consequence of an international cooperation among several institutions providing imagery, it consists of a large number of images, quite larger than those of the other data sets listed in the above reference, and is possibly the largest image data set related to Covid-19. Thirdly, its quality is very good, to the point that it was awarded by the Kaggle Community as the winner of the Covid-19 Dataset Award. Finally, it is freely available.

As summarized in Table 1, from the original Kaggle data set, we have considered all the examples of the two classes *Covid-19*, with 3,616 items, and *Normal*, containing 10,192 examples. Therefore, the data set we use here consists of 13,808 images.

Figure 2 shows some examples of the classes. It can be seen that the images are very similar in color and shape, the differences being quite small. Yet, the expert eye of a doctor can actually spot the zones where differences exist.

5 Experiments

In the experiments reported in this paper, we do not perform any data pre-processing, meaning with this no data cleaning, noise reduction, image transformation, and data normalization take place. This could somehow impact the quality of the results, as these preliminary steps often help improve the quality of the images. This choice also impacts the reproducibility of our experiments, making their replicability easier for the scientific community, in order to let people introduce some improvements and/or compare their results with ours.

About the filter tool, in this paper, we use a version made available in GitHub [43] by Xirong Li et al. who

Table 1 The distribution of the considered images (items) over the two classes: Covid-19 and Normal

	#items
Covid-19	3,616
Normal	10,192
Total	13,808

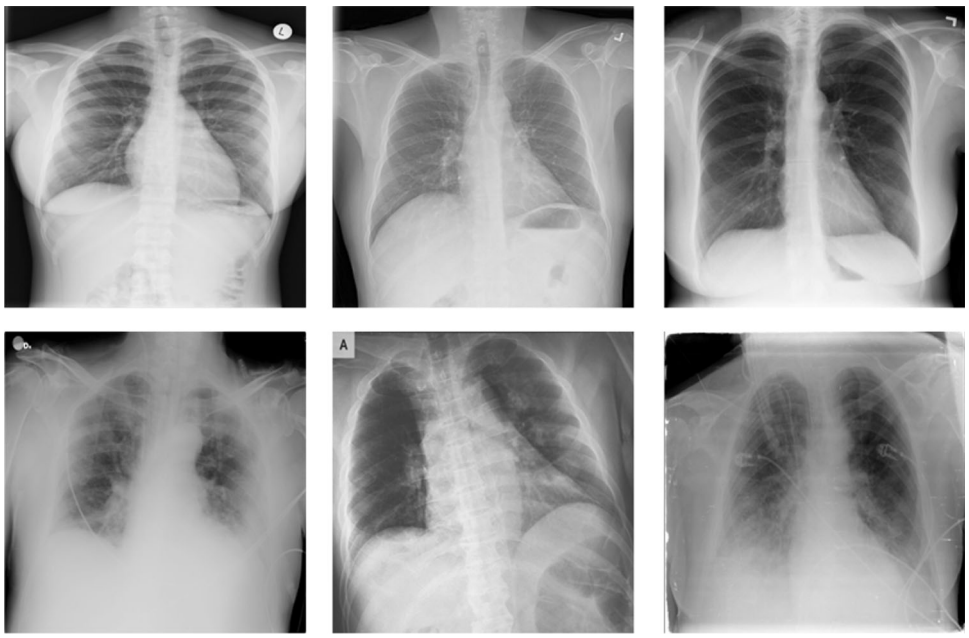

Fig. 2 Examples of items from the two classes. Top panes: *normal*. Bottom panes: *Covid-19*

made use of it as a part of a neighbor voting algorithm [44]. In that software, unlike Mingjing Li's paper, the order of the features is different: firstly we have the 44 correlogram features, then the 14 texture moment ones, and finally the six color moment ones.

Additionally, as said in Sect. 3.1, if the filter is applied to grayscale images, not all the 64 features will be meaningful, but only 29. In fact, we can see that 35 of them are always equal to 0.0 for all the items in the data set because all the correlogram features related to non-gray colors receive a value of 0.0 due to the nature of the image. Specifically, the only significant variations are, obviously, those related to black, white, and gray levels, which are represented by nine attributes as a whole. Therefore, the number of accounted features is reduced from 64 to 29.

Table 2 The order of the 29 attributes in the encoding

1	Color correlogram: black bin
2 to 8	Color correlogram: gray bins
9	Color correlogram: white bin
10–16	Mean values of the seven texture moment attributes
17–23	Variation values of the seven texture moment attributes
24	First-order R color moment
25	First-order G color moment
26	First-order B color moment
27	Second-order R color moment
28	Second-order G color moment
29	Second-order B color moment

Table 2 shows the position of each of the 29 attributes in the encoding, and shortly describes their meaning.

As for the classification step, instead, we have to highlight that DEREx is a stochastic algorithm, meaning that its execution depends on the value of a random seed that is initially set. Therefore, the algorithm has been run 25 runs with 25 different initial seeds, which provides different evolutions and, hence, different final solutions.

To investigate the classification ability, as the *fitness* function we have not taken into account the often-used accuracy A_{cc}, because the data set is highly unbalanced. In cases as the one faced here, instead, quality indices as F_score or Matthews correlation coefficient (MCC) should be used. Especially, this latter is becoming more and more recommended in case of unbalanced data sets, therefore we avail ourselves of it for this study.

To define it, firstly, we take as the positive class that containing the *Covid-19* items, the negative one being that with the items of the *Normal* class. Then, given a classification applied to the data set items, we can divide them as:

- true positive (*tp*): the positive items right assigned to the positive class;
- true negative (*tn*): the negative items right assigned to the negative class;
- false positive (*fp*): the negative items wrongly assigned to the positive class;
- false negative (*fn*): the positive items wrongly given to the negative class.

With these definitions, the MCC index is defined as:

$$MCC = \frac{(tp \cdot tn) - (fp \cdot fn)}{\sqrt{(tp + fp) \cdot (tp + fn) \cdot (tn + fp) \cdot (tn + fn)}} \quad (1)$$

The admissible range for MCC is [-1.0, 1.0], and the higher this value, the better the classification. Given this choice of the *fitness* function, the problem becomes a maximization problem.

To perform supervised learning, we divide the data set into a training set, containing the first 70% of the cases, and a test set with the remaining 30%.

5.1 Settings

The values for the parameters of DEREx have been set as shown in Table 3. These values have not been optimized by means of a preliminary tuning phase, rather they are the default ones, i.e., those we typically use when classifying over two-class numerical data sets.

These values say that we are looking for small rule sets composed of exactly two rules (one per class) (*N_Max_Rules* = 2 and *Rule_Thr* = 0.0), and each such a rule should contain a very low number of literals (*Lit_Thr* = 0.95). These choices imply our preference for compact and easy-to-understand knowledge, also to the detriment of accuracy values.

The two last rows of Table 3 report the correspondence of the class names to the integer values representing them.

After the filtering phase, the data set contains the items grouped class by class in sequential order, hence, we firstly randomly shuffle it, and then assign the first 9665 items to the training set and the last 4143 to the testing set. The resulting distribution of the items is summarized in Table 4.

To carry out our experimental phase, a version of DEREx has been implemented by us in C language, and a Mac Pro has been used that runs MacOS High Sierra and has the following hardware features: a total of 12 cores (two 3.5Ghz Intel Xeon E5 processors with 6 cores each), 256kB L2 cache for each core, 32 GB DDR3 ECC memory, 1TB PCIe-based SSD storage.

Table 3 The Parameter Setting for DEREx

Pop_Size	50
Max_Gens	500
Cr_Ratio	0.3
Mut_F	0.7
DE_Algo	DE/rand-to-best/1/bin
N_Max_Rules	2
Rule_Thr	0.0
Lit_Thr	0.95
Class 1	*Covid-19*
Class 2	*Normal*

Table 4 The distribution of the items over the training and testing sets. Each item is composed of 29 attributes and one class (Covid-19 or Normal)

	#Items	Percentage (%)	Covid-19	Normal
Training	9665	70	2541	7124
Testing	4143	30	1075	3068
Total	13,808	100	3616	10,192

5.2 Results

From a numerical viewpoint, our algorithm reaches good performance, because the average value for *MCC* over the train set is equal to 0.423 out of the 25 runs. Also, the generalization ability shown is good, because the *MCC* value over the previously unseen items in the test set, averaged over the 25 runs, is equal to 0.446. As it can be seen, there is no relevant difference in performance over the two sets, as these two numbers are quite similar to each other. This means that the generalization ability of the rule sets found is good.

We consider as the best run the one that achieves the highest *MCC* value on the training set, and we examine its behavior on the test set. For this problem, the best run obtains *MCC* values of 0.486 over the train and of 0.496 over the test set. Also, the *MCC* on the total data set is equal to 0.489.

Table 5 shows the confusion matrices over the train set, the test set, and the total data set for the best solution obtained. It is worth noting that the items are quite correctly assigned to both classes, although the *Normal* class is about three times the *Covid-19* one. In particular, for each class, the majority of the items is correctly assigned to that class, and this takes place for all three sets. Instead, the use of indices as the accuracy, not designed to effectively deal with such situations, could have likely led to the majority class incorporating many samples of the minority one, even more than half of them, and up to all of them. Thanks to *MCC*, this is not the case here. The last row of the table reports the values for Accuracy A_{cc} and *MCC*. As it can be seen, A_{cc} is higher than 80% overall three sets. This is especially important for the test set, as it means that generalization obtained by the system through the rules is good.

The best solution obtained is the following set of two rules:

Table 5 Confusion Matrices of the Best Rule Set. Covid-19 is class 1, instead Normal is class 2

Real class	Train set		Test set		Whole data set	
	Predicted class		Predicted class		Predicted class	
	Covid-19	Normal	Covid-19	Normal	Covid-19	Normal
Covid-19	1,574	967	670	405	2,244	1,372
Normal	955	6,169	396	2,672	1,351	8,841
	A_{cc}	*MCC*	A_{cc}	*MCC*	A_{cc}	*MCC*
	80.11%	0.486	80.67%	0.496	80.28%	0.489

IF (var5 OUT (0.235 0.301)) AND (var7 $\geq$ 0.143) AND (var27 $<$ 0.220) THEN class = 1
IF (var13 IN (0.074 0.098)) THEN class = 2

where class 1 is *Covid-19*, instead class 2 is *Normal*.

As it can be seen, just four parameters are contained out of the 29.

The rule for the *normal* class is extremely compact and easy to understand, as it contains just the mean value of the fourth texture moment attribute.

That for the *Covid-19* class, instead, contains three attributes, two of which are from the color correlogram, namely they correspond to two different gray levels, an intermediate level, and a light gray. The third attribute in the rule, instead, is related to the second-order R color moment. The meaning of this rule is that there are some levels of gray in the images related to Covid-19 that are not present in those from healthy subjects.

5.3 Comparison

To investigate the performance of the classification tool proposed here, we have compared its results against those provided by other widely used Machine Learning-based classifiers. Actually, such classifiers can be divided into groups, so that all those in the same group share the same, or similar, features. Namely, we have considered here the Bayesian methods, from which we have chosen the Bayes Net (BN) [45] and the Naive Bayes (NB) [46], the function-based, from which we have selected the radial basis function (RBF) [47] and the support vector machine (SVM) [48], the ensemble methods, for which AdaBoost (AB) [49] has been taken into account, and the rule-based tools, from which the one rule (OR) [50] has been picked.

Of course, the first step of the method has been left the same in all the cases, and we have given as input to these algorithms the same data set that is given in input to DEREx in the second step of our methodology.

To run these algorithms, we have used the Waikato Environment for Knowledge Analysis (WEKA) [51] tool, version 3.8.5. We have executed them in exactly the same experimental conditions used for DEREx: the same division into training and testing set, the same number of 25 runs only differing in the initial seed provided to the random number generator, and the absence of any preliminary parameter tuning phase.

Table 6 firstly reports for each of the algorithms the average final values over the 25 runs in terms of accuracy (A), and Matthews correlation coefficient (M). The next two columns report the highest values obtained in the 25 runs with respect to the same parameters. Finally, the last two columns show the corresponding values for the standard deviation.

For each column in the table, the best value achieved is shown in bold, and the second is reported in italic.

As it can be seen in the table, in terms of higher single performance, DEREx obtains the highest final values for both the considered parameters. In terms of MCC, the Bayes Net is the runner-up, whereas, when the accuracy is considered, the Radial Basis Function shows the second-best performance. This means that DEREx has found the best-quality classification model, which is the one shown in the previous subsection.

As far as the average final values are considered, instead, when the MCC is examined, the Bayes Net is the best performer, DEREx being the runner-up: they both are far superior to the other algorithms. In terms of accuracy, instead, the Radial Basis Function performs best, followed by DEREx.

Finally, in terms of standard deviation, as far as the accuracy is concerned, SVM has the lowest value, followed

Table 6 The results obtained by the algorithms

	Average		Best		Std. dev	
	A	M	A	M	A	M
BN	77.51	**0.474**	78.61	*0.492*	0.627	**0.012**
NB	76.62	0.372	77.41	0.397	0.516	**0.012**
RBF	**79.60**	0.410	*80.28*	0.442	*0.434*	0.016
SVM	78.55	0.361	79.39	0.391	**0.311**	**0.012**
AB	76.73	0.341	79.07	0.419	1.034	0.045
OR	76.31	0.317	77.76	0.354	0.935	0.023
DEREx	*78.86*	*0.446*	**80.67**	**0.496**	2.094	0.022

For each column, the best value achieved is shown in bold, and the second best is reported in italic

by RBF. When MCC is considered, instead, BN, NB, and SVM turn out to be the best. DEREx shows higher values than the other classifiers with respect to accuracy, whereas it is comparable with the other algorithms when MCC is considered.

It should be emphasized here that the algorithms the performance of which are closer to that of DEREx, i.e., the Bayes Net and the Radial Basis Function, do not provide easily interpretable knowledge, which limits their utility in the present scenario and, more generally, whenever easily understandable interpretable knowledge should be provided to the users.

5.4 Discussion

The first important pro of our approach lies in the fact that it is general-purpose, meaning with this that its applicability is not restricted to the specific area from which the images used here come, nor does it depend on their specific format. Rather, independently of the specific field, all it requires is a data set composed of images. Hence, if we remain in the medical area, it is applicable to other diseases/pathologies. Moreover, it can be profitably used in any other area where an image data set is available.

Then, the execution of a run of our approach on the above-described machine requires for this data set about six minutes, whereas Deep Neural Networks are typically slow, and, on this set, require many hours to obtain classification, even with small network configurations or low values of the parameters. As a first example, we have run a GoogLeNet DNN, consisting in a 22-layer deep convolutional neural network, that has required on the same machine described above about 190 minutes when set with low parameter values, whereas has taken about 19 hours when the parameter values are set higher. Noticeably, this difference in execution time increases with the data set size. As a further example, we have executed a ResNet-50 network, consisting of 50 layers. This latter has taken a training time range from about 90 minutes to about 600 minutes.

Another pro of the methodology proposed here lies in the fact that the images representing the classes are very similar, as can be appreciated in Fig. 2. Furthermore, also the typical colors of the images associated with the different classes are the same. This means that the algorithm has no easy way to assign the items to the classes thanks to different colors or shapes. Therefore, this experiment is a kind of a worst-case situation. Yet, the results are good.

Another positive aspect, we believe, is due to the fact that this methodology relies on colors, therefore its use seems promising whenever the items of different classes are typically represented by different colors. This is the case, for example, in the classification of birds on the CUB_200_2011 data set [52]. We have effected some preliminary experiments on it, with good results in discriminating among birds as common yellowthroat, red-cockaded woodpecker, and red-headed woodpecker. The data set used has three classes and about 60 items per class. Accuracy values obtained are 97.52% on the train set and 96.23% on the test set. These values imply that the use of colored images improves classification quality with respect to those in grayscale used here.

Moreover, the data set used here is relatively large, especially in the medical domain. In many practical cases, the opposite takes place: often doctors only have some dozens of examples, rather than hundreds or even thousands. In spite of this size, results are encouraging.

Another positive issue is the fact that this data set is quite unbalanced between the classes. This situation can in many cases lead to unsatisfactory classifications, yet here these problems do not hold true.

On the other side, a weakness of the experiments presented here lies in the fact that the behavior of this classifier has been tested on one data set only. This should be checked on different types of image data sets from other fields when items of a same class are represented in different colors, or items from different classes are represented in a same color.

Another investigation field is constituted by the filtering step. Research will be developed on the number of the filter attributes with reference to each of its three components: are more orders useful for the color moment? Does a number of bins different from 44 for the color correlogram provide better performance? Can other texture moments be defined?

It is very interesting to look for other filters that can provide other parameters more meaningful for us humans from the point of view of the interpretability. For instance, they could be pieces of information related to a given image area: “this part of the image suggests that...”.

6 Conclusions and future works

A new general-purpose two-step approach has been proposed to perform image classification. It relies both on Context-based Image Retrieval concepts and on an evolutionary-based automatic extraction of explicit knowledge. This approach has been tested on a data set of X-ray images related to Covid-19.

This paper represents a feasibility study for the proposed approach. The results obtained are encouraging, and some possible problems seem not to affect the approach, yet much investigation must be carried out to check its efficiency on other data sets, especially in the medical domain. Some directions for our future work have been evidenced

in the discussion reported in Sect. 5.4 in terms of investigation on the general applicability of this approach.

Funding No funding was received for conducting this study.

Declarations

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio (2016) Deep learning. MIT press Cambridge
2. J. Brownlee, Deep learning for computer vision: image classification, object detection, and face recognition in python. Mach Learn Mastery, 2019
3. D. Gunning, Explainable artificial intelligence (xai), Defense Advanced Research Projects Agency (DARPA), nd Web, 2(2), 2017
4. Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 1(5):206–215
5. World Health Organization, Coronavirus disease (covid-19) pandemic, https://www.who.int/emergencies/diseases/novel-coronavirus-2019, 2020, accessed: 2021-05-26
6. Chowdhury ME, Rahman T, Khandakar A, Mazhar R, Kadir MA, Mahbub ZB, Islam KR, Khan MS, Iqbal A, Al Emadi N et al. (2020) Can AI help in screening viral and covid-19 pneumonia? IEEE Access 8:132665–132676
7. Rahman T, Khandakar A, Qiblawey Y, Tahir A, Kiranyaz S, Kashem SBA, Islam MT, Al Maadeed S, Zughaier SM, Khan MS et al. (2021) Exploring the effect of image enhancement techniques on covid-19 detection using chest X-ray images. Comput Biol Med 132:104319
8. Nakane T, Bold N, Sun H, Lu X, Akashi T, Zhang C (2020) Application of evolutionary and swarm optimization in computer vision: a literature survey. IPSJ Trans Comput Vis Appl 12(1):1–34
9. R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy et al., Evolving deep neural networks, in Artificial intelligence in the age of neural networks and brain computing. Elsevier, 2019, pp 293–312
10. Sun Y, Xue B, Zhang M, Yen GG (2019) Evolving deep convolutional neural networks for image classification. IEEE Trans Evolut Comput 24(2):394–407
11. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
12. X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2010, pp 249–256
13. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J et al. (2018) Recent advances in convolutional neural networks. Pattern Recognit 77:354–377
14. T. Sinha, B. Verma, and A. Haidar, Optimization of convolutional neural network parameters for image classification, in 2017 IEEE Symposium Series on Computational Intelligence (SSCI).IEEE, 2017, pp 1–7
15. Zhang Q, Zhang M, Chen T, Sun Z, Ma Y, Yu B (2019) Recent advances in convolutional neural network acceleration. Neurocomput 323:37–51
16. LeCun Y, Jackel LD, Bottou L, Cortes C, Denker JS, Drucker H, Guyon I, Muller UA, Sackinger E, Simard P et al. (1995) Learning algorithms for classification: a comparison on handwritten digit recognition. Neural Netw: Stat Mech Perspect 261(276):2
17. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in Neural Information Prooces Syst 25:1097–1105
18. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9
19. K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, IEEE conference on computer vision and pattern recognition
20. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger,Densely connected convolutional networks, In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708
21. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185
22. K. Ganesan and H. Rajaguru, Performance analysis of KNN classifier with various distance metrics method for MRI images, in Soft computing and signal processing. Springer, 2019, pp 673–682
23. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
24. Quinlan JR (1987) Simplifying decision trees. Intl J Man-Mach Stud 27(3):221–234
25. Hertz J, Krogh A, Palmer RG, Horner H (1991) Introduction to the theory of neural computation. Phys Today 44(12):70
26. S. F. Da Silva, M. X. Ribeiro, J. d. E. B. Neto, C. Traina-Jr, and A. J. Traina, Improving the ranking quality of medical image retrieval using a genetic feature selection method, Decis Support Syst, 51(4): 810–820, 2011
27. D. S. Johnson, D. L. L. Johnson, P. Elavarasan, and A. Karunanithi, Feature selection using flower pollination optimization to diagnose lung cancer from CT images, In Future of Information and Communication Conference. Springer, 2020, pp. 604–620
28. Kharrat A, Mahmoud N (2019) Feature selection based on hybrid optimization for magnetic resonance imaging brain tumor classification and segmentation. Appl Med Inf 41(1):9–23
29. S. J. Narayanan, R. Soundrapandiyan, B. Perumal, and C. J. Baby, Emphysema medical image classification using fuzzy decision tree with fuzzy particle swarm optimization clustering, In Smart Intelligent Computing and Applications, 2019, pp 305–313
30. Li J, Fong S, Liu L-S, Dey N, Ashour AS, Moraru L (2019) Dual feature selection and rebalancing strategy using metaheuristic optimization algorithms in x-ray image datasets. Multimed Tools Appl 78(15):20913–20933
31. N. Dhanachandra and Y. J. Chanu, An image segmentation approach based on fuzzy c-means and dynamic particle swarm optimization algorithm. Multimed Appl
32. Li H, Pun C-M, Xu F, Pan L, Zong R, Gao H, Lu H (2021) A hybrid feature selection algorithm based on a discrete artificial bee colony for parkinsons diagnosis. ACM Trans Internet Technol 21(3):1–22
33. E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, Large-scale evolution of image

classifiers, In International Conference on Machine Learning. PMLR, 2017, pp 2902–2911
34. Ghosh A, Datta A, Ghosh S (2013) Self-adaptive differential evolution for feature selection in hyperspectral image data. Appl Soft Comput 13(4):1969–1977
35. Ghamisi P, Couceiro MS, Benediktsson JA (2014) A novel feature selection approach based on FODPSO and SVM. IEEE Trans Geosci Remote Sens 53(5):2935–2947
36. Ghamisi P, Chen Y, Zhu XX (2016) A self-improving convolution neural network for the classification of hyperspectral data. IEEE Geosci Remote Sens Lett 13(10):1537–1541
37. M. Li, (2007) Texture moment for content-based image retrieval, In 2007 IEEE International Conference on Multimedia and Expo. IEEE, 2007, pp 508–511
38. De Falco I (2013) Differential evolution for automatic rule extraction from medical databases. Appl Soft Comput 13(2):1265–1283
39. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359
40. Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer, Berlin
41. S. Dubey, Covid-19 radiography database, available online in kaggle, https://www.kaggle.com/tawsifurrahman/covid19-radiography-database, 2020, accessed: 2021-05-26
42. ELIXIR–IT, Covid–19 data portal italy, https://www.covid19dataportal.it/data_types/imaging_data/data/, 2020, accessed: 2021-10-14
43. X. Li, Features - a python lib for image feature extraction, available online in github, https://github.com/li-xirong/features, 2009, accessed: 2021-05-13
44. Li X, Snoek CG, Worring M (2009) Learning social tag relevance by neighbor voting. IEEE Trans Multimed 11(7):1310–1322
45. S. Russell and P. Norvig, Artificial intelligence: a modern approach, 2002
46. G. H. John and P. Langley (2013) Estimating continuous distributions in bayesian classifiers, arXiv preprint arXiv:1302.4964
47. Broomhead DS, Lowe D (1988) Radial basis functions, multivariable functional interpolation and adaptive networks. Tech. Rep, Royal Signals and Radar Establishment Malvern (United Kingdom)
48. Z.-Q. Zeng, H.-B. Yu, H.-R. Xu, Y.-Q. Xie, and J. Gao (2008) Fast training support vector machines using parallel sequential minimal optimization, In International conference on intelligent system and knowledge engineering, vol. 1. IEEE, pp. 997–1001
49. Y. Freund, R. E. Schapire et al., Experiments with a new boosting algorithm, in ICML, vol. 96. Citeseer, 1996, pp 148–156
50. Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Mach Learn 11(1):63–90
51. Garner SR et al. (1995) Weka: The waikato environment for knowledge analysis. New Zealand Comput Sci Res Stud Conf 1995:57–64
52. Caltech, Caltech-ucsd birds-200-2011, available online, http://www.vision.caltech.edu/visipedia/CUB-200-2011.html, accessed: 2021-05-13